

RH850/E2x-FCC1

User's Manual: Hardware

Renesas microcontroller
RH850 Family

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

Notes for CMOS devices

- (1) Voltage application waveform at input pin:** Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between VIL (MAX) and VIH (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between VIL (MAX) and VIH (MIN).
- (2) Handling of unused input pins:** Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to power supply or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.
- (3) Precaution against ESD:** A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.
- (4) Status before initialization:** Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.
- (5) Power ON/OFF sequence:** In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.
- (6) Input of signal during power off state:** Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

Table of Contents

Section 1 Overview	86
1.1 Outline.....	86
1.2 Features.....	88
1.3 Application Fields.....	95
1.4 Ordering Information.....	95
1.5 Differences in the Specifications of FCC Products.....	96
1.6 Pin Connection Diagram (Top View)	99
1.7 Functional Block Configuration.....	103
Section 2 Pin Function	105
2.1 Features of Pin Function.....	106
2.1.1 Port Group.....	106
2.1.2 Register Base Addresses	108
2.1.3 Clock Supplies	108
2.1.4 Interrupt Requests.....	108
2.1.5 Reset Sources.....	109
2.2 Overview.....	110
2.2.1 Functional Overview.....	110
2.2.1.1 Terms	110
2.2.1.2 Overview of Pin Functions	110
2.2.1.3 Pin Data Input/Output	113
2.3 Port Types	115
2.3.1 Pin Configuration.....	115
2.3.2 Port Control Logic	116
2.4 Registers.....	117
2.4.1 List of Registers	117
2.4.2 Port Group Unit Register Explanation.....	119
2.4.2.1 Configuration of PORT.....	119
2.4.2.2 Configuration of Pin Data Input/Output.....	129
2.4.2.3 Configuration of Electrical Characteristic.....	138
2.4.2.4 Other Registers.....	146
2.4.3 Pin-Unit Register Explanation	152
2.4.3.1 PCRn_m/JPCRn_m — Port Control Register.....	152
2.4.4 Port Setting Procedures	154
2.4.4.1 Batch Setting.....	155
2.4.4.2 Individual Setting.....	156
2.4.4.3 Protection Setting.....	160
2.4.5 Functional Selection.....	161
2.4.5.1 Register Configuration When Using Alternative Functions.....	161
2.4.5.2 Alternative Functions That Can Be Used in Direct I/O Control Alternative Mode...	161
2.4.5.3 Register Setting When Using an Analog Input Pin	163
2.4.5.4 Register Setting to Use the LVDS Buffer	164
2.4.5.5 Setting Flow of LVDS function	165

2.4.5.6	Selecton of Function for JTAG Port	166
2.4.5.7	Setting of the ERROROUT_C loopback	166
2.5	Organization of Port Groups	167
2.5.1	List of Port Functions	167
2.5.2	List of Alternative Functions	167
2.6	Functions	168
2.6.1	Digital Noise Filter	168
2.6.1.1	Example of Noise Elimination	168
2.6.1.2	Peripheral Function DNF	169
2.6.1.3	Edge Detection DNF	174
2.6.2	Port Safe State Mode	186
2.6.2.1	Block Diagram	187
2.6.2.2	Supported Pins	188
2.6.2.3	Configuration Flow	189
2.6.3	Virtual Port Function via RHSB (XBAR)	190
2.6.3.1	Block Diagram	190
2.6.4	Virtual Port Function via HSSPI	191
2.6.4.1	Block Diagram	191
2.6.5	Virtual Port Function via Analog	192
2.6.5.1	Block Diagram	192
2.6.6	Dedicated Analog Pin Table	193
2.6.7	ICUM Select Function	197
2.6.7.1	Block Diagram	197
2.6.8	RES_OUT Function	198
2.6.8.1	Overview	198
2.7	List of Pin Function	200
2.7.1	Overview	200
2.7.2	Details Function	201
2.8	Pins Status	224
2.9	Handling of The Unused Pins	229
2.9.1	Handling of Thermal Area Pins	232
2.9.1.1	The Case of using Emulation/Instrumentation RAM (SYSVCC and EMUVDD are always supplied)	232
2.9.1.2	The Case of using Emulation/Instrumentation RAM and Aurora I/F (EMUVCC and EMUVDD are always supplied)	233
2.9.1.3	The Case of using Emulation/Instrumentation RAM (EMUVCC and EMUVDD are always supplied)	234
2.9.1.4	The Case of Pin Assignment for Mass Production	235
Section 3	CPU System	236
3.1	Overview	236
3.1.1	Block Diagram	236
3.2	CPU	238
3.2.1	Core Functions	238
3.2.1.1	Features	238
3.2.2	Processor Model	238

3.2.2.1	CPU Operating Modes.....	239
3.2.2.2	Instruction Execution.....	245
3.2.2.3	Exceptions and Interrupts	247
3.2.2.4	Coprocessors.....	249
3.2.2.5	Registers.....	250
3.2.2.6	Data Types.....	253
3.2.2.7	Address Space.....	260
3.2.2.8	Execution Timing of a Store Instruction	269
3.2.2.9	Memory Ordering	269
3.2.2.10	Acquiring the CPU Number	269
3.2.2.11	System Protection Identifier (SPID)	269
3.2.2.12	Timestamp Counter	270
3.2.2.13	Performance Measurement Function	271
3.2.3	Register Set	272
3.2.3.1	Program Registers	272
3.2.3.2	Basic System Registers	274
3.2.3.3	Interrupt Function Registers	298
3.2.3.4	FPU Function Registers.....	302
3.2.3.5	FXU Function Registers.....	309
3.2.3.6	MPU Function Registers.....	318
3.2.3.7	Cache Operation Function Registers.....	326
3.2.3.8	Count Function Registers	334
3.2.3.9	Hardware Function Registers	342
3.2.4	Exceptions and Interrupts	346
3.2.4.1	Outline of Exceptions.....	346
3.2.4.2	Operation when Acknowledging an Exception	357
3.2.4.3	Return from Exception Handling.....	360
3.2.4.4	Exception Handler Address	362
3.2.4.5	Register Bank Function.....	372
3.2.4.6	List of Memory Access Exceptions	380
3.2.5	Memory Management	381
3.2.5.1	Memory Protection Unit (MPU).....	381
3.2.5.2	Cache.....	392
3.2.6	Coprocessor.....	401
3.2.6.1	Floating-Point Operation	401
3.2.6.2	Extended Floating-Point Operation.....	418
3.2.7	Hazard Control	431
3.2.7.1	Synchronization Processing	431
3.2.7.2	Guaranteeing the Completion of Store Instruction	433
3.2.7.3	Hazard Management after System Register Update	437
3.2.8	Reset.....	442
3.2.8.1	Status of Registers After Reset	442
3.2.9	Ensuring Coherency after Code Flash Programming	443
3.2.9.1	Registers	443
3.3	Inter-CPU Overview	446
3.3.1	Features	446
3.3.1.1	Number of Units	446
3.3.1.2	Register Base Addresses	447
3.3.1.3	Clock Supplies	447
3.3.1.4	Interrupt Requests	447
3.3.1.5	Reset Sources	447

3.3.1.6	External Input/Output Signals	448
3.3.2	Processor Element Identifier	449
3.4	Inter-Processor Interrupts	450
3.4.1	Inter-Processor Interrupts Overview	450
3.4.2	Inter-Processor Interrupts Registers	450
3.4.2.1	List of Registers	450
3.4.2.2	Self Region	451
3.4.2.3	IPIInENm — IPIRn Inter-PE Interrupt Enable Register m (n = 0 to 3)	452
3.4.2.4	IPIInFLGm — IPIRn Inter-PE Interrupt Flag Register m (n = 0 to 3)	453
3.4.2.5	IPIInFCLRm — IPIRn Inter-PE Interrupt Clear Register m (n = 0 to 3)	454
3.4.2.6	IPIInREQm — IPIRn Inter-PE Interrupt Request Register m (n = 0 to 3)	455
3.4.2.7	IPIInRCLRm — IPIRn Inter-PE Interrupt Request Clear Register m (n = 0 to 3)	456
3.4.3	Inter-Processor Interrupts Function	457
3.4.3.1	Initial Setting	457
3.4.3.2	Inter-PE Interrupt Request	458
3.4.3.3	Request Mask Function	463
3.4.3.4	Inter-PE Interrupt Requests of Multiple Systems	463
3.4.3.5	Inter-PE Interrupt Request Clear Function	464
3.5	Mutual Exclusion	465
3.5.1	Mutual Exclusion Overview	465
3.5.2	Mutual Exclusion Registers	466
3.5.2.1	List of Registers	466
3.5.2.2	G0MEVn — Mutual Exclusion Variable Register n (n = 0 to 31)	467
3.5.3	Mutual Exclusion Function	468
3.5.3.1	Operation Example Using Mutual Exclusion Variable Register (MEV)	468
3.5.3.2	Shared Data that does not Require Mutual Exclusion Processing	471
3.5.3.3	Operation of the LDL and STC Instructions	472
3.5.3.4	Performing Mutual Exclusion by Using the SET1 Instruction	474
3.5.3.5	Performing Mutual Exclusion by Using the CAXI Instruction	475
3.6	Barrier-Synchronization	476
3.6.1	Barrier-Synchronization Overview	476
3.6.2	Barrier-Synchronization Registers	477
3.6.2.1	List of Registers	477
3.6.2.2	Self Region	477
3.6.2.3	BRnINIT — Barrier-Synchronization n Initialization Register	478
3.6.2.4	BRnEN — Barrier-Synchronization n Barrier Participating PE Setting Register	478
3.6.2.5	BRnCHKm — Barrier-Synchronization n Barrier Check Register m	479
3.6.2.6	BRnSYNCm — Barrier-Synchronization n Barrier Synchronization Completion Register m	480
3.6.3	Barrier-Synchronization Function	481
3.6.3.1	Initial Setting	481
3.6.3.2	Barrier Synchronization	482
3.7	TPTM	483
3.7.1	TPTM Overview	483
3.7.2	TPTM Registers	484
3.7.2.1	List of Registers	484
3.7.2.2	Self Region	486
3.7.2.3	TPTMnIRUN — Counter Start Register of Interval Timer for PEn (n = 0 to 1)	488

3.7.2.4	TPTMnIRRUN — Counter Restart Register of Interval Timer for PEn (n = 0 to 1)	489
3.7.2.5	TPTMnISTP — Counter Stop Register of Interval Timer for PEn (n = 0 to 1)	490
3.7.2.6	TPTMnISTR — Counter Status Register of Interval Timer for PEn (n = 0 to 1)	491
3.7.2.7	TPTMnIEN — Interrupt Enable Register of Interval Timer for PEn (n = 0 to 1)	492
3.7.2.8	TPTMnIUSTR — Underflow Status Register of Interval Timer for PEn (n = 0 to 1)	493
3.7.2.9	TPTMnIDIV — Divider Register of Interval Timer for PEn (n = 0 to 1)	494
3.7.2.10	TPTMnFRUN — Counter Start Register of Free-run Timer for PEn (n = 0 to 1)	495
3.7.2.11	TPTMnFRRUN — Counter Restart Register of Free-run Timer for PEn (n = 0 to 1)	496
3.7.2.12	TPTMnFSTP — Counter Stop Register of Free-run Timer for PEn (n = 0 to 1)	497
3.7.2.13	TPTMnFSTR — Counter Status Register of Free-run Timer for PEn (n = 0 to 1)	498
3.7.2.14	TPTMnFDIV — Divider Register of Free-run Timer for PEn (n = 0 to 1)	499
3.7.2.15	TPTMnICNTm — Counter Channel m Register of Interval Timer for PEn (n = 0 to 1, m = 0 to 3)	500
3.7.2.16	TPTMnILDm — Load Channel m Register of Interval Timer for PEn (n = 0 to 1, m = 0 to 3)	501
3.7.2.17	TPTMnFCNT — Counter Register of Free-run Timer for PEn (n = 0 to 1)	502
3.7.3	TPTM Function	503
3.7.3.1	Normal Operation	503
3.7.3.2	Operation during a Restart	505
3.7.3.3	Operation during an Underflow	506
3.8	Usage Notes	507
3.8.1	Synchronization of Store Instruction Completion and Subsequent Instruction Generation	507
3.8.1.1	When Updated Results in the Control Registers or Memory are Reflected in the Implementation of a Subsequent Instruction	507
3.8.1.2	When the Updated Results in the Control Registers and Memories are Reflected in the Instruction Fetch of a Subsequent Instruction	508
3.8.1.3	Product information of SYNCM	508
3.8.1.4	Product information of Waiting for the Completion of Storage	508
3.8.2	Synchronization of Load Instruction Completion and Subsequent Instruction Generation	509
3.8.3	Accesses to Registers by Bit-Manipulation Instructions	509
3.8.4	Ensuring Coherency after Code Flash Programming	509
3.8.5	Overwriting Context when Acknowledging Multiple Exceptions	509
3.8.6	Usage Notes on Prefetching	510
3.8.7	Product Information of Initial Value for G4MH Register	511
3.8.8	Product Information of SYSERR Factor	512
3.8.9	Product Information of Cache Structure	512
3.8.10	Product Information of Fetch Size	512
3.8.11	Register Initialization	512
Section 4 Address MAP		513
4.1	Address MAP	513
4.2	Address Space Viewed from Each Bus Master	514
4.2.1	Space in which Instructions can be Fetched	514
4.2.2	Data Space Accessible by CPU0, CPU1	515

4.2.3	Data Space Accessible by ICUM	516
4.2.4	Data Space Accessible by DMA (sDMAC, DTS)	516
4.2.5	Data Space Accessible by H-Bus Master	516
4.3	Cluster RAM and Retention RAM	518
Section 5 Operating Modes		519
5.1	Features	519
5.1.1	Normal Operation Mode	520
5.1.2	User Boot Mode	520
5.1.3	Serial Programming Mode	520
5.2	Input Pins	521
5.3	Register Description	522
5.3.1	List of Registers	522
5.3.2	MODE — Mode Register	523
5.3.3	FHVE3 — FHVE3 Control Register	524
5.3.4	FHVE15 — FHVE15 Control Register	524
5.4	Mode Error	525
Section 6 Interrupts		526
6.1	Features of Interrupt Units	526
6.1.1	Units and Channels	526
6.1.2	Register Base Address	528
6.1.3	Clock Supply	528
6.1.4	Interrupt Requests	528
6.1.5	Reset Sources	529
6.1.6	External Input/Output Signals	529
6.2	Overview	530
6.2.1	Outline	530
6.2.2	Functional Overview	531
6.2.3	Interrupt Sources	532
6.2.3.1	FE Level Non-Maskable Interrupts	532
6.2.3.2	FE Level Maskable Interrupts	532
6.2.3.3	EI Level Maskable Interrupts	532
6.2.4	Block Diagram	533
6.3	Registers	534
6.3.1	List of Registers	534
6.3.2	EIC0 to EIC511 — EI Level Interrupt Control Register 0 to 511	535
6.3.3	IMR0 to IMR15 — EI Level Interrupt Mask Register 0 to 15	537
6.3.4	EIBD0 to EIBD31 — EI Level Interrupt Bind Register 0 to 31	538
6.3.5	EIBD32 to EIBD511 — EI Level Interrupt Bind Register 32 to 511	539
6.3.6	NMICTL — NMI Interrupt Control Register	540
6.3.7	EXINTCTL — External Interrupt Control Register	541
6.3.8	EXINTSTR — External Interrupt Status Register	542

6.3.9	EXINTSTC — External Interrupt Status Clear Register.....	543
6.3.10	SINTR0 to SINTR3 — Software Interrupt Register.....	544
6.3.11	PINTn + x — Peripheral Interrupt Status Register.....	545
6.3.12	PINTCLRn + x — Peripheral Interrupt Status Clear Register.....	546
6.3.13	TPTMSEL — TPTM Interrupt FE EI Select Register.....	547
6.4	Interrupt Operation.....	548
6.4.1	Level Interrupts.....	548
6.4.2	External Interrupts (NMI / IRQ).....	548
6.4.3	Inter-Processor Interrupts.....	548
6.4.4	Broadcast Interrupts.....	548
6.4.5	Software Interrupts.....	548
6.4.6	DTS Interrupt Merge Function.....	548
6.4.7	Priority Level Handling.....	549
6.5	Interrupt Processing Flow.....	550
6.5.1	Level Interrupt Processing Flow.....	550
6.5.2	NMI Processing Flow.....	551
6.5.3	External Interrupt Processing Flow.....	552
6.5.4	Broadcast Interrupt Processing Flow.....	553
6.5.5	Software Interrupt Processing Flow.....	554
6.5.6	DTS Interrupt Processing Flow.....	555
6.6	Interrupt Response Times.....	557
Section 7 sDMA Controller (sDMAC).....		558
7.1	Features of sDMAC.....	558
7.1.1	Number of Units.....	558
7.1.2	Register Base Addresses.....	559
7.1.3	Clock Supply.....	559
7.1.4	Interrupts.....	559
7.1.5	sDMAC Transfer Requests.....	562
7.1.6	Reset Sources.....	562
7.2	Overview.....	563
7.2.1	Functional Overview.....	563
7.2.2	Block Diagram.....	564
7.3	Registers of sDMAC.....	566
7.3.1	List of Registers.....	566
7.3.2	DMAjESTA — DMA Address Error Notification Status Register.....	568
7.3.3	DMAjISTA — DMA Interrupt Status Register.....	569
7.3.4	DMAjCHPRI — DMA Channel Request Priority Register.....	570
7.3.5	DMAjOR — DMA Operation Register.....	571
7.3.6	DMAjCHRST — DMA Channel Reset Register.....	572
7.3.7	DMAjCM_n — DMA Channel Master Setting Register.....	573
7.3.8	DMAjSAR_n — DMA Source Address Register.....	574

7.3.9	DMAjDAR_n — DMA Destination Address Register	575
7.3.10	DMAjTSR_n — DMA Transfer Size Register	576
7.3.11	DMAjTSRB_n — DMA Transfer Size Register B.....	577
7.3.12	DMAjTMR_n — DMA Transfer Mode Register	578
7.3.13	DMAjCHCR_n — DMA Channel Control Register.....	580
7.3.14	DMAjCHSTP_n — DMA Channel Stop Register	582
7.3.15	DMAjCHSTA_n — DMA Channel Status Register	583
7.3.16	DMAjCHFCR_n — DMA Channel Flag Clear Register	585
7.3.17	DMAjGIAI_n — DMA Gather Inner Address Increment Register	587
7.3.18	DMAjGOAI_n — DMA Gather Outer Address Increment Register.....	587
7.3.19	DMAjSIAI_n — DMA Scatter Inner Address Increment Register	588
7.3.20	DMAjSOAI_n — DMA Scatter Outer Address Increment Register.....	588
7.3.21	DMAjSGST_n — DMA Scatter Gather Status Register.....	589
7.3.22	DMAjSGCR_n — DMA Scatter Gather Control Register.....	590
7.3.23	DMAjRS_n — DMA Resource Select Register.....	591
7.3.24	DMAjBUFCR_n — DMA Buffer Control Register	593
7.3.25	DMAjDPPTR_n — DMA Descriptor Pointer Register	594
7.3.26	DMAjDPCR_n — DMA Descriptor Control Register.....	595
7.3.27	Descriptor RAM — Descriptor Memory.....	595
7.3.28	DMACSELj_m — sDMACj Transfer Request Group Selection Register m (m = 0 to 15).....	596
7.4	Operation	598
7.4.1	DMA Transfer Requests	598
7.4.2	DMA Interrupts	603
7.4.3	Channel Priority.....	604
7.4.4	Slow Speed Mode	605
7.4.5	Scatter Gather Transfer	605
7.4.6	Descriptors	610
7.4.7	Transfer Flow	623
7.4.8	Performance.....	626
7.5	Usage Notes	627
7.6	Reliability Function.....	630
7.6.1	Overview	630
7.6.2	Master Information Inheritance Function	630
7.7	Setting up DMA Transfer	631
7.7.1	Overview of Setting up DMA Transfer	631
7.7.2	Setting up the Transfer Request Group Selection	631
7.7.3	Setting up the Overall DMA Operation.....	631
7.7.4	Setting up the DMA Channel Setting	632
Section 8 DTS Controller		633
8.1	Features of DTS	633
8.1.1	Number of Units and Channels.....	633
8.1.2	Register Base Addresses	633

8.1.3	Clock Supplies	634
8.1.4	Interrupt Requests.....	634
8.1.5	DTS Transfer Requests for DTS.....	634
8.1.6	Reset Sources.....	634
8.1.7	External Input/Output Signals	634
8.2	Overview	635
8.2.1	Functional Overview.....	635
8.2.2	Definition of Terms	636
8.3	Registers.....	637
8.3.1	List of Registers	637
8.3.2	Description of Global Registers	639
8.3.2.1	DTSCCTL1 — DTS Control Register 1	639
8.3.2.2	DTSCCTL2 — DTS Control Register 2	640
8.3.2.3	DTSSSTS — DTS Status Register	641
8.3.2.4	DTSER — DTS Error Register	642
8.3.2.5	DTSPRn — DTS Channel Priority Setting Register (n = 0 to 7).....	643
8.3.2.6	DTSnnnCM — DTS Channel Master Setting Register (nnn = 000 to 127)	647
8.3.3	Description of DTS Channel Registers	649
8.3.3.1	Transfer Information of DTS (TI).....	649
8.3.3.2	Details of DTS Channel Registers	653
8.3.4	Detail DTSSSELM	668
8.3.4.1	DTSSSELM — DTS Transfer Request Group Selection Register (m = 0 to 15)	668
8.4	Operation	670
8.4.1	Basic Operation of DTS Transfer.....	670
8.4.1.1	Transfer Mode.....	670
8.4.1.2	Executing a DTS Cycle	670
8.4.1.3	Updating Transfer Information	671
8.4.1.4	Last Transfer and Address Reload Transfer	672
8.4.1.5	Transfer Completion Interrupt and Transfer Count Match Interrupt Output	672
8.4.1.6	Continuous Transfer	673
8.4.2	Channel Priority Order	674
8.4.2.1	DTS Channel Arbitration	674
8.4.3	Reload Function	676
8.4.3.1	Overview of the Reload Function.....	676
8.4.3.2	Operation of Reload Function 1	676
8.4.3.3	Reload Function 2.....	678
8.4.3.4	Timing of Setting DTS Reload Registers	679
8.4.4	Chain Function	680
8.4.4.1	Overview	680
8.4.4.2	Setting Up the Chain Function.....	681
8.4.5	DTS Operation	682
8.4.5.1	Types of DTS Transfer Requests and Assigning DTS Transfer Requests	682
8.4.5.2	Generating and Accepting a DTS Transfer Request	682
8.4.5.3	Executing DTS Transfer.....	683
8.4.5.4	DTSRAM Access	683
8.5	Suspension, Resume, and Transfer Abort, and Clearing a DTS Transfer Request	684
8.5.1	Suspend, Resume, and Transfer Abort of a DTS Transfer.....	684

8.5.2	Masking and Clearing a Hardware DTS Transfer Request by DTSFSL.....	685
8.5.3	List of Suspend, Resume, and Transfer Abort Functions	685
8.6	Error Control	686
8.6.1	Types of Errors.....	686
8.6.2	DTS Transfer Error.....	687
8.6.2.1	Operation of DTS When a DTS Transfer Error Occurs	687
8.6.3	DTSRAM Error	687
8.7	Reliability Functions	688
8.7.1	Overview	688
8.7.2	Master Information Inheritance Function	688
8.7.3	Restriction on Chain Function.....	689
8.8	Setting up DTS Transfer	690
8.8.1	Overview of Setting up DTS Transfer	690
8.8.2	Setting up the Transfer Request Group Selection	690
8.8.3	Setting up the Overall DTS Operation	691
8.8.4	Configuring the DTS Channel Settings.....	692
Section 9	RAM.....	693
9.1	List of On-Chip RAMs	693
9.2	Features.....	694
9.3	RAM Data Retention.....	694
9.4	Usage Notes	694
Section 10	Reset Controller	695
10.1	Features.....	695
10.1.1	Register Base Addresses	696
10.1.2	Input/Output Pins	696
10.2	Register Description	697
10.2.1	Register Protection.....	697
10.2.2	List of Registers	697
10.2.3	Reset of Registers.....	698
10.2.4	BOOTCTRL — Boot Control Register	699
10.2.5	RESF — Reset Factor Register	700
10.2.6	RESFC — Reset Factor Clear Register.....	703
10.2.7	SWSRESA — Software System Reset Assertion Register	705
10.2.8	SWARESAS — Software Application Reset Assertion Register	706
10.2.9	RESC — Reset Configuration Register	707
10.2.10	STAC_DTSRAM — RAM Initialization Mode Control Register for DTSRAM	708
10.2.11	STAC_DPRAM — RAM Initialization Mode Control Register for sDMAC Descriptor RAM	709
10.2.12	STAC_DFE — RAM Initialization Mode Control Register for DFE	710
10.2.13	STAC_FLXA — RAM Initialization Mode Control Register for FlexRay	711

10.2.14	STAC_ETHER — RAM Initialization Mode Control Register for Ethernet.....	712
10.2.15	STAC_GTM — RAM Initialization Mode Control Register for GTM	713
10.2.16	RESKCPROT — Reset Controller Register Key Code Protection Register.....	714
10.3	Operation	715
10.3.1	Reset Categories	715
10.3.2	Reset Sources.....	716
10.3.3	Reset Flags.....	718
10.3.4	FACI Reset Transfer	719
10.3.5	HW BIST	719
10.3.6	RAM Initialization	719
10.3.7	Reset Output ($\overline{\text{RES_OUT}}$).....	719
10.4	Usage Notes	720
10.4.1	RAM and Register Data Retention when External Reset Is Asserted	720
10.4.2	VMON Reset Assertion	720
Section 11	Power Supply.....	721
11.1	Features.....	721
11.2	External Pin List.....	721
11.3	Example of Power Supply Connection	723
11.3.1	Example of Power Supply Connection.....	723
11.4	Power Up/Down Timing.....	725
11.5	Usage Note.....	725
Section 12	Power Supply Voltage Monitor.....	727
12.1	Overall Configuration	727
12.1.1	Features	727
12.2	Primary Detection of Voltage Monitor (VMON).....	730
12.2.1	Features	730
12.2.2	Register Base Address	730
12.2.3	Clock Supply	730
12.2.4	Interrupt Requests.....	730
12.2.5	External Input and Output Pins	730
12.2.6	Overview	731
12.2.6.1	Block Diagram.....	731
12.2.7	Registers	733
12.2.7.1	List of Registers	733
12.2.7.2	VMONF — VMON Factor Register.....	735
12.2.7.3	VMONFC — VMONF Clear Register.....	736
12.2.7.4	VMONDIAGME — VMON DIAG Monitor Enable Register.....	737
12.2.7.5	VMONDIAGMEW — VMONDIAGME Write Register.....	738
12.2.7.6	VMONDMASK — VMON Detection Output Diagnosis Mask Register.....	739
12.2.7.7	VMONDIAG — VMON DIAG Mode Setting Register	740
12.2.7.8	VMONDIAGFE — VMON DIAG Filter Enable Register.....	741
12.2.7.9	VDDDE — VDD Detection Enable Register	742

12.2.7.10	VDDFCR — VDD Monitor Filter Control Register	743
12.2.7.11	VCCDE — VCC Detection Enable Register	744
12.2.7.12	VCCFCR — VCC Monitor Filter Control Register	745
12.2.7.13	EVCCDE — EVCC Detection Enable Register	746
12.2.7.14	EVCCFCR — EVCC Monitor Filter Control Register	747
12.2.7.15	VMONKCPROT — VMON Register Key Code Protection Register	748
12.2.8	Operation	749
12.2.8.1	VMON Basic Function.....	749
12.2.8.2	VMON Function in BIST (Mask of VMON_VDD_ERROROUT and VMON_VDDREQRESET).....	752
12.2.8.3	VMON Diagnosis Function	753
12.2.9	Usage Notes	755
12.2.9.1	Flag Bit (VMONF)	755
12.2.9.2	VCC Primary Low Detection	755
12.2.9.3	External Reset occurs during self-diagnosis.....	755
12.3	Delay Monitor (DMON)	756
12.3.1	Features	756
12.3.2	Register Base Addresses	756
12.3.3	Clock Supply	756
12.3.4	Interrupt and DMA/DTS Requests	756
12.3.5	External Input and Output Pins	756
12.3.6	Overview	757
12.3.6.1	Functional Overview	757
12.3.6.2	Block Diagram.....	757
12.3.7	Registers	758
12.3.7.1	List of Registers	758
12.3.7.2	DMONF — DMON Factor Register	759
12.3.7.3	DMONFC — DMONF Clear Register	760
12.3.7.4	DMONDIAGME — DMON DIAG Monitor Enable Register	761
12.3.7.5	DMONDIAGMEW — DMONDIAGME Write Register	762
12.3.7.6	DMONDIAG — DMON DIAG Mode Setting Register.....	763
12.3.7.7	DMONDMASK — DMON Detection Output Diagnosis Mask Register	764
12.3.7.8	DMONDE — DMON Detection Enable Register	765
12.3.7.9	DMONFCR — DMON Filter Control Register.....	766
12.3.7.10	DMONKCPROT0 — DMON Register Key Code Protection Register 0	767
12.3.8	Operation	768
12.3.8.1	DMON Basic Function	768
12.3.8.2	DMON Diagnosis Function	773
12.4	Usage Notes	776
12.4.1	Judgment Method of Reset Factors for Voltage Monitor	776
Section 13	Temperature Sensor	777
13.1	Features.....	777
13.1.1	Number of Channels	777
13.1.2	Register Base Address	777
13.1.3	Clock Supply	777
13.1.4	Interrupt Requests.....	777
13.1.5	External Input and Output Pins	777

13.2	Overview.....	778
13.2.1	Functional Overview.....	778
13.2.2	Block Diagram.....	779
13.3	Register.....	780
13.3.1	List of registers.....	780
13.3.2	Reset of Registers.....	780
13.3.3	OTS00TSTCR — Temperature Measurement Start Control Register.....	781
13.3.4	OTS00TENDCR — Temperature Measurement End Control Register.....	782
13.3.5	OTS00TCR — Temperature Sensor Control Register.....	783
13.3.6	OTS00TFCR — Temperature Sensor Flag Clear Register.....	784
13.3.7	OTS00TFR — Temperature Sensor Flag Register.....	785
13.3.8	OTS00TSTR — Temperature Status Register.....	786
13.3.9	OTS00TDR — Temperature Data Register.....	788
13.3.10	OTS0HTBRmn — High-Temperature Border mn Register (m = A or B, n = U or L).....	789
13.3.11	OTS0LTBRAn — Low-Temperature Border An Register (n = U or L).....	790
13.3.12	OTS0TDLR — Temperature Difference Limiting Register.....	791
13.3.13	OTS0COEFFRn — Coefficient n Register (n = A, B, C).....	792
13.4	Operation.....	793
13.4.1	Temperature Measurement Sequence.....	793
13.4.2	Examples of Temperature Measurement Operation.....	794
13.4.3	Temperature Measurement End Interrupt Request.....	796
13.4.4	Temperature Alarm Error and Temperature Rise/Drop Interrupt and Temperature Sensor Error Interrupt Requests.....	797
Section 14	Clock Monitor.....	799
14.1	Features.....	799
14.1.1	Number of Channels.....	799
14.1.2	Register Base Addresses.....	799
14.1.3	Clock Supply.....	800
14.1.4	Interrupt Requests.....	800
14.1.5	External Input and Output Pins.....	800
14.2	Overview.....	801
14.2.1	Functional Overview.....	801
14.2.2	Block Diagram.....	801
14.3	Register.....	802
14.3.1	Register Protection.....	802
14.3.2	List of Registers.....	802
14.3.3	Resetting the Registers.....	803
14.3.4	CLMAnCTL — CLMAn Control Register.....	804
14.3.5	CLMAnCMPL — CLMAn Compare Register L.....	805
14.3.6	CLMAnCMPH — CLMAn Compare Register H.....	806
14.3.7	CLMATEST — Clock Monitor Test Register.....	807

14.3.8	CLMATESTS — Clock Monitor Test Status Register	808
14.3.9	CLMABCE — Clock Monitor Backup Clock Enable Register	809
14.3.10	CLMAKCPROT — Clock Monitor Register Key Code Protection Register	810
14.4	Functional Description	811
14.4.1	Detection of Abnormal Clock Frequencies.....	811
14.4.2	Error Notification	814
14.4.3	Self-Diagnosis	814
14.5	Operation	815
14.5.1	Procedures to Enable CLMAn	815
14.5.2	Reset Procedure Using CLMATEST.RESCLM	815
14.5.3	Procedures to do Self-Diagnosis	816
Section 15	Clock Controller	818
15.1	Features.....	818
15.1.1	Register Base Addresses	818
15.1.2	External Input / Output Pins	818
15.2	Overview	819
15.2.1	Types of Clocks.....	819
15.2.2	Block Diagram.....	821
15.3	Register Description	822
15.3.1	Register Protection.....	822
15.3.2	List of Registers	822
15.3.3	Reset of Registers.....	822
15.3.4	CLKD0DIV — Clock Divider 0 Divisor Register	823
15.3.5	CLKD0STAT — Clock Divider 0 Status Register.....	824
15.3.6	CLKDnDIV — Clock Divider Divisor Register (n = 2).....	825
15.3.7	CLKDnSTAT — Clock Divider Status Register (n = 2)	826
15.3.8	CKSC0C — Clock Selector 0 Control Register	827
15.3.9	CKSC0S — Clock Selector 0 Status Register	828
15.3.10	CKSCnC — Clock Selector Control Register (n = 2).....	829
15.3.11	CKSCnS — Clock Selector Status Register (n = 2).....	830
15.3.12	PLLCLKS — PLL Oscillation Status Register.....	831
15.3.13	MOSCS — Main Oscillation Status Register	832
15.3.14	OSCMONMD — OSC Monitor Mode Entry Register.....	833
15.3.15	CLKKCPROT0 — Clock Controller Register Key Code Protection Register 0.....	834
15.3.16	CLKKCPROT1 — Clock Controller Register Key Code Protection Register 1.....	835
15.4	Operation	836
15.4.1	Sequence for Shifting the Clock Gear Up	836
15.4.2	External Clock Output Pins (CK).....	837
15.4.3	Main OSC Configuration and Stabilization Sequence	838
15.4.4	OSC Monitor Mode	840
15.5	Usage Notes	842

15.5.1	Main OSC setting	842
15.5.2	Clock Gear Up.....	842
15.5.3	FPU Operation Notes.....	842
15.5.4	Board Design Notes	842
Section 16	Standby Controller	843
16.1	Features.....	843
16.1.1	Register Base Addresses	843
16.2	Power Down Modes.....	844
16.2.1	Power Off Standby Mode	844
16.2.2	Emulation/Instrumentation RAM and Aurora Retention Mode (E2x-FCC1 Only).....	844
16.2.3	HALT Mode	844
16.2.4	Module Standby	844
16.3	Description of Registers.....	845
16.3.1	Register Protection.....	845
16.3.2	List of Registers	845
16.3.3	Reset of Registers.....	845
16.3.4	MSR_ATU — Module Standby Register for ATU5	846
16.3.5	MSR_GTM — Module Standby Register for GTM.....	847
16.3.6	MSR_OSTM — Module Standby Register for OSTM.....	848
16.3.7	MSR_WDT — Module Standby Register for WDTB.....	849
16.3.8	MSR_SCI — Module Standby Register for SCI3.....	850
16.3.9	MSR_HSSPI — Module Standby Register for HSSPI	851
16.3.10	MSR_RLIN — Module Standby Register for RLIN3	852
16.3.11	MSR_CSIH — Module Standby Register for CSIH	853
16.3.12	MSR_RHSIF — Module Standby Register for RHSIF	854
16.3.13	MSR_RHSB — Module Standby Register for RHSB.....	855
16.3.14	MSR_CAN — Module Standby Register for CAN	856
16.3.15	MSR_FLXA — Module Standby Register for FlexRay	857
16.3.16	MSR_ETHER — Module Standby Register for Ethernet controller	858
16.3.17	MSR_RSENT — Module Standby Register for RSENT	859
16.3.18	MSR_PSI5S — Module Standby Register for PSI5S	861
16.3.19	MSR_DFE — Module Standby Register for DFE	862
16.3.20	MSR_DAD — Module Standby Register for DS-ADC	863
16.3.21	MSR_CAD — Module Standby Register for Cyclic-ADC.....	864
16.3.22	MSR_SAD — Module Standby Register for SAR-ADC.....	865
16.3.23	MSR_ENCA — Module Standby Register for ENCA.....	866
16.3.24	MSRKCPROT — Module Standby Register Key Code Protection Register	867
Section 17	Clocked Serial Interface H (CSIH)	868
17.1	Features of CSIH.....	868
17.1.1	Units and Channels.....	868
17.1.2	Register Base Address	869

17.1.3	Clock Supply	869
17.1.4	Interrupt Requests.....	870
17.1.5	Reset Sources.....	871
17.1.6	External Input/Output Signals	872
17.1.7	Data Consistency Check.....	874
17.1.8	Notes on Pin Combinations	875
17.2	Overview.....	876
17.2.1	Functional Overview.....	876
17.2.2	Functional Overview Description	877
17.2.3	Block Diagram.....	878
17.3	Registers.....	879
17.3.1	List of Registers	879
17.3.2	CSIHnCTL0 — CSIHn Control Register 0	882
17.3.3	CSIHnCTL1 — CSIHn Control Register 1	883
17.3.4	CSIHnCTL2 — CSIHn Control Register 2	886
17.3.5	CSIHnSTR0 — CSIHn Status Register 0	888
17.3.6	CSIHnSTCR0 — CSIHn Status Clear Register 0	890
17.3.7	CSIHnCFGx — CSIHn Configuration Register x (x = 0 to 5).....	891
17.3.8	CSIHnTX0W — CSIHn Transmit Data Register 0 for Word Access	896
17.3.9	CSIHnTX0H — CSIHn Transmit Data Register 0 for Half Word Access.....	898
17.3.10	CSIHnRX0W — CSIHn Receive Data Register 0 for Word Access	899
17.3.11	CSIHnRX0H — CSIHn Receive Data Register 0 for Half Word Access	900
17.3.12	CSIHnBRSy — CSIHn Baud Rate Setting Register y (y = 0 to 3).....	901
17.3.13	CSIHnEIE — CSIHn Interrupt Control Register.....	902
17.3.14	CSIHnECTL0 — CSIHn Extended Communication Control Register 0	903
17.3.15	CSIHnECTL1 — CSIHn Extended Communication Control Register 1	904
17.3.16	CSIHnETXDAm — CSIHn Extended Communication Transmit Data Register m (m = 0 to 15).....	906
17.3.17	CSIHnERXDAm — CSIHn Extended Communication Receive Data Register m (m = 0 to 15).....	907
17.3.18	CSIHnESTT — CSIHn Extended Communication Stop Trigger Register	908
17.3.19	CSIHnESTF — CSIHn Extended Communication Status Register	909
17.3.20	List of Cautions	910
17.4	Interrupt Sources	913
17.4.1	Overview	913
17.4.2	Interrupt Delay.....	914
17.4.3	INTCSIHTIC (Communication Status Interrupt).....	915
17.4.3.1	INTCSIHTIC in Direct Access Mode.....	915
17.4.3.2	INTCSIHTIC in Job Mode	917
17.4.4	INTCSIHTIR (Reception Status Interrupt)	918
17.4.4.1	INTCSIHTIR in Direct Access Mode.....	918
17.4.5	INTCSIHTIRE (Communication Error Interrupt)	919
17.4.6	INTCSIHTIJC (Job Completion Interrupt)	920

17.5	Operation	921
17.5.1	Operating Modes (Master/Slave)	921
17.5.1.1	Master Mode	921
17.5.1.2	Slave Mode	922
17.5.2	Master/Slave Connections	923
17.5.2.1	One Master and One Slave	923
17.5.2.2	One Master and Multiple Slaves	923
17.5.3	Chip Selection (CS) Features	924
17.5.3.1	Configuration Registers	924
17.5.3.2	CS Example	926
17.5.3.3	Job Concept	927
17.5.4	Details of Chip Select Timing	928
17.5.4.1	Changing the Clock Phase	928
17.5.4.2	Changing the Data Phase	930
17.5.5	Transmission Clock Selection	931
17.5.5.1	Transfer Clock Frequency Calculation	931
17.5.5.2	Transfer Clock Frequency Upper and Lower Limits	932
17.5.6	Data Transfer Modes	932
17.5.6.1	Transmit-Only Mode	932
17.5.6.2	Receive-Only Mode	932
17.5.6.3	Transmit/Receive Mode	932
17.5.6.4	Summary	932
17.5.7	Data Length Selection	933
17.5.7.1	Data Length from 2 to 16 Bits	933
17.5.7.2	Data Length Greater than 16 Bits	934
17.5.8	Serial Data Direction Selection	936
17.5.9	Slave Select (SS) Function	937
17.5.9.1	Communication Timing Using SS Function	937
17.5.9.2	CSIHTSSO Operation	938
17.5.10	Handshake Function	939
17.5.10.1	Slave Mode	939
17.5.10.2	Master Mode	941
17.5.11	Error Detection	942
17.5.11.1	Data Consistency Check	942
17.5.11.2	Parity Check	944
17.5.11.3	Overrun Error	945
17.5.12	Loop-Back Mode	946
17.5.13	Enforced Chip Select Idle Setting	947
17.5.14	Extended Communication Function	948
17.5.14.1	Data Transfer Direction Selection Function	948
17.5.14.2	Extended Communication Stop Function	951
17.5.14.3	Extended Communication Status	952
17.5.14.4	Extended Interrupt Control Function	954
17.6	Operating Procedures	955
17.6.1	Procedures in Direct Access Mode	955
17.6.1.1	Transmit/Receive in Master Mode when Job Mode Is Disabled	955
17.6.1.2	Transmit/Receive in Master Mode when Job Mode is Enabled	957
17.6.2	Procedures in Extended Communication Function Mode	959

17.6.2.1	Trigger to Start Extended Communication.....	965
17.6.2.2	Storage Address of Received Data in Extended Communication	968
Section 18	Serial Communication Interface 3 (SCI3).....	969
18.1	Features of SCI3.....	969
18.1.1	Units and Channels.....	969
18.1.2	Register Base Address	969
18.1.3	Clock Supply	970
18.1.4	Interrupt Requests.....	970
18.1.5	Reset Sources.....	970
18.1.6	External Input / Output Signals	971
18.2	Overview.....	972
18.2.1	Functional Overview.....	972
18.2.2	Serial Communication Modes	973
18.2.3	Block Diagram.....	974
18.3	Registers.....	975
18.3.1	List of Registers	975
18.3.2	SCI3nRSR — Receive Shift Register	976
18.3.3	SCI3nRDR — Receive Data Register.....	976
18.3.4	SCI3nTDR — Transmit Data Register	976
18.3.5	SCI3nTSR — Transmit Shift Register.....	976
18.3.6	SCI3nSMR — Serial Mode Register.....	977
18.3.7	SCI3nSCR — Serial Control Register.....	978
18.3.8	SCI3nSSR — Serial Status Register	980
18.3.9	SCI3nSCMR — Serial Transfer Format Register	982
18.3.10	SCI3nSEMR — Serial Extended Mode Register	983
18.3.11	SCI3nBRR — Bit Rate Register.....	984
18.3.12	SCI3nMDDR — Modulation Duty Register	987
18.4	Operation in Asynchronous Mode	988
18.4.1	Transmission / Reception Format.....	989
18.4.2	Receive Data Sampling Timing and Reception Margin	990
18.4.3	Clock	991
18.4.4	Double-Speed Operation	991
18.4.5	SCI3 Initialization (Asynchronous Mode).....	992
18.4.6	Serial Data Transmission (Asynchronous Mode)	993
18.4.7	Serial Data Reception (Asynchronous Mode).....	998
18.5	Multi-Processor Communication Function.....	1002
18.5.1	Overview and Sample Connection.....	1002
18.5.2	Multi-Processor Serial Data Transmission.....	1003
18.5.3	Multi-Processor Serial Data Reception	1004
18.6	Operation in Clock Synchronous Mode	1008
18.6.1	Clock	1008
18.6.2	SCI3 Initialization (Clock Synchronous Mode).....	1009

18.6.3	Serial Data Transmission (Clock Synchronous Mode)	1010
18.6.4	Serial Data Reception (Clock Synchronous Mode)	1013
18.6.5	Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)	1015
18.7	Bit Rate Modulation Function	1017
18.8	Interrupt Sources	1018
18.9	Usage Notes	1020
18.9.1	Break Detection and Processing	1020
18.9.2	Mark State and Production of Breaks	1020
18.9.3	Receive Error Flags and Transmit Operations in Clock Synchronous Mode	1020
18.9.4	Relationship between Writing to SCI3nTDR and the TDRE Flag	1020
18.9.5	Restrictions on Using an External Clock for Transmission in Clock Synchronous Mode	1021
18.9.6	External Clock Input in Clock Synchronous Mode	1021
Section 19	LIN/UART Interface (RLIN3)	1022
19.1	Features of RLIN3	1022
19.1.1	Units and Channels	1022
19.1.2	Register Base Addresses	1023
19.1.3	Clock Supply	1023
19.1.4	Interrupt Requests	1024
19.1.5	Reset Sources	1025
19.1.6	External Input/Output Signals	1025
19.2	Overview	1026
19.2.1	Functional Overview	1026
19.2.2	Block Diagram	1029
19.2.3	Terms Used in Block Diagram	1029
19.3	Registers	1030
19.3.1	List of Registers	1030
19.3.2	LIN Master Related Registers	1031
19.3.2.1	RLN3nLWBR — LIN Wake-Up Baud Rate Select Register	1031
19.3.2.2	RLN3nLBRP0 — LIN Baud Rate Prescaler 0 Register	1032
19.3.2.3	RLN3nLBRP1 — LIN Baud Rate Prescaler 1 Register	1033
19.3.2.4	RLN3nLSTC — LIN Self-Test Control Register	1034
19.3.2.5	RLN3nLMD — LIN Mode Register	1035
19.3.2.6	RLN3nLBFC — LIN Break Field Configuration Register	1037
19.3.2.7	RLN3nLSC — LIN Space Configuration Register	1038
19.3.2.8	RLN3nLWUP — LIN Wake-Up Configuration Register	1039
19.3.2.9	RLN3nLIE — LIN Interrupt Enable Register	1040
19.3.2.10	RLN3nLEDE — LIN Error Detection Enable Register	1042
19.3.2.11	RLN3nLCUC — LIN Control Register	1044
19.3.2.12	RLN3nLTRC — LIN Transmission Control Register	1045
19.3.2.13	RLN3nLMST — LIN Mode Status Register	1046
19.3.2.14	RLN3nLST — LIN Status Register	1047
19.3.2.15	RLN3nLEST — LIN Error Status Register	1049
19.3.2.16	RLN3nLDFC — LIN Data Field Configuration Register	1051
19.3.2.17	RLN3nLIDB — LIN ID Buffer Register	1053

19.3.2.18	RLN3nLCBR — LIN Checksum Buffer Register	1054
19.3.2.19	RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)	1055
19.3.3	LIN Slave Related Registers	1056
19.3.3.1	RLN3nLWBR — LIN Wake-Up Baud Rate Select Register	1056
19.3.3.2	RLN3nLBRP01 — LIN Baud Rate Prescaler 01 Register	1058
19.3.3.3	RLN3nLSTC — LIN Self-Test Control Register	1059
19.3.3.4	RLN3nLMD — LIN Mode Register	1060
19.3.3.5	RLN3nLBFC — LIN Break Field Configuration Register	1061
19.3.3.6	RLN3nLSC — LIN Space Configuration Register	1062
19.3.3.7	RLN3nLWUP — LIN Wake-Up Configuration Register	1063
19.3.3.8	RLN3nLIE — LIN Interrupt Enable Register	1064
19.3.3.9	RLN3nLEDE — LIN Error Detection Enable Register	1066
19.3.3.10	RLN3nLCUC — LIN Control Register	1068
19.3.3.11	RLN3nLTRC — LIN Transmission Control Register	1069
19.3.3.12	RLN3nLMST — LIN Mode Status Register	1071
19.3.3.13	RLN3nLST — LIN Status Register	1072
19.3.3.14	RLN3nLEST — LIN Error Status Register	1074
19.3.3.15	RLN3nLDFC — LIN Data Field Configuration Register	1076
19.3.3.16	RLN3nLIDB — LIN ID Buffer Register	1078
19.3.3.17	RLN3nLCBR — LIN Checksum Buffer Register	1079
19.3.3.18	RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)	1080
19.3.4	UART Related Registers	1081
19.3.4.1	RLN3nLWBR — UART Wake-Up Baud Rate Select Register	1081
19.3.4.2	RLN3nLBRP01 — UART Baud Rate Prescaler 01 Register	1082
19.3.4.3	RLN3nLMD — UART Mode Register	1083
19.3.4.4	RLN3nLBFC — UART Configuration Register	1084
19.3.4.5	RLN3nLSC — UART Space Configuration Register	1086
19.3.4.6	RLN3nLEDE — UART Error Detection Enable Register	1087
19.3.4.7	RLN3nLCUC — UART Control Register	1089
19.3.4.8	RLN3nLTRC — UART Transmission Control Register	1090
19.3.4.9	RLN3nLMST — UART Mode Status Register	1091
19.3.4.10	RLN3nLST — UART Status Register	1092
19.3.4.11	RLN3nLEST — UART Error Status Register	1094
19.3.4.12	RLN3nLDFC — UART Data Field Configuration Register	1096
19.3.4.13	RLN3nLIDB — UART ID Buffer Register	1097
19.3.4.14	RLN3nLUDB0 — UART Data Buffer 0 Register	1098
19.3.4.15	RLN3nLDBRb — UART Data Buffer b Register (b = 1 to 8)	1099
19.3.4.16	RLN3nLUOER — UART Operation Enable Register	1100
19.3.4.17	RLN3nLUOR1 — UART Option Register 1	1101
19.3.4.18	RLN3nLUTDR — UART Transmission Data Register	1103
19.3.4.19	RLN3nLURDR — UART Reception Data Register	1104
19.3.4.20	RLN3nLUWTDR — UART Wait Transmission Data Register	1105
19.4	Interrupt Sources	1106
19.5	Operation	1107
19.5.1	Modes	1107
19.5.2	LIN Reset Mode	1109
19.5.3	LIN Mode	1110
19.5.3.1	LIN Master Mode	1113
19.5.3.2	LIN Slave Mode	1117
19.5.3.3	Data Transmission/Reception	1125
19.5.3.4	Transmission/Reception Data Buffering	1127

19.5.3.5	Wake-up Transmission/Reception	1130
19.5.3.6	Status	1132
19.5.3.7	Error Status	1134
19.5.4	UART Mode	1138
19.5.4.1	Transmission	1138
19.5.4.2	Reception	1147
19.5.4.3	Expansion Bits	1149
19.5.4.4	Status	1153
19.5.4.5	Error Status	1154
19.5.5	LIN Self-Test Mode	1155
19.5.5.1	Transitioning to LIN Self-Test Mode	1157
19.5.5.2	Transmission in LIN Master Self-Test Mode	1158
19.5.5.3	Reception in LIN Master Self-Test Mode	1160
19.5.5.4	Transmission in LIN Slave Self-Test Mode	1162
19.5.5.5	Reception in LIN Slave Self-Test Mode	1164
19.5.5.6	Terminating LIN Self-Test Mode	1166
19.5.6	Baud Rate Generator	1167
19.5.6.1	LIN Master Mode	1168
19.5.6.2	LIN Slave Mode	1169
19.5.6.3	UART Mode	1170
19.5.7	Noise Filter	1171
Section 20	CANFD Interface (RS-CANFD)	1173
20.1	Features of RS-CANFD	1173
20.1.1	Units and Channels	1173
20.1.2	Register Base Address	1175
20.1.3	Clock Supply	1175
20.1.4	Interrupt Requests	1176
20.1.5	Reset Sources	1178
20.1.6	External Input/Output Signals	1178
20.2	Overview	1179
20.2.1	Functional Outline	1179
20.2.2	Interface Modes	1181
20.2.3	CAN FD Protocol	1181
20.2.4	Block Diagram	1182
20.3	Registers (Classical CAN Mode)	1183
20.3.1	List of Registers	1183
20.3.2	Details of Interface Mode-Related Registers	1188
20.3.2.1	RSCANnGRMCFG — Global Interface Mode Select Register	1188
20.3.3	Details of Channel-Related Registers	1189
20.3.3.1	RSCANnCmCFG — Channel m Configuration Register	1189
20.3.3.2	RSCANnCmCTR — Channel m Control Register	1191
20.3.3.3	RSCANnCmSTS — Channel m Status Register	1196
20.3.3.4	RSCANnCmERFL — Channel m Error Flag Register	1198
20.3.4	Details of Global-Related Registers	1202
20.3.4.1	RSCANnGCFG — Global Configuration Register	1202
20.3.4.2	RSCANnGCTR — Global Control Register	1205

20.3.4.3	RSCANnGSTS — Global Status Register	1207
20.3.4.4	RSCANnGERFL — Global Error Flag Register.....	1208
20.3.4.5	RSCANnGTSC — Global Timestamp Counter Register	1210
20.3.4.6	RSCANnGTINTSTS0 — Global TX Interrupt Status Register 0.....	1211
20.3.4.7	RSCANnGTINTSTS1 — Global TX Interrupt Status Register 1.....	1214
20.3.4.8	RSCANnGFDCFG — Global FD Configuration Register	1216
20.3.5	Details of Receive Rule-Related Registers.....	1217
20.3.5.1	RSCANnGAFLECTR — Receive Rule Entry Control Register	1217
20.3.5.2	RSCANnGAFLCFG0 — Receive Rule Configuration Register 0	1218
20.3.5.3	RSCANnGAFLCFG1 — Receive Rule Configuration Register 1	1219
20.3.5.4	RSCANnGAFLIDj — Receive Rule ID Register j	1220
20.3.5.5	RSCANnGAFLMj — Receive Rule Mask Register j	1221
20.3.5.6	RSCANnGAFLP0_j — Receive Rule Pointer 0 Register j.....	1222
20.3.5.7	RSCANnGAFLP1_j — Receive Rule Pointer 1 Register j.....	1224
20.3.6	Details of Receive Buffer-Related Registers	1225
20.3.6.1	RSCANnRMNB — Receive Buffer Number Register	1225
20.3.6.2	RSCANnRMNDy — Receive Buffer New Data Register y	1226
20.3.6.3	RSCANnRMIDq — Receive Buffer ID Register q.....	1227
20.3.6.4	RSCANnRMPTRq — Receive Buffer Pointer Register q	1228
20.3.6.5	RSCANnRMDf0_q — Receive Buffer Data Field 0 Register q	1229
20.3.6.6	RSCANnRMDf1_q — Receive Buffer Data Field 1 Register q	1230
20.3.7	Details of Receive FIFO Buffer-Related Registers	1231
20.3.7.1	RSCANnRFCCx — Receive FIFO Buffer Configuration and Control Register x .	1231
20.3.7.2	RSCANnRFSTsx — Receive FIFO Buffer Status Register x.....	1233
20.3.7.3	RSCANnRFPCTRx — Receive FIFO Buffer Pointer Control Register x.....	1235
20.3.7.4	RSCANnRFIDx — Receive FIFO Buffer Access ID Register x.....	1236
20.3.7.5	RSCANnRFPTRx — Receive FIFO Buffer Access Pointer Register x	1237
20.3.7.6	RSCANnRFDF0_x — Receive FIFO Buffer Access Data Field 0 Register x.....	1238
20.3.7.7	RSCANnRFDF1_x — Receive FIFO Buffer Access Data Field 1 Register x.....	1239
20.3.8	Details of Transmit/Receive FIFO Buffer-Related Registers	1240
20.3.8.1	RSCANnCFCCk — Transmit/Receive FIFO Buffer Configuration and Control Register k.....	1240
20.3.8.2	RSCANnCFSTSk — Transmit/Receive FIFO Buffer Status Register k	1244
20.3.8.3	RSCANnCFPCTRk — Transmit/Receive FIFO Buffer Pointer Control Register k.....	1247
20.3.8.4	RSCANnCFIDk — Transmit/Receive FIFO Buffer Access ID Register k.....	1248
20.3.8.5	RSCANnCFPTRk — Transmit/Receive FIFO Buffer Access Pointer Register k .	1250
20.3.8.6	RSCANnCFDF0_k — Transmit/Receive FIFO Buffer Access Data Field 0 Register k.....	1252
20.3.8.7	RSCANnCFDF1_k — Transmit/Receive FIFO Buffer Access Data Field 1 Register k.....	1253
20.3.9	Details of FIFO Status-Related Registers.....	1254
20.3.9.1	RSCANnFESTS — FIFO Empty Status Register	1254
20.3.9.2	RSCANnFFSTS — FIFO Full Status Register.....	1256
20.3.9.3	RSCANnFMSTS — FIFO Message Lost Status Register	1258
20.3.9.4	RSCANnRFISTS — Receive FIFO Buffer Interrupt Flag Status Register.....	1260
20.3.9.5	RSCANnCFRISTS — Transmit/Receive FIFO Buffer Receive Interrupt Flag Status Register	1261
20.3.9.6	RSCANnCFTISTS — Transmit/Receive FIFO Buffer Transmit Interrupt Flag Status Register	1262
20.3.10	Details of Transmit Buffer-Related Registers	1263
20.3.10.1	RSCANnTMCp — Transmit Buffer Control Register p.....	1263

20.3.10.2	RSCANnTMSTSp — Transmit Buffer Status Register p	1265
20.3.10.3	RSCANnTMIDp — Transmit Buffer ID Register p	1267
20.3.10.4	RSCANnTMPTRp — Transmit Buffer Pointer Register p.....	1268
20.3.10.5	RSCANnTMDF0_p — Transmit Buffer Data Field 0 Register p.....	1269
20.3.10.6	RSCANnTMDF1_p — Transmit Buffer Data Field 1 Register p.....	1270
20.3.10.7	RSCANnTMIECy — Transmit Buffer Interrupt Enable Configuration Register y .	1271
20.3.11	Details of Transmit Buffer Status-Related Registers	1273
20.3.11.1	RSCANnTMTRSTSy — Transmit Buffer Transmit Request Status Register y	1273
20.3.11.2	RSCANnTMTARSTSy — Transmit Buffer Transmit Abort Request Status Register y.....	1275
20.3.11.3	RSCANnTMTCASTSy — Transmit Buffer Transmit Complete Status Register y ..	1277
20.3.11.4	RSCANnTMTASTSy — Transmit Buffer Transmit Abort Status Register y	1279
20.3.12	Details of Transmit Queue-Related Registers	1281
20.3.12.1	RSCANnTXQCCm — Transmit Queue Configuration and Control Register m ...	1281
20.3.12.2	RSCANnTXQSTSm — Transmit Queue Status Register m.....	1283
20.3.12.3	RSCANnTXQPCTRM — Transmit Queue Pointer Control Register m.....	1285
20.3.13	Details of Transmit History-Related Registers	1286
20.3.13.1	RSCANnTHLCCm — Transmit History List Configuration and Control Register m.....	1286
20.3.13.2	RSCANnTHLSTSm — Transmit History List Status Register m	1288
20.3.13.3	RSCANnTHLPCTRM — Transmit History List Pointer Control Register m	1290
20.3.13.4	RSCANnTHLACCm — Transmit History List Access Register m	1291
20.3.14	Details of Test-Related Registers	1292
20.3.14.1	RSCANnGTSTCFG — Global Test Configuration Register	1292
20.3.14.2	RSCANnGTSTCTR — Global Test Control Register	1294
20.3.14.3	RSCANnGLOCKK — Global Lock Key Register	1295
20.3.14.4	RSCANnRPGACCr — RAM Test Page Access Register r	1296
20.4	Registers (CAN FD Mode).....	1297
20.4.1	List of Registers	1297
20.4.2	Details of Interface Mode-Related Registers	1303
20.4.2.1	RSCFDnCFDGRMCFG — Global Interface Mode Select Register	1303
20.4.3	Details of Channel-Related Registers.....	1304
20.4.3.1	RSCFDnCFDCmNCFG — Channel m Nominal Bit Rate Configuration Register	1304
20.4.3.2	RSCFDnCFDCmCTR — Channel m Control Register.....	1306
20.4.3.3	RSCFDnCFDCmSTS — Channel m Status Register.....	1311
20.4.3.4	RSCFDnCFDCmERFL — Channel m Error Flag Register.....	1313
20.4.3.5	RSCFDnCFDCmDCFG — Channel m Data Bit Rate Configuration Register.....	1317
20.4.3.6	RSCFDnCFDCmFDCFG — Channel m CAN FD Configuration Register	1320
20.4.3.7	RSCFDnCFDCmFDCTR — Channel m CAN FD Control Register.....	1324
20.4.3.8	RSCFDnCFDCmFDSTS — Channel m CAN FD Status Register	1325
20.4.3.9	RSCFDnCFDCmFDCRC — Channel m CAN FD CRC Register	1327
20.4.4	Details of Global-Related Registers.....	1328
20.4.4.1	RSCFDnCFDGCFG — Global Configuration Register.....	1328
20.4.4.2	RSCFDnCFDGCTR — Global Control Register.....	1331
20.4.4.3	RSCFDnCFDGSTS — Global Status Register	1333
20.4.4.4	RSCFDnCFDGERFL — Global Error Flag Register	1334
20.4.4.5	RSCFDnCFDGTSC — Global Timestamp Counter Register.....	1336
20.4.4.6	RSCFDnCFDGTINTSTS0 — Global TX Interrupt Status Register 0	1337
20.4.4.7	RSCFDnCFDGTINTSTS1 — Global TX Interrupt Status Register 1	1340
20.4.4.8	RSCFDnCFDGFDCFG — Global FD Configuration Register	1342

20.4.5	Details of Receive Rule-Related Registers	1343
20.4.5.1	RSCFDnCFDGAFLECTR — Receive Rule Entry Control Register	1343
20.4.5.2	RSCFDnCFDGAFLCFG0 — Receive Rule Configuration Register 0	1344
20.4.5.3	RSCFDnCFDGAFLCFG1 — Receive Rule Configuration Register 1	1345
20.4.5.4	RSCFDnCFDGAFLIDj — Receive Rule ID Register j	1346
20.4.5.5	RSCFDnCFDGAFLMj — Receive Rule Mask Register j	1347
20.4.5.6	RSCFDnCFDGAFLP0_j — Receive Rule Pointer 0 Register j	1348
20.4.5.7	RSCFDnCFDGAFLP1_j — Receive Rule Pointer 1 Register j	1350
20.4.6	Details of Receive Buffer-Related Registers	1351
20.4.6.1	RSCFDnCFDRMNB — Receive Buffer Number Register	1351
20.4.6.2	RSCFDnCFDRMNDy — Receive Buffer New Data Register y	1352
20.4.6.3	RSCFDnCFDRMIDq — Receive Buffer ID Register q	1353
20.4.6.4	RSCFDnCFDRMPTRq — Receive Buffer Pointer Register q	1354
20.4.6.5	RSCFDnCFDRMFDSTSq — Receive Buffer CAN FD Status Register q	1356
20.4.6.6	RSCFDnCFDRMDFb_q — Receive Buffer Data Field b Register q	1357
20.4.7	Details of Receive FIFO Buffer-Related Registers	1358
20.4.7.1	RSCFDnCFDRFCCx — Receive FIFO Buffer Configuration and Control Register x	1358
20.4.7.2	RSCFDnCFDRFSTSx — Receive FIFO Buffer Status Register x	1360
20.4.7.3	RSCFDnCFDRFPCTRx — Receive FIFO Buffer Pointer Control Register x	1362
20.4.7.4	RSCFDnCFDRFIDx — Receive FIFO Buffer Access ID Register x	1363
20.4.7.5	RSCFDnCFDRFPTRx — Receive FIFO Buffer Access Pointer Register x	1364
20.4.7.6	RSCFDnCFDRFFDSTSx — Receive FIFO CAN FD Status Register x	1365
20.4.7.7	RSCFDnCFDRFDFd_x — Receive FIFO Buffer Access Data Field d Register x	1366
20.4.8	Details of Transmit/Receive FIFO Buffer Related Registers	1367
20.4.8.1	RSCFDnCFDCFCCK — Transmit/Receive FIFO Buffer Configuration and Control Register k	1367
20.4.8.2	RSCFDnCFDCFSTS k — Transmit/Receive FIFO Buffer Status Register k	1371
20.4.8.3	RSCFDnCFDCFPCTRk — Transmit/Receive FIFO Buffer Pointer Control Register k	1374
20.4.8.4	RSCFDnCFDCFIDk — Transmit/Receive FIFO Buffer Access ID Register k	1376
20.4.8.5	RSCFDnCFDCFPTRk — Transmit/Receive FIFO Buffer Access Pointer Register k	1378
20.4.8.6	RSCFDnCFDCFFDCSTS k — Transmit/Receive FIFO CAN FD Configuration/Status Register k	1380
20.4.8.7	RSCFDnCFDCFDf_k — Transmit/Receive FIFO Buffer Access Data Field d Register k	1382
20.4.9	Details of FIFO Status-Related Registers	1383
20.4.9.1	RSCFDnCFDFESTS — FIFO Empty Status Register	1383
20.4.9.2	RSCFDnCFDFFSTS — FIFO Full Status Register	1385
20.4.9.3	RSCFDnCFDFMSTS — FIFO Message Lost Status Register	1387
20.4.9.4	RSCFDnCFDRFISTS — Receive FIFO Buffer Interrupt Flag Status Register	1389
20.4.9.5	RSCFDnCFDCFRISTS — Transmit/Receive FIFO Buffer Receive Interrupt Flag Status Register	1390
20.4.9.6	RSCFDnCFDCFTISTS — Transmit/Receive FIFO Buffer Transmit Interrupt Flag Status Register	1391
20.4.10	Details of FIFO DMA-Related Registers	1392
20.4.10.1	RSCFDnCFDCDTCT — DMA Enable Register	1392
20.4.10.2	RSCFDnCFDCDTSTS — DMA Status Register	1394
20.4.11	Details of Transmit Buffer-Related Registers	1396
20.4.11.1	RSCFDnCFDTMCP — Transmit Buffer Control Register p	1396

20.4.11.2	RSCFDnCFDTMSTSp — Transmit Buffer Status Register p.....	1398
20.4.11.3	RSCFDnCFDTMIDp — Transmit Buffer ID Register p.....	1400
20.4.11.4	RSCFDnCFDTMPTRp — Transmit Buffer Pointer Register p.....	1401
20.4.11.5	RSCFDnCFDTMFDCTRp — Transmit Buffer CAN FD Configuration Register p.....	1403
20.4.11.6	RSCFDnCFDTMDFb_p — Transmit Buffer Data Field b Register p.....	1404
20.4.11.7	RSCFDnCFDTMIECy — Transmit Buffer Interrupt Enable Configuration Register y.....	1405
20.4.12	Details of Transmit Buffer Status-Related Registers	1407
20.4.12.1	RSCFDnCFDTMTRSTSy — Transmit Buffer Transmit Request Status Register y.....	1407
20.4.12.2	RSCFDnCFDTMTARSTSy — Transmit Buffer Transmit Abort Request Status Register y.....	1409
20.4.12.3	RSCFDnCFDTMTCSTSy — Transmit Buffer Transmit Complete Status Register y.....	1411
20.4.12.4	RSCFDnCFDTMTASTSy — Transmit Buffer Transmit Abort Status Register y..	1413
20.4.13	Details of Transmit Queue-Related Registers	1415
20.4.13.1	RSCFDnCFDTXQCCm — Transmit Queue Configuration and Control Register m.....	1415
20.4.13.2	RSCFDnCFDTXQSTSm — Transmit Queue Status Register m	1417
20.4.13.3	RSCFDnCFDTXQPCTRM — Transmit Queue Pointer Control Register m.....	1419
20.4.14	Details of Transmit History-Related Registers	1420
20.4.14.1	RSCFDnCFDTHLCCm — Transmit History List Configuration and Control Register m.....	1420
20.4.14.2	RSCFDnCFDTHLSTSm — Transmit History List Status Register m.....	1422
20.4.14.3	RSCFDnCFDTHLPCTRM — Transmit History List Pointer Control Register m..	1424
20.4.14.4	RSCFDnCFDTHLACCm — Transmit History List Access Register m.....	1425
20.4.15	Details of Test-Related Registers	1426
20.4.15.1	RSCFDnCFDGTSTCFG — Global Test Configuration Register.....	1426
20.4.15.2	RSCFDnCFDGTSTCTR — Global Test Control Register.....	1428
20.4.15.3	RSCFDnCFDGLOCKK — Global Lock Key Register.....	1429
20.4.15.4	RSCFDnCFDRPGACCr — RAM Test Page Access Register r.....	1430
20.5	Interrupt Sources and DMA Trigger.....	1431
20.5.1	Interrupt Sources.....	1431
20.5.2	DMA Triger (Only in CAN FD Mode).....	1434
20.6	CAN Modes	1435
20.6.1	Global Modes	1435
20.6.1.1	Global Stop Mode	1436
20.6.1.2	Global Reset Mode	1436
20.6.1.3	Global Test Mode.....	1437
20.6.1.4	Global Operating Mode.....	1437
20.6.2	Channel Modes	1438
20.6.2.1	Channel Stop Mode	1439
20.6.2.2	Channel Reset Mode	1439
20.6.2.3	Channel Halt Mode	1440
20.6.2.4	Channel Communication Mode	1440
20.6.2.5	Bus Off State.....	1441
20.6.3	Initializing Registers by Transition to CAN Mode.....	1442
20.7	Reception Functions	1444

20.7.1	Data Processing Using the Receive Rule Table	1444
20.7.1.1	Acceptance Filter Processing	1445
20.7.1.2	DLC Filter Processing	1446
20.7.1.3	Routing Processing	1446
20.7.1.4	Label Addition Processing	1446
20.7.1.5	Mirror Function Processing	1446
20.7.1.6	Timestamp	1447
20.8	Transmission Functions	1448
20.8.1	Transmit Priority Determination	1449
20.8.2	Transmission Using Transmit Buffers	1450
20.8.2.1	Transmit Abort Function	1450
20.8.2.2	One-Shot Transmission Function (Retransmission Disabling Function)	1450
20.8.2.3	Transmit Buffer Merge Mode (Only in CAN FD Mode)	1450
20.8.3	Transmission Using FIFO Buffers	1451
20.8.3.1	Interval Transmission Function	1452
20.8.4	Transmission Using Transmit Queues	1454
20.8.5	Transmit Data Padding (Only in CAN FD Mode)	1454
20.8.6	Transmit History Function	1455
20.9	Gateway Function	1457
20.9.1	CAN-CAN FD Gateway (Only in CAN FD Mode)	1457
20.10	Test Function	1458
20.10.1	Standard Test Mode	1458
20.10.2	Listen-Only Mode	1459
20.10.3	Self-Test Mode (Loopback Mode)	1460
20.10.3.1	Self-Test Mode 0 (External Loopback Mode)	1460
20.10.3.2	Self-Test Mode 1 (Internal Loopback Mode)	1461
20.10.4	Restricted Operation Mode (Only in CAN FD Mode)	1461
20.10.5	RAM Test Mode	1462
20.10.6	Inter-Channel Communication Test	1463
20.10.6.1	CRC Error Test	1464
20.11	RS-CANFD Setting Procedure	1465
20.11.1	Initial Settings	1465
20.11.1.1	Clock Setting	1467
20.11.1.2	Bit Timing Setting	1467
20.11.1.3	Communication Speed Setting	1469
20.11.1.4	Receive Rule Setting	1471
20.11.1.5	Buffer Setting	1472
20.11.1.6	Transmitter Delay Compensation (Only in CAN FD Mode)	1475
20.11.2	Reception Procedure	1476
20.11.2.1	Receive Buffer Reading Procedure	1476
20.11.2.2	FIFO Buffer Reading Procedure	1478
20.11.2.3	FIFO Buffer Reading Procedure by DMA Transfer	1482
20.11.3	Transmission Procedure	1483
20.11.3.1	Procedure for Transmission from Transmit Buffers	1483
20.11.3.2	Procedure for Transmission from Transmit/Receive FIFO Buffers	1488
20.11.3.3	Procedure for Transmission from the Transmit Queue	1492
20.11.3.4	Transmit History Buffer Reading Procedure	1493

20.11.4	Test Settings	1494
20.11.4.1	Self-Test Mode Setting Procedure.....	1494
20.11.4.2	Procedure for Releasing the Protection.....	1494
20.11.4.3	RAM Test Setting Procedure	1496
20.11.4.4	Inter-Channel Communication Test Setting Procedure	1497
20.12	Notes on the RS-CANFD Module.....	1498
Section 21	FlexRay (FLXA)	1500
21.1	Features of FLXA.....	1500
21.1.1	Number of Units	1500
21.1.2	Register Base Address	1501
21.1.3	Clock Supply	1501
21.1.4	Interrupt Requests.....	1502
21.1.5	Reset Sources.....	1503
21.1.6	External Input/Output Signals	1503
21.2	Overview	1504
21.2.1	Functional Overview.....	1504
21.2.2	Block Diagram.....	1505
21.3	Registers.....	1510
21.3.1	List of Registers	1510
21.3.2	FlexRay Operation Register.....	1514
21.3.2.1	FLXAnFROC — FlexRay Operation Control Register.....	1514
21.3.2.2	FLXAnFROS — FlexRay Operation Status Register	1517
21.3.3	Special Registers	1519
21.3.3.1	FLXAnFRLCK — FlexRay Lock Register	1519
21.3.4	Interrupt Registers.....	1520
21.3.4.1	FLXAnFREIR — FlexRay Error Interrupt Register	1520
21.3.4.2	FLXAnFRSIR — FlexRay Status Interrupt Register	1525
21.3.4.3	FLXAnFREILS — FlexRay Error Interrupt Line Select Register.....	1531
21.3.4.4	FLXAnFRSILS — FlexRay Status Interrupt Line Select Register	1533
21.3.4.5	FLXAnFREIES — FlexRay Error Interrupt Enable Set Register	1535
21.3.4.6	FLXAnFREIER — FlexRay Error Interrupt Enable Reset Register	1537
21.3.4.7	FLXAnFRSIES — FlexRay Status Interrupt Enable Set Register	1539
21.3.4.8	FLXAnFRSIER — FlexRay Status Interrupt Enable Reset Register.....	1541
21.3.4.9	FLXAnFRILE — FlexRay Interrupt Line Enable Register.....	1543
21.3.5	FlexRay Timer Registers	1544
21.3.5.1	FLXAnFRT0C — FlexRay Timer 0 Configuration Register	1544
21.3.5.2	FLXAnFRT1C — FlexRay Timer 1 Configuration Register	1546
21.3.5.3	FLXAnFRT2C — FlexRay Timer 2 Configuration Register	1548
21.3.5.4	FLXAnFRSTPW1 — FlexRay Stop Watch Register 1.....	1550
21.3.5.5	FLXAnFRSTPW2 — FlexRay Stop Watch Register 2.....	1553
21.3.6	CC Control Registers	1554
21.3.6.1	FLXAnFRSUCC1 — FlexRay SUC Configuration Register 1	1554
21.3.6.2	FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2	1561
21.3.6.3	FLXAnFRSUCC3 — FlexRay SUC Configuration Register 3	1562
21.3.6.4	FLXAnFRNEMC — FlexRay NEM Configuration Register.....	1563
21.3.6.5	FLXAnFRPRTC1 — FlexRay PRT Configuration Register 1	1564
21.3.6.6	FLXAnFRPRTC2 — FlexRay PRT Configuration Register 2	1567

21.3.6.7	FLXAnFRMHDC — FlexRay MHD Configuration Register	1569
21.3.6.8	FLXAnFRGTUC1 — FlexRay GTU Configuration Register 1	1570
21.3.6.9	FLXAnFRGTUC2 — FlexRay GTU Configuration Register 2	1571
21.3.6.10	FLXAnFRGTUC3 — FlexRay GTU Configuration Register 3	1572
21.3.6.11	FLXAnFRGTUC4 — FlexRay GTU Configuration Register 4	1574
21.3.6.12	FLXAnFRGTUC5 — FlexRay GTU Configuration Register 5	1575
21.3.6.13	FLXAnFRGTUC6 — FlexRay GTU Configuration Register 6	1577
21.3.6.14	FLXAnFRGTUC7 — FlexRay GTU Configuration Register 7	1578
21.3.6.15	FLXAnFRGTUC8 — FlexRay GTU Configuration Register 8	1579
21.3.6.16	FLXAnFRGTUC9 — FlexRay GTU Configuration Register 9	1580
21.3.6.17	FLXAnFRGTUC10 — FlexRay GTU Configuration Register 10	1582
21.3.6.18	FLXAnFRGTUC11 — FlexRay GTU Configuration Register 11	1583
21.3.7	CC Status Registers.....	1585
21.3.7.1	FLXAnFRCCSV — FlexRay CC Status Vector Register.....	1585
21.3.7.2	FLXAnFRCCEV — FlexRay CC Error Vector Register	1589
21.3.7.3	FLXAnFRSCV — FlexRay Slot Counter Value Register	1591
21.3.7.4	FLXAnFRMTCCV — FlexRay Macrotick and Cycle Counter Value Register	1592
21.3.7.5	FLXAnFRRCV — FlexRay Rate Correction Value Register.....	1593
21.3.7.6	FLXAnFROCV — FlexRay Offset Correction Value Register	1594
21.3.7.7	FLXAnFRSFS — FlexRay Sync Frame Status Register	1595
21.3.7.8	FLXAnFRSWNIT — FlexRay Symbol Window and NIT Status Register	1598
21.3.7.9	FLXAnFRACS — FlexRay Aggregated Channel Status Register	1601
21.3.7.10	FLXAnFRESIDm — FlexRay Even Sync ID Register m (m = 1 to 15).....	1605
21.3.7.11	FLXAnFROSIDm — FlexRay Odd Sync ID Register m (m = 1 to 15).....	1607
21.3.7.12	FLXAnFRNMVm — FlexRay Network Management Vector Register m (m = 1 to 3).....	1609
21.3.8	Message Buffer Control Registers	1610
21.3.8.1	FLXAnFRMRC — FlexRay Message RAM Configuration Register	1610
21.3.8.2	FLXAnFRFRF — FlexRay FIFO Rejection Filter Register	1614
21.3.8.3	FLXAnFRFRFM — FlexRay FIFO Rejection Filter Mask Register.....	1616
21.3.8.4	FLXAnFRFCL — FlexRay FIFO Critical Level Register	1617
21.3.9	Message Buffer Status Registers	1618
21.3.9.1	FLXAnFRMHDS — FlexRay Message Handler Status Register	1618
21.3.9.2	FLXAnFRLDTS — FlexRay Last Dynamic Transmit Slot Register	1621
21.3.9.3	FLXAnFRFSR — FlexRay FIFO Status Register	1622
21.3.9.4	FLXAnFRMHDF — FlexRay Message Handler Constraints Flags Register.....	1624
21.3.9.5	FLXAnFRTXRQi — FlexRay Transmission Request Register i (i = 1 to 4).....	1628
21.3.9.6	FLXAnFRNDATi — FlexRay New Data Register i (i = 1 to 4).....	1629
21.3.9.7	FLXAnFRMBSCi — FlexRay Message Buffer Status Changed Register i (i = 1 to 4).....	1630
21.3.10	Input Buffer.....	1631
21.3.10.1	FLXAnFRWRDSx — FlexRay Write Data Section Register x (x = 1 to 64).....	1631
21.3.10.2	FLXAnFRWRHS1 — FlexRay Write Header Section Register 1	1633
21.3.10.3	FLXAnFRWRHS2 — FlexRay Write Header Section Register 2	1635
21.3.10.4	FLXAnFRWRHS3 — FlexRay Write Header Section Register 3	1636
21.3.10.5	FLXAnFRIBCM — FlexRay Input Buffer Command Mask Register.....	1637
21.3.10.6	FLXAnFRIBCR — FlexRay Input Buffer Command Request Register	1639
21.3.11	Output Buffer.....	1641
21.3.11.1	FLXAnFRRDDSx — FlexRay Read Data Section Register x (x = 1 to 64).....	1641
21.3.11.2	FLXAnFRRDHS1 — FlexRay Read Header Section Register 1	1642
21.3.11.3	FLXAnFRRDHS2 — FlexRay Read Header Section Register 2	1644
21.3.11.4	FLXAnFRRDHS3 — FlexRay Read Header Section Register 3.....	1646

21.3.11.5	FLXAnFRMBS — FlexRay Message Buffer Status Register	1648
21.3.11.6	FLXAnFROBCM — FlexRay Output Buffer Command Mask Register	1654
21.3.11.7	FLXAnFROBCR — FlexRay Output Buffer Command Request Register	1656
21.3.12	Data Transfer Control Register	1658
21.3.12.1	FLXAnFRITC — FlexRay Input Transfer Configuration Register	1658
21.3.12.2	FLXAnFROTC — FlexRay Output Transfer Configuration Register	1660
21.3.12.3	FLXAnFRIBA — FlexRay Input Pointer Table Base Address Register	1663
21.3.12.4	FLXAnFRFBA — FlexRay FIFO Pointer Table Base Address Register	1664
21.3.12.5	FLXAnFROBA — FlexRay Output Pointer Table Base Address Register	1665
21.3.12.6	FLXAnFRIQC — FlexRay Input Queue Control Register	1666
21.3.12.7	FLXAnFRUIR — FlexRay User Input Transfer Request Register	1667
21.3.12.8	FLXAnFRUOR — FlexRay User Output Transfer Request Register	1668
21.3.13	Data Transfer Status Register	1669
21.3.13.1	FLXAnFRITS — FlexRay Input Transfer Status Register	1669
21.3.13.2	FLXAnFROTS — FlexRay Output Transfer Status Register	1673
21.3.13.3	FLXAnFRAES — FlexRay Access Error Status Register	1678
21.3.13.4	FLXAnFRAEA — FlexRay Access Error Address Register	1680
21.3.13.5	FLXAnFRDAi — FlexRay Message Data Available Register i (i = 0 to 3)	1681
21.4	Operation	1682
21.4.1	FlexRay Module Operation Control	1682
21.4.1.1	FlexRay Module Enable	1682
21.4.1.2	FlexRay Module Disable	1683
21.4.2	Communication Cycle	1684
21.4.3	Communication Modes	1687
21.4.3.1	Time-triggered Distributed (TT-D)	1687
21.4.4	Clock Synchronization	1688
21.4.4.1	Global Time	1688
21.4.4.2	Local Time	1688
21.4.4.3	Synchronization Process	1688
21.4.5	Error Handling	1690
21.4.5.1	Clock Correction Failed Counter	1690
21.4.5.2	Passive to Active Counter	1691
21.4.5.3	HALT Command	1691
21.4.5.4	FREEZE Command	1691
21.4.6	Communication Controller States	1692
21.4.6.1	Communication Controller State Diagram	1692
21.4.6.2	DEFAULT_CONFIG State	1694
21.4.6.3	CONFIG State	1694
21.4.6.4	READY State	1695
21.4.6.5	WAKEUP State	1696
21.4.6.6	STARTUP State	1701
21.4.6.7	NORMAL_ACTIVE State	1706
21.4.6.8	NORMAL_PASSIVE State	1706
21.4.6.9	HALT State	1707
21.4.7	Network Management	1708
21.4.8	Filtering and Masking	1709
21.4.8.1	Slot Counter Filtering	1709
21.4.8.2	Cycle Counter Filtering	1709
21.4.8.3	Channel ID Filtering	1711
21.4.8.4	FIFO Filtering	1711

21.4.9	Transmit Process	1712
21.4.9.1	Static Segment.....	1712
21.4.9.2	Dynamic Segment.....	1712
21.4.9.3	Transmit Buffers.....	1712
21.4.9.4	Frame Transmission	1713
21.4.9.5	Null Frame Transmission	1714
21.4.10	Receive Process	1715
21.4.10.1	Dedicated Receive Buffers	1715
21.4.10.2	Frame Reception	1715
21.4.10.3	Null Frame Reception	1716
21.4.11	FIFO Function	1717
21.4.11.1	Description	1717
21.4.11.2	Configuration of the FIFO	1718
21.4.11.3	Access to the FIFO	1719
21.4.12	Message Handling	1720
21.4.12.1	Reconfiguration of Message Buffers.....	1720
21.4.12.2	Host Access to Message RAM	1722
21.4.12.3	FlexRay Protocol Controller Access to Message RAM	1728
21.4.13	Message RAM.....	1729
21.4.13.1	Header Partition	1731
21.4.13.2	Data Partition	1735
21.4.13.3	Message Data Integrity Check.....	1736
21.4.13.4	Host Handling of Access Errors.....	1738
21.4.14	Interrupts	1739
21.4.15	Assignment of FlexRay Configuration Parameters.....	1740
21.4.16	Usage of Data Transfer	1742
21.4.16.1	Input Data Transfer.....	1742
21.4.16.2	Output Data Transfer	1751
21.4.16.3	Data Structure Transfer Scheduling	1759
21.4.16.4	Behavior in Case of Data Transfer Access Error.....	1760
21.4.16.5	Behaviors in Case of RAM Read Errors	1762
21.4.17	Byte Alignment.....	1764
21.4.17.1	Little Endian Alignment	1764
21.4.17.2	Big Endian Alignment.....	1766
Section 22	Renesas High Speed Bus (RHSB).....	1767
22.1	Features of RHSB.....	1767
22.1.1	Number of Units	1767
22.1.2	Register Base Addresses	1768
22.1.3	Clock Supply	1768
22.1.4	Interrupt Requests.....	1769
22.1.5	Reset Sources.....	1770
22.1.6	External Input/Output Signals	1771
22.2	Overview.....	1772
22.2.1	Functional Overview.....	1772
22.2.1.1	Downstream Channel Communication Functions	1772
22.2.1.2	Upstream Channel Communication Functions	1773
22.2.1.3	Other Features.....	1773

22.2.1.4	Caution.....	1773
22.2.2	Block Diagram.....	1774
22.3	Registers.....	1775
22.3.1	List of Registers	1775
22.3.2	Legend	1776
22.3.3	Description of the Registers in the Common Controller.....	1777
22.3.3.1	RHSBjGC — Global Configuration Register.....	1777
22.3.3.2	RHSBjMSR — Module Status Register.....	1779
22.3.4	Downstream (Tx) Registers	1781
22.3.4.1	RHSBjDCR — Downstream Configuration Register	1781
22.3.4.2	RHSBjDCR1 — Downstream Configuration Register for Period 1.....	1785
22.3.4.3	RHSBjDEC — Data Element Configuration Register	1786
22.3.4.4	RHSBjSDCi — Slave Device Configuration Register i (i = 0).....	1788
22.3.4.5	RHSBjDEBAm — Data Element Bit Assignment Register m (m = 0 to 3).....	1790
22.3.4.6	RHSBjEBEi — Emergency Bit Enable Register i (i = 0, 1).....	1792
22.3.4.7	RHSBjDTC — Downstream Transmission Control Register	1793
22.3.4.8	RHSBjDCD — Downstream Command Data Register.....	1795
22.3.4.9	RHSBjDDRi — Downstream Data Register i (i = 0, 1).....	1796
22.3.4.10	RHSBjDEDi — Downstream Emergency Data Register i (i = 0, 1).....	1797
22.3.5	Upstream (Rx) Registers	1798
22.3.5.1	RHSBjUCR — Upstream Configuration Register	1798
22.3.5.2	RHSBjUCC — Upstream Channel Configuration Register.....	1800
22.3.5.3	RHSBjUCS — Upstream Channel Selection Register	1802
22.3.5.4	RHSBjUDR — Upstream Data Read Register	1804
22.3.5.5	RHSBjUDi — Upstream Data Register i (i = 0, 1).....	1807
22.3.5.6	RHSBjUSS — Upstream Status Summary Register	1810
22.3.6	Interrupt Registers.....	1812
22.3.6.1	RHSBjIC — Interrupt Control Register	1812
22.3.6.2	RHSBjIS — Interrupt Status Register.....	1814
22.4	Operation	1818
22.4.1	Operation Description	1818
22.4.1.1	RESET State.....	1818
22.4.1.2	CONFIG State.....	1818
22.4.1.3	ACTIVE State.....	1818
22.4.1.4	TEST State	1819
22.4.2	Activation of the RHSB Module (Leaving RESET State).....	1819
22.4.3	Deactivation of the RHSB Module (Entering RESET State).....	1820
22.4.4	Changing between Operation States (ACTIVE, CONFIG, TEST).....	1820
22.4.5	Leaving ACTIVE State without Interrupting Ongoing Transmission	1821
22.5	Operating Procedures.....	1822
22.5.1	Downstream Communication.....	1822
22.5.1.1	Downstream Communication Phases.....	1824
22.5.1.2	Frame Dependent Flagging	1826
22.5.1.3	Downstream Modes	1828
22.5.1.4	Command Frame Insertion Methods in Multi-Period Repetition Mode.....	1835
22.5.1.5	Physical Frame Format.....	1837
22.5.1.6	Data Frame Assembling	1838
22.5.1.7	Command Frame Assembling	1841
22.5.1.8	Emergency Function	1842

22.5.1.9	Downstream Bit Rates	1844
22.5.1.10	Data Update and Data Frame Transmission Request.....	1845
22.5.1.11	Command Frame Transmission Request.....	1846
22.5.2	Upstream Communication	1847
22.5.2.1	Upstream Modes.....	1849
22.5.2.2	Individual Slave Configuration	1850
22.5.2.3	Frame Format Types	1850
22.5.2.4	Frame Storing	1851
22.5.2.5	Upstream Bit Rates.....	1852
22.5.2.6	Update of Decoding Status	1852
22.5.3	Timeout Detection	1853
22.5.3.1	Command Transmission with Remote Data Request.....	1855
22.5.3.2	Timeout Detection Details.....	1855
22.5.4	Test Mode Operation	1856
22.5.4.1	Test Mode Data Generation	1858
22.5.5	Interrupts	1859
22.5.5.1	Downstream Related Interrupts (Transmission).....	1859
22.5.5.2	Upstream Related Interrupts (Reception).....	1859
22.5.5.3	Interrupt Request Details	1860
22.5.6	DMA Capability	1861
22.5.6.1	DMA Usage for Downstream Data Transmission.....	1861
22.5.6.2	DMA Usage for Downstream Command Transmission.....	1863
22.5.6.3	DMA Usage for Upstream Data Reception.....	1864
22.6	Cross Bar (XBAR).....	1865
22.6.1	Overview	1865
22.6.1.1	Number of Units	1865
22.6.1.2	Register Base Addresses	1865
22.6.1.3	Clock Supply	1866
22.6.1.4	Reset Source	1866
22.6.2	Module Configuration	1867
22.6.2.1	RHSB XBAR Configuration.....	1867
22.6.3	Registers	1871
22.6.3.1	List of Registers	1871
22.6.3.2	RHSBGjCRkH — Microsecond Bus Control Register H (ATU-V)	1872
22.6.3.3	RHSBGjCRkL — Microsecond Bus Control Register L (ATU-V)	1881
22.6.3.4	RHSBGjCROk — Microsecond Bus Control Register Out (ATU-V).....	1890
22.6.3.5	RHSBGjCRkH — Microsecond Bus Control Register H (GTM).....	1894
22.6.3.6	RHSBGjCRkL — Microsecond Bus Control Register L (GTM)	1903
22.6.3.7	RHSBGjCROk — Microsecond Bus Control Register Out (GTM).....	1912
22.6.4	Operation	1916
22.6.4.1	List of Bits that can be Selected in XBAR when Connected with RHSB and ATU-V is Used	1916
22.6.4.2	List of Bits that can be Selected in XBAR when Connected with RHSB and GTM is Used	1922
Section 23	Single Edge Nibble Transmission (RSENT).....	1926
23.1	Features of RSENT	1926
23.1.1	Number of Channels	1926
23.1.2	Register Base Addresses	1927

23.1.3	Clock Supply	1928
23.1.4	Interrupts Requests.....	1928
23.1.5	Reset Source	1929
23.1.6	External Input/Output Signals	1930
23.2	Overview.....	1932
23.2.1	Functional Overview.....	1932
23.2.2	Block Diagram.....	1933
23.3	Registers.....	1934
23.3.1	List of Registers	1934
23.3.2	RSENTnTSPC — RSENT Timestamp Register	1935
23.3.3	RSENTnTSC — RSENT Timestamp Counter Register.....	1937
23.3.4	RSENTnCC — RSENT Communication Configuration Register.....	1938
23.3.5	RSENTnBRP — RSENT Baud Rate Prescaler Register.....	1941
23.3.6	RSENTnIDE — RSENT Interrupt / DMA Enable Register.....	1943
23.3.7	RSENTnMDC — RSENT Mode Control Register.....	1945
23.3.8	RSENTnSPCT — RSENT SPC Transmission Register	1946
23.3.9	RSENTnMST — RSENT Mode Status Register.....	1947
23.3.10	RSENTnCS — RSENT Communication Status Register	1948
23.3.11	RSENTnCSC — RSENT Communication Status Clear Register.....	1952
23.3.12	RSENTnSRTS — RSENT Slow Channel Receive Timestamp Register.....	1954
23.3.13	RSENTnSRXD — RSENT Slow Channel Receive Data Register	1955
23.3.14	RSENTnCPL — RSENT Calibration Pulse Length Register	1957
23.3.15	RSENTnML — RSENT Message Length Register.....	1958
23.3.16	RSENTnFRTS — RSENT Fast Channel Receive Timestamp Register.....	1959
23.3.17	RSENTnFRXD — RSENT Fast Channel Receive Data Register	1960
23.3.18	RSENTnCPLM — RSENT Calibration Pulse Length Mirror Register.....	1962
23.3.19	RSENTnMLM — RSENT Message Length Mirror Register	1963
23.3.20	RSENTnFRTSM — RSENT Fast Channel Receive Timestamp Mirror Register	1964
23.3.21	RSENTnEFRD0 — RSENT Expanded Fast Channel Receive Data Register 0	1965
23.3.22	RSENTnEFRD1 — RSENT Expanded Fast Channel Receive Data Register 1	1967
23.4	Operation	1968
23.4.1	Modes of Operation.....	1968
23.4.1.1	RESET Mode	1969
23.4.1.2	CONFIGURATION Mode.....	1969
23.4.1.3	OPERATION IDLE Mode.....	1969
23.4.1.4	OPERATION ACTIVE Mode.....	1969
23.4.1.5	Register Behavior in Operation Modes.....	1970
23.4.1.6	Mode Status Change Timing	1971
23.4.2	Clock Configuration.....	1972
23.4.2.1	Timestamp	1972
23.4.2.2	Communication Clock Configuration	1974
23.4.3	RSENT Operation	1975
23.4.3.1	Changing Operation Modes.....	1975
23.4.3.2	Message Reception	1977

23.4.4	SPC Function	1992
23.4.4.1	Multiplexing of the RSENTnRX and RSENTnSPCO Pin Functions	1994
23.4.5	Interrupts and Checks	1996
Section 24	Ethernet Controller (ETNC).....	1998
24.1	Features of ETNC.....	1998
24.1.1	Number of Units and Channels.....	1998
24.1.2	Register Addresses.....	1999
24.1.3	Clock Supply	1999
24.1.4	Interrupt Requests.....	1999
24.1.5	Reset Source	1999
24.1.6	External Input/Output Signals	2000
24.2	Overview.....	2003
24.2.1	Functional Overview.....	2003
24.2.2	Block Diagram.....	2004
24.3	Registers.....	2005
24.3.1	List of Registers	2005
24.3.2	ETHERC Registers	2007
24.3.2.1	ETNCnECMR — ETHERC Mode Register.....	2007
24.3.2.2	ETNCnRFLR — Receive Frame Maximum Length Register	2010
24.3.2.3	ETNCnECSR — ETHERC Status Register	2011
24.3.2.4	ETNCnECSIPR — ETHERC Interrupt Enable Register	2013
24.3.2.5	ETNCnPIR — PHY Interface Register.....	2014
24.3.2.6	ETNCnPSR — PHY Status Register.....	2015
24.3.2.7	ETNCnRDMLR — Random Number Generation Counter Limit Setting Register.....	2016
24.3.2.8	ETNCnIPGR — Interpacket Gap Register.....	2017
24.3.2.9	ETNCnAPR — Automatic PAUSE Frame Register	2018
24.3.2.10	ETNCnMPR — Manual PAUSE Frame Register	2019
24.3.2.11	ETNCnRFCF — Received PAUSE Frame Counter	2020
24.3.2.12	ETNCnTPAUSER — PAUSE Frame Retransmit Count Setting Register.....	2021
24.3.2.13	ETNCnTPAUSECR — PAUSE Frame Retransmit Counter Register	2022
24.3.2.14	ETNCnBCFRR — Broadcast Frame Receive Count Setting Register.....	2023
24.3.2.15	ETNCnMAHR — MAC Address Upper Bit Register	2024
24.3.2.16	ETNCnMALR — MAC Address Lower Bit Register.....	2025
24.3.2.17	ETNCnTROCR — Transmit Retry Over Counter Register.....	2026
24.3.2.18	ETNCnCDCR — Late Collision Detect Counter Register.....	2027
24.3.2.19	ETNCnLCCR — Lost Carrier Counter Register	2028
24.3.2.20	ETNCnCNDCCR — Carrier Not Detected Counter Register	2029
24.3.2.21	ETNCnCEFCR — CRC Error Frame Receive Counter Register	2030
24.3.2.22	ETNCnFRECR — Frame Receive Error Counter Register	2031
24.3.2.23	ETNCnTSFRCR — Too-Short Frame Receive Counter Register.....	2032
24.3.2.24	ETNCnTLFRCR — Too-Long Frame Receive Counter Register.....	2033
24.3.2.25	ETNCnRFCR — Received Alignment Error Frame Counter Register	2034
24.3.2.26	ETNCnMAFCR — Multicast Address Frame Receive Counter Register	2035
24.3.3	EDMAC Registers	2036
24.3.3.1	ETNCnEDMR — EDMAC Mode Register	2036
24.3.3.2	ETNCnEDTRR — EDMAC Transmit Request Register	2037
24.3.3.3	ETNCnEDRRR — EDMAC Receive Request Register.....	2038
24.3.3.4	ETNCnTDLAR — Transmit Descriptor List Start Address Register	2039

24.3.3.5	ETNCnRDLAR — Receive Descriptor List Start Address Register	2040
24.3.3.6	ETNCnEESR — ETHERC/EDMAC Status Register	2041
24.3.3.7	ETNCnEESIPR — ETHERC/EDMAC Status Interrupt Enable Register	2045
24.3.3.8	ETNCnTRSCER — ETHERC/EDMAC Transmit/Receive Status Copy Enable Register	2047
24.3.3.9	ETNCnRMFCR — Missed-Frame Counter Register	2048
24.3.3.10	ETNCnTFTR — Transmit FIFO Threshold Register	2049
24.3.3.11	ETNCnFDR — FIFO Depth Register	2050
24.3.3.12	ETNCnRMCR — Receive Method Control Register	2051
24.3.3.13	ETNCnTFUCR — Transmit FIFO Underflow Counter	2052
24.3.3.14	ETNCnRFOCR — Receive FIFO Overflow Counter	2053
24.3.3.15	ETNCnIOSR — Independent Output Signal Setting Register	2054
24.3.3.16	ETNCnFCFTR — Flow Control Start FIFO Threshold Setting Register	2055
24.3.3.17	ETNCnRPADIR — Receive Data Padding Insert Register	2056
24.3.3.18	ETNCnTRIMD — Transmit Interrupt Setting Register	2057
24.3.3.19	ETNCnRBWAR — Receive Buffer Write Address Register	2058
24.3.3.20	ETNCnRDFAR — Receive Descriptor Fetch Address Register	2059
24.3.3.21	ETNCnTBRAR — Transmit Buffer Read Address Register	2060
24.3.3.22	ETNCnTDFAR — Transmit Descriptor Fetch Address Register	2061
24.4	Operation	2062
24.4.1	ETHERC Operation	2062
24.4.1.1	Transmission	2062
24.4.1.2	Reception	2064
24.4.1.3	Frame Timings	2066
24.4.1.4	Accessing MII/RMII Registers	2069
24.4.1.5	Magic Packet Detection	2072
24.4.1.6	Adjusting Transmission Efficiency by Changing the IPG	2073
24.4.1.7	Flow Control	2073
24.4.1.8	Interrupts	2075
24.4.2	EDMAC Operations	2076
24.4.2.1	Descriptor Lists and Data Buffers	2076
24.4.2.2	Transmission	2082
24.4.2.3	Reception	2083
24.4.2.4	Multi-Buffer Frame Transmission	2084
24.4.2.5	Interrupts	2086
24.5	Usage Notes	2087
24.5.1	Operation in Each Communication Mode	2087
24.5.2	Conditions for the LCHNG Flag to Become 1	2087
24.5.3	Input to the RMII_RX_ER Pin While the RMII is Selected	2087
24.5.4	Stopping the EDMAC during Operations	2087
Section 25	Peripheral Sensor Interface 5 S (PSI5-S)	2088
25.1	Features of PSI5-S	2088
25.1.1	Units and Channels	2088
25.1.2	Register Base Address	2088
25.1.3	Clock Supply	2088
25.1.4	Interrupt Requests	2089
25.1.5	Reset Sources	2090
25.1.6	External Input/Output Signals	2090

25.2	Outline of Functions.....	2091
25.2.1	Functional Overview.....	2091
25.2.1.1	Terms.....	2092
25.3	Registers.....	2095
25.3.1	List of Registers	2095
25.3.1.1	Common Register/Config	2095
25.3.2	Common Register/Config.....	2108
25.3.2.1	PSI5SPUOEB — PSI5-S/UART Operation Enable Register.....	2108
25.3.2.2	PSI5SPUOMD — PSI5-S/UART Operation Mode Register	2109
25.3.2.3	PSI5SPUOS — PSI5-S/UART Operation Status Register	2110
25.3.2.4	PSI5SPUNFST — PSI5-S/UART Noise Filter Set Register	2111
25.3.2.5	PSI5SPUSWR — PSI5-S/UART Software Reset Register	2112
25.3.2.6	PSI5SPRMBC — PSI5-S Receive MailBox Data Clear Register.....	2113
25.3.2.7	PSI5SPUCLB — PSI5-S/UART Communication Loop Back Register	2114
25.3.2.8	PSI5SPUPTS — PSI5-S/UART Rx/Tx Parity Set Register	2115
25.3.2.9	PSI5SPUBCE — PSI5-S/UART Baud Rate Clock Enable Register	2116
25.3.2.10	PSI5SPUBPR — PSI5-S/UART Baud Rate Paramater Register.....	2117
25.3.2.11	PSI5SPTPS — PSI5-S Timestamp Prescaler Register.....	2118
25.3.2.12	PSI5SPTCAS — PSI5-S Timestamp Counter A Select Register.....	2119
25.3.2.13	PSI5SPTCBS — PSI5-S Timestamp Counter B Select Register.....	2120
25.3.2.14	PSI5SPTCAE — PSI5-S Timestamp Counter A Enable Register.....	2121
25.3.2.15	PSI5SPTCAC — PSI5-S Timestamp Counter A Clear Register	2122
25.3.2.16	PSI5SPTCBE — PSI5-S Timestamp Counter B Enable Register.....	2123
25.3.2.17	PSI5SPTCBC — PSI5-S Timestamp Counter B Clear Register.....	2124
25.3.2.18	PSI5SPATCE — PSI5-S All Timestamp Counter Enable Register	2125
25.3.2.19	PSI5SPATCC — PSI5-S All Timestamp Counter Clear Register	2126
25.3.2.20	PSI5SUCRIE — UART Communication Rx Interrupt Enable.....	2127
25.3.2.21	PSI5SUCTIE — UART Communication Tx Interrupt Enable.....	2128
25.3.2.22	PSI5SUCDRE — UART Communication DMA Request Enable Register	2129
25.3.3	Common Register/Rx.....	2130
25.3.3.1	PSI5SUCRD — UART Communication Rx Data Register	2130
25.3.3.2	PSI5SUCRS — UART Communication Rx Status Register.....	2131
25.3.3.3	PSI5SUCRSC — UART Communication Rx Status Clear Register	2133
25.3.4	Common Register/Tx	2134
25.3.4.1	PSI5SPTFST — PSI5-S Tx Frame Start Register.....	2134
25.3.4.2	PSI5SPTFNM — PSI5-S Tx Frame Number Register	2135
25.3.4.3	PSI5SPTFD1 — PSI5-S Tx Frame Data1 Register.....	2136
25.3.4.4	PSI5SPTFD2 — PSI5-S Tx Frame Data2 Register.....	2137
25.3.4.5	PSI5SPTFS — PSI5-S Tx Frame Status Register	2138
25.3.4.6	PSI5SPTFIS — PSI5-S Tx FIFO Status Register	2139
25.3.4.7	PSI5SUCTD — UART Communication Tx Data Register	2140
25.3.4.8	PSI5SUCTM — UART Communication Tx Monitoring Register	2141
25.3.4.9	PSI5SUCTS — UART Communication Tx Status Register.....	2142
25.3.4.10	PSI5SUCTSC — UART Communication Tx Status Clear Register	2143
25.3.5	Ch0 Register/Config.....	2144
25.3.5.1	PSI5SPRCF10 — PSI5-S Receive Config1 Ch0 Register	2144
25.3.5.2	PSI5SPRCF20 — PSI5-S Receive Config2 Ch0 Register	2147
25.3.5.3	PSI5SPWDE0 — PSI5-S WDT Enable Ch0 Register	2149
25.3.5.4	PSI5SPWDP0 — PSI5-S WDT Prescaler Ch0 Register	2150
25.3.5.5	PSI5SPWDEV0 — PSI5-S WDT Expiration Value Ch0 Register.....	2151
25.3.5.6	PSI5SPCIE0 — PSI5-S CPU Interrupt Enable Ch0 Register.....	2152

25.3.5.7	PSI5SPDRE0 — PSI5-S DMA Transfer Request Enable Ch0 Register	2154
25.3.6	Ch0 Register/Rx.....	2155
25.3.6.1	PSI5SPRES0 — PSI5-S Receive Error Status Ch0 Register	2155
25.3.6.2	PSI5SPRESC0 — PSI5-S Receive Error Status Clear Ch0 Register	2156
25.3.6.3	PSI5SPTCDT0 — PSI5-S Timestamp Capture Data Ch0 Register	2157
25.3.6.4	PSI5SPTCDC0 — PSI5-S Timestamp Capture Data Clear Ch0 Register	2158
25.3.7	Ch0 Register/Interrupt.....	2159
25.3.7.1	PSI5SPCIS0 — PSI5-S CPU Interrupt Status Ch0 Register	2159
25.3.7.2	PSI5SPCISC0 — PSI5-S CPU Interrupt Status Clear Ch0 Register	2162
25.3.8	Ch <i>n</i> Register/Config (<i>n</i> : 1 to 7)	2164
25.3.8.1	PSI5SPRCF1 <i>n</i> — PSI5-S Receive Config1 Chn Register (<i>n</i> : 1 to 7).....	2164
25.3.8.2	PSI5SPRCF2 <i>n</i> — PSI5-S Receive Config2 Chn Register	2167
25.3.8.3	PSI5SPWDE <i>n</i> — PSI5-S WDT Enable Chn Register	2170
25.3.8.4	PSI5SPWDP <i>n</i> — PSI5-S WDT Prescaler Chn Register	2171
25.3.8.5	PSI5SPWDEV <i>n</i> — PSI5-S WDT Expiration Value Chn Register.....	2172
25.3.8.6	PSI5SPTCD <i>n</i> — PSI5-S Tx Command Data Chn Register	2173
25.3.8.7	PSI5SPCIEn — PSI5-S CPU Interrupt Enable Chn Register.....	2174
25.3.8.8	PSI5SPDRE <i>n</i> — PSI5-S DMA Transfer Request Enable Chn Register	2176
25.3.8.9	PSI5SPSTP <i>n</i> — PSI5-S Sync Trigger Prescaler Chn Register	2177
25.3.8.10	PSI5SPSTEV <i>n</i> — PSI5-S Sync Trigger Expiration Value Chn Register.....	2178
25.3.8.11	PSI5SPSTS <i>n</i> — PSI5-S Sync Trigger Select Ch <i>n</i> Register.....	2179
25.3.9	Ch <i>n</i> Register/Rx (<i>n</i> : 1 to 7)	2180
25.3.9.1	PSI5SPRES <i>n</i> — PSI5-S Receive Error Status Chn Register	2180
25.3.9.2	PSI5SPRESC <i>n</i> — PSI5-S Receive Error Status Clear Chn Register	2182
25.3.9.3	PSI5SPTCDT <i>n</i> — PSI5-S Timestamp Capture Data Chn Register	2184
25.3.9.4	PSI5SPTCDC <i>n</i> — PSI5-S Timestamp Capture Data Clear Chn Register	2185
25.3.10	Ch <i>n</i> Register/Tx (<i>n</i> : 1 to 7).....	2186
25.3.10.1	PSI5SPDDTP <i>n</i> — PSI5-S DDSR Type Chn Register	2186
25.3.10.2	PSI5SPDDD <i>n</i> — PSI5-S DDSR Data Chn Register	2187
25.3.10.3	PSI5SPDDSn — PSI5-S DDSR Status Chn Register	2188
25.3.10.4	PSI5SPDDSP <i>n</i> — PSI5-S DDSR Stop Chn Register	2189
25.3.11	Ch <i>n</i> Register/Interrupt (<i>n</i> : 1 to 7)	2190
25.3.11.1	PSI5SPCIS <i>n</i> — PSI5-S CPU Interrupt Status Chn Register.....	2190
25.3.11.2	PSI5SPCISC <i>n</i> — PSI5-S CPU Interrupt Status Clear Chn Register	2193
25.3.12	Ch 0 Frm <i>m</i> MB Data (<i>m</i> : 1, 2).....	2196
25.3.12.1	PSI5SPMB0 <i>m</i> S — PSI5-S Receive MailBox Ch0 Frm <i>m</i> Status Register.....	2196
25.3.12.2	PSI5SPMB0 <i>m</i> D — PSI5-S Receive MailBox Ch0 Frm <i>m</i> Data Register	2198
25.3.12.3	PSI5SPMB0 <i>m</i> T — PSI5-S Receive MailBox Ch0 Frm <i>m</i> Timestamp Register	2199
25.3.13	Ch <i>n</i> Frm <i>m</i> MB Data (<i>n</i> : 1 to 7) (<i>m</i> : 1 to 6)	2200
25.3.13.1	PSI5SPMB <i>n</i> <i>m</i> S — PSI5-S Receive MailBox ch <i>n</i> Frm <i>m</i> Status Register	2200
25.3.13.2	PSI5SPMB <i>n</i> <i>m</i> D — PSI5-S Receive MailBox Ch0 Frm <i>m</i> Data Register	2202
25.3.13.3	PSI5SPMB <i>n</i> <i>m</i> T — PSI5-S Receive MailBox Chn Frm <i>m</i> Timestamp Register	2203
25.4	Operation Modes	2204
25.5	Setting Procedures	2206
25.5.1	Common Setting Procedure.....	2207
25.5.2	PSI5S Mode Setting Procedure	2209
25.5.2.1	PSI5 Frame Reception Setting	2211
25.5.2.2	WDT Setting.....	2213
25.5.2.3	Timestamp Setting	2214
25.5.2.4	Interrupt Setting	2216

25.5.2.5	DMA Transfer Request Setting	2217
25.5.2.6	Synchronization Pulse Setting Procedure	2218
25.5.2.7	Tx Command Setting Procedure	2219
25.5.3	UART Mode Setting Procedure	2220
25.5.4	Procedure for Transmitting Transceiver Commands	2222
25.5.5	DDSR Transmission Procedure	2224
25.5.6	PSI5 Frame Reception Procedures	2226
25.5.6.1	Normal Reception	2226
25.5.6.2	Abnormal Reception	2228
25.5.6.3	Response Frame from Transceiver Reception	2229
25.5.7	Initialization and Operation Stop Procedures	2231
25.5.7.1	Initialization by Software Reset	2231
25.5.7.2	DDSR Transmission Stop	2232
25.5.7.3	MailBox Initialization	2232
25.5.8	Interrupt Processing Procedures	2233
25.5.8.1	Interrupt Processing in PSI5S Mode	2233
25.5.8.2	Interrupt Processing in UART Mode	2234
25.5.9	DMA Request Processing Procedures	2235
25.5.9.1	DMA Request (Reception) Processing in PSI5S Mode	2235
25.5.9.2	DMA Request (Transmission) Processing in PSI5S Mode	2236
25.5.9.3	DMA Request (Reception) Processing in UART Mode	2236
25.5.9.4	DMA Request (Transmission) Processing in UART Mode	2237
25.5.10	UART Mode Communication Procedures	2238
25.5.10.1	UART Reception	2238
25.5.10.2	UART Transmission	2240
25.5.11	Loopback Test	2243
25.5.11.1	Loopback Test Setting Procedure	2243
25.5.11.2	Loopback Test Procedure	2244
25.5.12	Configuration Mode Entering Procedure	2246
25.6	Operations	2247
25.6.1	Functional Configuration	2247
25.6.1.1	PSI5-S Reception Function	2248
25.6.1.2	PSI5-S Transmission Function	2250
25.6.1.3	Timestamp Generation Function	2252
25.6.1.4	Synchronization Pulse Generation Function	2255
25.6.2	Operations in PSI5S Mode	2256
25.6.2.1	Normal Reception (in Synchronous Mode)	2257
25.6.2.2	Normal Reception (in Asynchronous Mode)	2267
25.6.2.3	Normal Transmission (Command Data)	2270
25.6.2.4	Normal Transmission (ECU-to-Sensor Data)	2271
25.6.2.5	Normal Transmission (ECU-to-Sensor Data Transmission Stopped)	2275
25.6.2.6	Normal Transmission (Transmit Request Arbitration)	2276
25.6.2.7	Abnormal Reception (Reception Error)	2278
25.6.2.8	Abnormal Reception (WDT Error)	2281
25.6.2.9	Abnormal Transmission (Transmission Error)	2289
25.6.2.10	Interrupt Output	2291
25.6.2.11	DMA Request Output	2294
25.6.2.12	Transition to Configuration Mode	2295
25.6.3	Operations in UART Mode	2297
25.6.3.1	Normal Reception	2297

25.6.3.2	Normal Transmission	2298
25.6.3.3	Abnormal Reception (Reception Error).....	2299
25.6.3.4	Abnormal Transmission (Transmission Error)	2301
25.6.3.5	Interrupt Output.....	2302
25.6.3.6	DMA Request Output.....	2303
25.6.3.7	Transition to Configuration Mode.....	2303
25.6.4	Clearing Status Signals.....	2305
25.7	Note	2306
25.7.1	Restrictions	2306
25.7.2	Notes	2306
Section 26	High Speed Serial Peripheral Interface (HS-SPI).....	2307
26.1	Features of HS-SPI.....	2307
26.1.1	Units and Channels.....	2307
26.1.2	Register Base Address	2307
26.1.3	Clock Supply	2308
26.1.4	Interrupt Requests.....	2308
26.1.5	Reset Sources.....	2308
26.1.6	External Input/Output Signals	2308
26.2	Overview.....	2309
26.2.1	Functional Overview.....	2309
26.2.2	Block Diagram.....	2311
26.3	Registers.....	2312
26.3.1	List of Registers	2312
26.3.2	HSPInCKEN — Clock Control Register	2313
26.3.3	HSPInSRST — Software Reset Register	2314
26.3.4	HSPInEN — Communication Enabling Register	2315
26.3.5	HSPInMD — Communication Mode Register	2317
26.3.6	HSPInCTL — Communication Control Register	2319
26.3.7	HSPInSSCTL — SSL Polarity Selection Register	2321
26.3.8	HSPInSCKDLY — SPCK Delay Selection Register	2322
26.3.9	HSPInSSNDLY — SSL Negation Delay Selection Register.....	2323
26.3.10	HSPInNFMDLY — Next Frame Delay Selection Register.....	2324
26.3.11	HSPInSPLDLY — Sampling Delay Selection Register.....	2325
26.3.12	HSPInCDIV — Communication Clock Division Selection Register	2326
26.3.13	HSPInTSAR — Transmission Data Transfer Source Address Setting Register	2327
26.3.14	HSPInRDAR — Reception Data Transfer Destination Address Setting Register	2328
26.3.15	HSPInCFSET — Communication Frame Count Setting Register	2329
26.3.16	HSPInIREN — Interrupt Enabling Register	2330
26.3.17	HSPInIRST — Interrupt Status Register.....	2332
26.3.18	HSPInIRCL — Interrupt Status Clearing Register	2336
26.3.19	HSPInACTST — Operation Status Register.....	2338
26.3.20	HSPInCTFM — Current Transmission Frame Count Register.....	2339
26.3.21	HSPInCRFM — Current Reception Frame Count Register.....	2340

26.3.22	HSPInVPEN — Virtual Port Output Enabling Register	2341
26.3.23	HSPInTESTMD — Test Mode Register	2342
26.4	Detailed Functional Descriptions	2344
26.4.1	Overview of the HS-SPI Operation	2344
26.4.1.1	Master Transmission/Reception Mode	2344
26.4.1.2	Master Transmission Operation	2346
26.4.1.3	Slave Transmission/Reception	2347
26.4.1.4	Slave Transmission	2348
26.4.1.5	Slave Reception	2349
26.4.1.6	Slave Reception (When the Virtual Port Output is Enabled)	2350
26.4.2	Control on the HS-SPI Pins	2351
26.4.3	Data Format	2352
26.4.3.1	Frame Data Format	2352
26.4.3.2	Relationship between the Serial Clock and Serial Data	2355
26.4.3.3	MSB First and LSB First	2357
26.4.4	Parity Bit	2358
26.4.4.1	Even Parity	2358
26.4.4.2	Odd Parity	2358
26.4.4.3	Parity Omission	2358
26.4.5	Buffer Structure	2359
26.4.5.1	Structure of the Transmission Buffer	2359
26.4.5.2	Reception Buffer Structure	2360
26.4.6	Interrupt Requests	2361
26.4.6.1	Interrupt Sources	2361
26.4.6.2	H-Bus Error Interrupt Requests	2362
26.4.6.3	Error Interrupt Requests	2363
26.4.6.4	Communication Start Interrupt Requests	2364
26.4.6.5	Communication Complete Interrupts	2364
26.4.7	Reception and Transmission during Suspension of Transfer	2365
26.4.8	Operations in cases of Frame Overrun during Transmission	2365
26.4.9	Operation in the cases of Frame Overrun during Reception	2365
26.4.10	Timeouts	2366
26.4.11	Burst Operation	2367
26.4.12	Virtual Port Function	2368
26.5	Operation	2369
26.5.1	Power-on, Initialization and Stop Processing Flows	2369
26.5.2	Master Mode	2370
26.5.2.1	Transmission and Reception Setting Flows	2370
26.5.2.2	Transmission Setting Flow	2372
26.5.2.3	Loop-back Test Setting Flow	2373
26.5.3	Slave Mode	2374
26.5.3.1	Transmission and Reception Setting Flows	2374
26.5.3.2	Transmission Setting Flow	2375
26.5.3.3	Reception Setting Flow	2376
26.5.3.4	Virtual Port Reception Setting Flow	2377
26.5.3.5	Flow of Loop-back Test Setting	2378
26.5.4	Flow of Re-send Using Communications Start Trigger	2379
26.5.5	Error Handling Flow	2380

26.5.6	Suspend Communication Procedure	2381
26.5.7	Notes for Suspending Communication	2382
26.5.8	Loop-Back Testing	2383
26.6	Notes	2384
26.6.1	Notes for Suspending Communication	2384
26.6.2	Register Write Protection and Software Resets	2384
26.6.2.1	Write Protection	2384
26.6.2.2	Registers for Software Resets	2384
26.6.2.3	Notes when Using Communications Start Trigger Bit	2384
26.6.3	DMA Function	2385
26.6.3.1	DMA Operation during Suspended Communication	2385
26.6.3.2	DMA Operation in Response to H-Bus Errors	2385
26.6.3.3	Transmission Underruns in Slave Mode	2385
26.6.3.4	DMA Operation When Virtual Port Output Enabled	2385
26.6.4	Data Storage and Address Space	2386
26.6.5	Errors	2388
26.6.5.1	Resume Transfer after an Error Occurs	2388
26.6.5.2	Operations When a Reception Overflow Occurs	2388
26.6.5.3	Transmission Underrun in Slave Mode	2388
Section 27	Renesas High-Speed Serial I/F (RHSIF)	2389
27.1	Feature	2389
27.1.1	Number of Units and Channels	2389
27.1.2	Register Base Address	2389
27.1.3	Clock Supply	2389
27.1.4	Interrupt Requests	2390
27.1.5	Reset Sources	2390
27.1.6	External Input/Output Signals	2390
27.2	Overview	2391
27.2.1	Functional Overview	2391
27.2.2	Block Diagram	2392
27.2.3	Communication Protocol	2393
27.2.3.1	Protocol Definitions	2393
27.2.3.2	Connection	2393
27.2.3.3	L1 Frame	2394
27.2.3.4	L2 Frame	2397
27.2.3.5	Data Format	2400
27.2.3.6	All Sending and Receiving Frames	2401
27.2.3.7	Number of Transfer Bits in each Frame	2402
27.2.3.8	List of Command and Response	2402
27.2.3.9	Command Pipelining	2402
27.3	Operation	2403
27.3.1	Operation Flow	2403
27.4	Transport Layer (L2)	2404
27.4.1	Overview	2404
27.4.1.1	Functional Overview	2404

27.4.1.2	Block Diagram.....	2406
27.4.2	Registers.....	2407
27.4.2.1	List of Registers.....	2407
27.4.2.2	Legend.....	2410
27.4.2.3	Detailed Register Descriptions.....	2411
27.4.3	Operation.....	2453
27.4.3.1	Initial Settings.....	2453
27.4.3.2	Non-Stream Command Transmission by Initiator Node.....	2465
27.4.3.3	Non-Stream Command Reception by Target Node.....	2470
27.4.3.4	Stream Command Transmission by Initiator Node.....	2475
27.4.3.5	Stream Command Reception by Target Node.....	2479
27.4.3.6	CRC Error Handling.....	2484
27.4.4	Interrupt Specification.....	2485
27.4.4.1	Interrupt Signals and Corresponding Interrupt Sources.....	2485
27.4.4.2	Interrupt Signal Status Register.....	2486
27.5	Datalink Layer (L1).....	2487
27.5.1	Overview.....	2487
27.5.1.1	Functional Overview.....	2487
27.5.1.2	Block Diagram.....	2488
27.5.2	Registers.....	2489
27.5.2.1	List of Registers.....	2489
27.5.2.2	Legend.....	2491
27.5.2.3	Detailed Register Description.....	2492
27.5.3	Operation.....	2526
27.5.3.1	Initial Settings.....	2526
27.5.3.2	ICLC Command Control Method.....	2529
27.5.3.3	Changing Communication Speed Using ICLC Commands.....	2532
27.5.3.4	Transiting Slave State Using ICLC Commands.....	2538
27.5.3.5	CTS Commands.....	2540
27.5.3.6	Sleep Mode.....	2542
27.5.4	Interrupt Specification.....	2543
27.5.4.1	Interrupt Source List.....	2543
27.5.4.2	Error Detect Conditions.....	2546
27.5.4.3	Operation Upon Error Occurrence.....	2548
27.5.5	Debug.....	2548
27.5.6	Usage Note.....	2549
27.5.6.1	About Fast Mode of Slave.....	2549
Section 28	Window Watchdog Timer (WDTB).....	2550
28.1	Features of WDTB.....	2550
28.1.1	Units and Channels.....	2550
28.1.2	Register Base Addresses.....	2550
28.1.3	Clock Supply.....	2551
28.1.4	Interrupt Requests.....	2551
28.1.5	Reset Sources.....	2551
28.1.6	Correspondence of WDTB and CPU.....	2551
28.2	Overview.....	2552
28.2.1	Functional Overview.....	2552

28.2.2	Block Diagram.....	2553
28.3	Registers.....	2554
28.3.1	List of Registers	2554
28.3.2	Detail of Registers.....	2555
28.3.2.1	WDTBnWDTE — WDTB Enable Register.....	2555
28.3.2.2	WDTBnMD — WDTB Mode Register.....	2556
28.3.2.3	WDTBnWOST — WDTB Window Open Start Register	2558
28.3.2.4	WDTBnWIS — WDTB Interrupt Output Timing Setting Register	2559
28.4	Operation.....	2560
28.4.1	WDTB after Reset Release.....	2560
28.4.1.1	Start Modes.....	2560
28.4.1.2	Start Mode Selection (Only for WDTB0).....	2560
28.4.1.3	WDTB Settings after Reset Release	2560
28.4.1.4	Default Start Mode Timing (Only for WDTB0)	2561
28.4.1.5	Software Trigger Start Mode Timing (Common to all WDTB)	2562
28.4.2	WDTB Trigger	2563
28.4.3	WDTB Error Detection	2563
28.4.3.1	WDTB Error Mode	2564
28.4.4	WDTB Interrupt Output	2565
28.4.5	Window Function.....	2567
Section 29	OS Timer (OSTM).....	2569
29.1	Features of OSTM	2569
29.1.1	Number of Units	2569
29.1.2	Register Base Addresses	2569
29.1.3	Clock Supply	2569
29.1.4	Interrupts Requests.....	2570
29.1.5	Reset Sources.....	2570
29.1.6	Correspondence of OSTM and CPU	2570
29.2	Overview.....	2571
29.2.1	Function Overview	2571
29.2.2	Block Diagram.....	2572
29.2.3	Output Modes (only for OSTM0).....	2573
29.2.4	Interrupt Requests (OSTMnTINT).....	2574
29.3	Registers.....	2575
29.3.1	List of Registers	2575
29.3.2	Detail of Registers.....	2576
29.3.2.1	OSTMnCMP — OSTM Compare Register	2576
29.3.2.2	OSTMnCNT — OSTM Counter Register.....	2577
29.3.2.3	OSTMnTO — OSTM Output Register	2578
29.3.2.4	OSTMnTOE — OSTM Output Enable Register	2579
29.3.2.5	OSTMnTE — OSTM Count Enable Status Register	2580
29.3.2.6	OSTMnTS — OSTM Count Start Trigger Register.....	2581
29.3.2.7	OSTMnTT — OSTM Count Stop Trigger Register	2582
29.3.2.8	OSTMnCTL — OSTM Control Register.....	2583
29.4	Operation.....	2584

29.4.1	Starting and Stopping OSTM	2584
29.4.2	Interval Timer Mode	2586
29.4.2.1	Basic Operation in Interval Timer Mode	2586
29.4.2.2	Operation when OSTMnCMP = 0000 0000 _H	2588
29.4.2.3	Forced Restart Operation in Interval Timer Mode	2589
29.4.3	Free-Run Compare Mode	2590
29.4.3.1	Basic Operation in Free-Run Compare Mode	2590
29.4.3.2	Operation when OSTMnCMP = 0000 0000 _H	2592
29.4.3.3	Forced Restart Operation in Free-Run Compare Mode	2593
Section 30	ATU-V	2594
30.1	Features of ATU-V	2594
30.1.1	Units and Channels	2594
30.1.2	Register Base Address	2594
30.1.3	Clock Supply	2594
30.1.4	Interrupts Requests	2595
30.1.5	Reset Sources	2619
30.1.6	External Input/Output Signals	2619
30.2	Overview	2622
30.2.1	Configuration of ATU-V	2627
30.2.2	ATU-V Registers	2629
30.3	Common Controller	2695
30.3.1	Overview of Operation	2695
30.3.2	Registers Related to Common Controller	2696
30.3.2.1	ATUENR — ATU Master Enable Register	2696
30.3.2.2	CBCNT — Clock Bus Control Register	2699
30.3.2.3	NCMR — Noise Cancellation Mode Register	2700
30.3.2.4	Description of Operations for Noise Cancellation	2706
30.4	Prescaler	2710
30.4.1	Overview of Operation	2710
30.4.2	Registers Related to Prescaler	2711
30.4.2.1	PSCRx — Prescaler Registers x	2711
30.4.3	Details of Operation	2712
30.4.3.1	Starting Prescalers	2712
30.4.3.2	Stopping and Restarting Operation	2712
30.5	Timer A	2713
30.5.1	Overview of Operation	2713
30.5.2	Registers Related to Timer A	2715
30.5.2.1	TCR1A — Timer Control Register 1A	2715
30.5.2.2	TCR2A — Timer Control Register 2A	2717
30.5.2.3	TCR2AS1 — Timer Control Register 2AS1	2718
30.5.2.4	TCR2AS2 — Timer Control Register 2AS2	2719
30.5.2.5	TCR3A — Timer Control Register 3A	2720
30.5.2.6	TCR4A — Timer Control Register 4A	2721
30.5.2.7	TCR5A — Timer Control Register 5A	2722
30.5.2.8	NCMCR1A — Noise Cancellation Mode Channel Register 1A	2723
30.5.2.9	NCMCR2A — Noise Cancellation Mode Channel Register 2A	2724

30.5.2.10	TIOR1A — Timer I/O Control Register 1	2725
30.5.2.11	TIOR2A — Timer I/O Control Register 2	2726
30.5.2.12	TIERA — Timer Interrupt Enable Register A	2728
30.5.2.13	TSRA — Timer Status Register A	2729
30.5.2.14	TSCRA — Timer Status Clear Register A	2730
30.5.2.15	ICRAx — Input Capture Registers Ax.....	2731
30.5.2.16	ICRA2x — Input Capture Registers A2x.....	2732
30.5.2.17	TCNTA — Free-Running Counter A	2733
30.5.2.18	TILRA — Timer Input Signal Level Register.....	2734
30.5.2.19	NCNTAx — Noise Canceler Counters Ax.....	2735
30.5.2.20	NCRAx — Noise Cancel Registers Ax	2737
30.5.3	Details of Operation	2738
30.5.3.1	Operation of Noise Canceler	2738
30.5.3.2	Operation of Free-Running Counter	2738
30.5.3.3	Input Capture	2739
30.5.3.4	Pin Level Capture Operation	2741
30.6	Timer B	2742
30.6.1	Overview of Operation	2742
30.6.2	Timer B Control Registers.....	2747
30.6.2.1	TCRB — Timer Control Register B.....	2747
30.6.2.2	TCRBS1 — Timer Control Register BS1	2749
30.6.2.3	TCRBS2 — Timer Control Register BS2	2750
30.6.2.4	TIORB — Timer I/O Control Register B.....	2751
30.6.2.5	TSRB — Timer Status Register B	2753
30.6.2.6	TSCRB — Timer Status Clear Register B	2758
30.6.2.7	TICRB — Timer Interrupt Control Register B	2760
30.6.2.8	TIERB — Timer Interrupt Enable Register B.....	2762
30.6.2.9	CHSELBR — Channel Select Register B.....	2763
30.6.2.10	TCNT4CRB — Multiplied-and-Corrected Clock Generating Counter B4 Control Register	2764
30.6.2.11	TCNTB0 — Edge Interval Measuring Counter B0.....	2765
30.6.2.12	TCNTB0S1 — Edge Interval Measuring Counter B0S1	2766
30.6.2.13	TCNTB0S2 — Edge Interval Measuring Counter B0S2	2767
30.6.2.14	ICRB0 — Input Capture Register B0 DICRB0 — Input Capture Register B0 MIICRB0 — Mirror Input Capture Register B0	2768
30.6.2.15	ICRB0S1 — Input Capture Register B0S1	2769
30.6.2.16	ICRB0S2 — Input Capture Register B0S2	2770
30.6.2.17	RECRB1 to RECRB6 — Record Registers B1 to B6 DRECRB1 — Record Registers B1	2771
30.6.2.18	RBURB0 to RBURB6 — Record Backup Register B0 to B6	2772
30.6.2.19	ICRB30 to ICRB37 — Input Capture Registers B30 to B37	2773
30.6.2.20	OCRB0 — Output Compare Register B0	2774
30.6.2.21	OCRB0S1 — Output Compare Register B0S1	2775
30.6.2.22	OCRB0S2 — Output Compare Register B0S2	2776
30.6.2.23	TCNTB1 — Event Counter B1	2777
30.6.2.24	TCNTB1S1 — Event Counter B1S1	2778
30.6.2.25	TCNTB1S2 — Event Counter B1S2	2779
30.6.2.26	OCRB1 — Output Compare Register B1	2780
30.6.2.27	OCRB1S1 — Output Compare Register B1S1	2781
30.6.2.28	OCRB1S2 — Output Compare Register B1S2	2782
30.6.2.29	OCRB10 — Output Compare Register B10	2783
30.6.2.30	OCRB10S1 — Output Compare Register B10S1	2784

30.6.2.31	OCRB10S2 — Output Compare Register B10S2	2785
30.6.2.32	OCRB11 — Output Compare Register B11	2786
30.6.2.33	OCRB11S1 — Output Compare Register B11S1	2787
30.6.2.34	OCRB11S2 — Output Compare Register B11S2	2788
30.6.2.35	OCRB12 — Output Compare Register B12	2789
30.6.2.36	OCRB12S1 — Output Compare Register B12S1	2790
30.6.2.37	OCRB12S2 — Output Compare Register B12S2	2791
30.6.2.38	ICRB1 — Input Capture Register B1	2792
30.6.2.39	ICRB1S1 — Input Capture Register B1S1	2793
30.6.2.40	ICRB1S2 — Input Capture Register B1S2	2794
30.6.2.41	ICRB2 — Input Capture Register B2	2795
30.6.2.42	ICRB2S1 — Input Capture Register B2S1	2796
30.6.2.43	ICRB2S2 — Input Capture Register B2S2	2797
30.6.2.44	LDB — Load Register B	2798
30.6.2.45	RLDB — Reload Register B	2799
30.6.2.46	TCNTB2 — Reloadable Counter B2	2800
30.6.2.47	PIMR1 — Pulse Interval Multiplier Register 1	2801
30.6.2.48	PIMR2 — Pulse Interval Multiplier Register 2	2802
30.6.2.49	TCNTB6 — Multiplied Clock Counter B6	2803
30.6.2.50	ICRB6 — Input Capture Register B6	2804
30.6.2.51	RARB6 — Multiplication Setting Register B6	2805
30.6.2.52	TCNTB6M — Frequency-Multiplied Clock Counter B6M	2806
30.6.2.53	OCRB6 — Output Compare Register B6	2807
30.6.2.54	OCRB7 — Output Compare Register B7	2808
30.6.2.55	OCRB20 to OCRB43 — Output Compare Registers B20 to 43	2809
30.6.2.56	TSEQCRB — Timer Sequencer Control Register	2810
30.6.2.57	TSEQRB — Timer Sequencer Register	2811
30.6.2.58	TSEQENB0 to TSEQENB2 — Timer Sequencer Enable Registers 0 to 2	2812
30.6.2.59	TEPCFB0 to TEPCFB23 — Timer Event Cycle Correction Formula Definition Registers B0 to 23	2813
30.6.2.60	TCNTB3 — Correction Event Counter B3	2815
30.6.2.61	OCRB8 — Output Compare Register B8	2816
30.6.2.62	TEPCFENB0 to TEPCFENB2 — Timer Event Cycle Correction Enable Registers 0 to 2	2817
30.6.2.63	TEPCVALRB0 — Timer Event Input Cycle Correction Result Retention Register	2818
30.6.2.64	TEPCRECRB1 — Timer Event Input Cycle Correction Result Log Register	2819
30.6.2.65	TCNTB4 — Multiplied-and-Corrected Clock Counter B4	2820
30.6.2.66	TCNTB5 — Multiplied-and-Corrected Clock Generating Counter B5	2821
30.6.2.67	TCCLFRB — Correcting Counter Clear Flag Register B	2823
30.6.2.68	TCCLRB — Correcting Counter Clearing Register B	2824
30.6.3	Details of Operation	2825
30.6.3.1	Edge Interval Measuring Function and Edge Input Stopping Function	2825
30.6.3.2	Frequency-Multiplied Clock Generator	2833
30.6.3.3	Frequency-Multiplied Clock Signal Corrector	2837
30.7	Timer C	2844
30.7.1	Overview of Operation	2844
30.7.2	Registers Related to Timer C	2846
30.7.2.1	TSTRC — Timer Start Register C	2846
30.7.2.2	NCCRCx — Noise Canceler Control Register Cx	2847
30.7.2.3	TCRCx — Timer Control Registers Cx	2849
30.7.2.4	TSRCx — Timer Status Registers Cx	2854

30.7.2.5	TSCRCx — Timer Status Clear Register Cx	2857
30.7.2.6	TIORCx — Timer I/O Control Registers Cx	2858
30.7.2.7	TCNTCx — Timer Counters Cx	2861
30.7.2.8	GRCxy — Timer General Registers Cxy	2862
30.7.2.9	NCNTCxy — Noise Canceler Counters Cxy	2864
30.7.2.10	NCRCxy — Noise Canceler Registers Cxy	2866
30.7.2.11	OCRCxy — Output Compare Registers Cxy	2867
30.7.2.12	OCMRCxy — Output Compare Mirror Registers Cxy	2868
30.7.2.13	GMRCxy — Timer General Mirror Registers Cxy	2869
30.7.2.14	TIERCx — Timer Interrupt Enable Registers Cx	2870
30.7.2.15	CUCRCx — Counter Upper-Limit Setting Compare Registers Cx	2871
30.7.2.16	RCR1Cx — Range Comparison Value Setting Register 1Cx	2872
30.7.2.17	RCR2Cx — Range Comparison Value Setting Register 2Cx	2873
30.7.2.18	NCMCR1C — Noise Cancellation Mode Channel Register 1C	2875
30.7.2.19	NCMCR2C — Noise Cancellation Mode Channel Register 2C	2876
30.7.3	Details of Operation	2877
30.7.3.1	Input Capture Mode (Input Capture Function)	2877
30.7.3.2	Compare Match Mode (Compare Match Function)	2879
30.7.3.3	PWM Function	2882
30.7.3.4	One-Shot Pulse Mode (One-Shot Pulse Function)	2884
30.7.3.5	Counter Upper-Limit Setting Function	2891
30.7.3.6	Range Compare Function	2893
30.8	Timer D	2898
30.8.1	Overview of Operation	2898
30.8.2	Registers Related to Timer D	2903
30.8.2.1	TSTRD — Timer Start Register D	2903
30.8.2.2	TCRDx — Timer Control Registers Dx	2904
30.8.2.3	TCCRLRDx — Timer Compare Control Register Dx	2908
30.8.2.4	TIOR1Dx — Timer I/O Control Registers 1Dx	2910
30.8.2.5	TIOR2Dx — Timer I/O Control Register 2Dx	2912
30.8.2.6	TCMPEDx — Timer Compare Match Control Register Dx	2914
30.8.2.7	TIORDx — Timer I/O Control Register Dx	2915
30.8.2.8	OSELRDx — Output Select Register Dx	2916
30.8.2.9	ODRDx — Output Value Register Dx	2917
30.8.2.10	DSTRDx — Down Counter Start Register Dx	2918
30.8.2.11	DSR1Dx — Down Counter Status Registers 1Dx	2919
30.8.2.12	DSR2Dx — Down Count Status Register 2Dx	2921
30.8.2.13	DSCRDX — Down Count Status Clear Register Dx	2922
30.8.2.14	DCRDx — Down Counter Control Registers Dx	2923
30.8.2.15	TSRDx — Timer Status Registers Dx	2925
30.8.2.16	TSCRDX — Timer Status Clear Register Dx	2929
30.8.2.17	TOCRDX — Timer Output Control Registers Dx	2932
30.8.2.18	TOCCRDx — Timer Output Channel Control Registers Dx	2933
30.8.2.19	OSBRDX — Timer Offset Base Registers Dx	2935
30.8.2.20	TICTSELDx — Timer Input Capture Trigger Select Register Dx	2936
30.8.2.21	TCNT1Dx — Timer Counter 1Dx	2937
30.8.2.22	TCNT2Dx — Timer Counter 2Dx	2938
30.8.2.23	CUCR1Dx — Counter Upper-Limit Setting Compare Register 1Dx	2939
30.8.2.24	CUCR2Dx — Counter Upper-Limit Setting Compare Register 2Dx	2940
30.8.2.25	OCR1Dxy — Output Compare Registers 1Dxy	2941
30.8.2.26	RCR1Dx — Range Comparison Value Setting Register 1Dx	2942
30.8.2.27	OCR2Dxy — Output Compare Registers 2Dxy	2944

30.8.2.28	RCR2Dx — Range Comparison Value Setting Register 2Dx.....	2945
30.8.2.29	ICR1Dxy — Input Capture Register 1Dxy	2947
30.8.2.30	ICR2Dxy — Input capture register 2Dxy.....	2947
30.8.2.31	DCNTDxy — Timer Down Counters Dxy.....	2948
30.8.2.32	TIER1Dx — Timer Interrupt Enable Register 1Dx.....	2950
30.8.2.33	TIER2Dx — Timer Interrupt Enable Register 2Dx.....	2951
30.8.2.34	TIER3Dx — Timer Interrupt Enable Register 3Dx.....	2952
30.8.2.35	MIGCRDx — MIN Guard Control Register Dx.....	2953
30.8.2.36	MIGSELDx — MIN Guard Pulse Select Register Dx.....	2954
30.8.2.37	OFMINDxy —Pulse Output Off Minimum Width Setting Register Dxy	2955
30.8.2.38	ONMINDxy —Pulse Output On Minimum Width Setting Register Dxy	2955
30.8.2.39	OTOMINDxy —Pulse Output On-to-On Minimum Width Setting Register Dxy....	2956
30.8.2.40	OFMICNTDxy —Pulse Output Off Minimum Width Measurement Counter Dxy..	2956
30.8.2.41	ONMICNTDxy —Pulse Output On Minimum Width Measurement Counter Dxy .	2957
30.8.2.42	OTOMICNTDxy —Pulse Output On-to-On Minimum Width Measurement Counter Dxy	2957
30.8.2.43	ONCAP1Dxy — Input Capture Register ONCAP1Dxy.....	2958
30.8.2.44	ONCAP2Dxy — Input Capture Register ONCAP2Dxy.....	2958
30.8.2.45	OFCAP1Dxy — Input Capture Register OFCAP1Dxy.....	2959
30.8.2.46	OFCAP2Dxy — Input Capture Register OFCAP2Dxy.....	2959
30.8.2.47	TSR2Dx — Timer Status Register 2Dx	2960
30.8.2.48	TSCR2Dx — Timer Status Clear Register 2Dx	2961
30.8.2.49	CCAPSELD — Common Input Capture Select Register D	2962
30.8.2.50	CICRD0-7 — Common Input Capture Register D0-7	2963
30.8.3	Details of Operation	2964
30.8.3.1	Range Comparison Function	2969
30.8.3.2	Counter Upper-Limit Setting Function	2975
30.8.3.3	Capture Function	2977
30.8.3.4	MIN Guard Function.....	2978
30.9	Timer E	2988
30.9.1	Overview of Operation	2988
30.9.2	Registers Related to Timer E.....	2990
30.9.2.1	TSTRE — Timer Start Register E.....	2990
30.9.2.2	SBRLNE — Subblock Reload Enable Register E	2991
30.9.2.3	SSTREx — Subblock Starting Registers Ex.....	2992
30.9.2.4	PSCREx — Prescaler Registers Ex	2994
30.9.2.5	PSCCRExy — Prescaler Channel Registers Exy.....	2995
30.9.2.6	TCREx — Timer Control Register Ex	2996
30.9.2.7	RLDCREx — Reload Control Registers Ex	2997
30.9.2.8	POECREx — Output Shutoff Control Register Ex.....	2998
30.9.2.9	SOLVLEx — Output Shutoff Level Setting Register Ex.....	2999
30.9.2.10	TSREx — Timer Status Registers Ex.....	3000
30.9.2.11	TSCREx — Timer Status Clear Registers Ex.....	3002
30.9.2.12	TIEREx — Timer Interrupt Enable Registers Ex	3004
30.9.2.13	TOCREx — Timer Output Control Registers Ex.....	3006
30.9.2.14	FCTRGEEx — Forced Compare Match Trigger Registers Ex.....	3007
30.9.2.15	TCNTExy — Timer Counters Exy	3008
30.9.2.16	CYLRExy — Cycle-Setting Registers Exy	3009
30.9.2.17	DTRExy — Duty Cycle Setting Registers Exy	3010
30.9.2.18	CRLDExy — Cycle Reload Registers Exy	3011
30.9.2.19	DRLDExy — Duty Reload Registers Exy	3012
30.9.3	Details of Operation	3013

30.10	Timer F	3018
30.10.1	Overview of Operation	3018
30.10.2	Registers Related to Timer F	3020
30.10.2.1	TSTRF — Timer Start Register F	3020
30.10.2.2	NCMCR1F — Noise Cancellation Mode Channel Register 1F	3021
30.10.2.3	NCMCR2F — Noise Cancellation Mode Channel Register 2F	3023
30.10.2.4	NCCRF — Noise Canceller Control Register F	3024
30.10.2.5	PVFCRF — Private Function Control Register F	3026
30.10.2.6	TCR1Fx — Timer Control Registers 1Fx	3027
30.10.2.7	TCR2Fx — Timer Control Registers 2Fx	3029
30.10.2.8	TIERFx — Timer Interrupt Enable Registers Fx	3030
30.10.2.9	BKCRFx — Backup Control Register Fx	3032
30.10.2.10	TSRFx — Timer Status Registers Fx	3034
30.10.2.11	TSCRFx — Timer Status Clear Register Fx	3036
30.10.2.12	ECNTAFx — Timer Measurement Counters AFx	3038
30.10.2.13	ECNTBFx — Event Counters Fx	3039
30.10.2.14	ECNTCFx — Time Measurement Counters CFx	3040
30.10.2.15	GRAFx — General Registers AFx	3041
30.10.2.16	BGRAFx — Backup General Registers AFx	3042
30.10.2.17	GRBFx — General Registers BFx	3043
30.10.2.18	GRCFx — General Registers CFx	3044
30.10.2.19	BGRCFx — Backup General Registers CFx	3045
30.10.2.20	GRDFx — General Registers DFx	3046
30.10.2.21	BGRDFx — Backup General Registers DFx	3047
30.10.2.22	CDRFx — Capture Output Registers Fx	3048
30.10.2.23	NCNTFAx — Noise Canceller Counters FAX	3049
30.10.2.24	NCNTFBx — Noise Canceller Counters FBx	3051
30.10.2.25	NCRFAx — Noise Cancel Registers FAX	3053
30.10.2.26	NCRFBx — Noise Cancel Registers FBx	3054
30.10.3	Details of Operation	3056
30.10.3.1	Edge Counting in a Given Time	3056
30.10.3.2	Effective Edge Interval Counting	3058
30.10.3.3	Measurement of Time during High/Low Input Levels	3059
30.10.3.4	Measurement of PWM Input Waveform Timing	3061
30.10.3.5	Rotation Speed/Pulse Measurement	3064
30.10.3.6	Up/Down Event Count	3066
30.10.3.7	Four-time Multiplication Event Count	3068
30.10.3.8	Overflow and Underflow	3069
30.10.3.9	Simultaneous Access of Multiple Registers	3070
30.11	Timer G	3071
30.11.1	Overview of Operation	3071
30.11.2	Registers Related to Timer G	3072
30.11.2.1	TSTRG — Timer Start Register G	3072
30.11.2.2	TIERG — Timer Interrupt Enable Register G	3073
30.11.2.3	RLDCRG — Reload Control Register G	3074
30.11.2.4	TCRGx — Timer Control Register Gx	3075
30.11.2.5	TSRGx — Timer Status Registers Gx	3076
30.11.2.6	TSCRGx — Timer Status Clear Register Gx	3078
30.11.2.7	TCNTGx — Timer Counters Gx	3079
30.11.2.8	OCRGx — Compare Match Registers Gx	3080
30.11.2.9	RLDGx — Reload Registers Gx	3081

30.11.3	Details of Operation	3082
30.12	Automatic Switching of DMA and A/D Requests	3084
30.12.1	Operation	3084
30.12.2	Registers Related to Automatic Switching of DMA and AD Requests	3085
30.12.2.1	TRGSRDMA0 — Trigger Status Register DMA0	3085
30.12.2.2	TRGSELDMA00 — Trigger Select Register DMA00	3086
30.12.2.3	TRGSELDMA01 — Trigger Select Register DMA01	3087
30.12.2.4	TRGSELAD — Trigger Select Register AD	3088
30.12.2.5	TRGSRDMA1 — Trigger Status Register DMA1	3089
30.12.2.6	TRGSELDMA10 — Trigger Select Register DMA10	3090
30.12.2.7	TRGSELDMA11 — Trigger Select Register DMA11	3091
30.12.3	Details of Operation	3092
30.12.3.1	Data DMA Trigger	3092
30.12.3.2	Count DMA Trigger	3093
30.12.3.3	SAR-AD Trigger	3094
30.13	Trigger Select Function	3095
30.13.1	Overview of Operation	3095
30.13.2	Registers Related to Trigger Select Function	3095
30.13.2.1	ATUINTSELA — ATU TimerA Interrupt Selection Control Register	3095
30.13.2.2	ATUINTSELD0 — ATU TimerD Interrupt Selection Control Register 0	3096
30.13.2.3	ATUINTSELD1 — ATU TimerD Interrupt Selection Control Register 1	3097
30.13.2.4	ATUINTSELD2 — ATU TimerD Interrupt Selection Control Register 2	3098
30.13.2.5	ATUINTSELD3 — ATU TimerD Interrupt Selection Control Register 3	3099
30.13.2.6	ATUINTSELD4 — ATU TimerD Interrupt Selection Control Register 4	3100
30.13.2.7	ATUDMASELB — ATU TimerB DMA Selection Control Register	3101
30.13.2.8	ATUDMASELCD0 — ATU TimerC/TimerD DMA Selection Control Register 0 ...	3102
30.13.2.9	ATUDMASELCD1 — ATU TimerC/TimerD DMA Selection Control Register 1 ...	3105
30.13.2.10	ATUDMASELCD2 — ATU TimerC/TimerD DMA Selection Control Register 2 ...	3108
30.13.2.11	ATUDMASELCD3 — ATU TimerC/TimerD DMA Selection Control Register 3 ...	3111
30.13.2.12	ATUDMASELCD4 — ATU TimerC/TimerD DMA Selection Control Register 4 ...	3113
30.13.2.13	ATUDMASELCD5 — ATU TimerC/TimerD DMA Selection Control Register 5 ...	3116
30.13.2.14	ATUDMASELCD6 — ATU TimerC/TimerD DMA Selection Control Register 6 ...	3119
30.13.2.15	ATUDMASELCD7 — ATU TimerC/TimerD DMA Selection Control Register 7 ...	3122
30.13.2.16	ATUDMASELE — ATU TimerE DMA Selection Control Register	3124
30.13.2.17	ATUDFEENTQi — ATU to DFE Capture Trigger i Input Enable	3125
30.13.2.18	ATUDFESELi — ATU to DFE Capture Trigger i Input Select	3126
30.13.2.19	ATUDFESELD1T — ATU to DFE Timer FIFO Capture Trigger Select	3128
30.13.2.20	ATUDSSELDSTS — ATU to DSADC Read Gate Trigger Select	3129
30.13.2.21	ATUCASELCATS — ATU to CADC Read Gate Trigger Select	3131
30.13.2.22	ATUP5SSEL — ATU to PSI5S Time Stamp Trigger Select	3132
30.13.3	Details of Operation	3134
30.13.3.1	Interrupt signal communicated to INTC is selected.	3134
30.13.3.2	Interrupt signal communicated to sDMA/DTS is selected.	3136
30.13.3.3	“Timer DFE Capture trigger i (i = 0 to 2)” communicated to DFE selected.	3140
30.13.3.4	“Timer FIFO Capture trigger” communicated to DFE is selected.	3141
30.13.3.5	“read gate” communicated to DSADC is selected.	3142
30.13.3.6	“read gate” communicated to CADC is selected.	3143
30.13.3.7	“time stamp” and “sync pulse” communicated to PSI5S is selected.	3144
30.14	Usage Notes	3145
30.14.1	Input Capture Conflict Operation	3145

30.14.1.1	Conflict between Writing to General Register and Input Capture.....	3145
30.14.1.2	Conflict between Writing to Counter and Input Capture	3146
30.14.1.3	Conflict between Setting and Clearing of Input Capture Status Flag	3147
30.14.2	Compare Match Conflict Operation.....	3148
30.14.2.1	Compare Match Signal in Cases of Consecutive Comparison.....	3148
30.14.2.2	Conflict between Writing to Compare Match General Register and Compare Match	3149
30.14.2.3	Conflict between CYLRExy Write and Cycle Match with TCNTExy	3150
30.14.2.4	Conflict between DTRExy Write and Cycle Match with TCNTExy	3150
30.14.2.5	Conflict between Writing to Counter and Compare Match	3151
30.14.2.6	Conflict between Writing to Counter and Counter Clearing by Compare Match ..	3152
30.14.2.7	Conflict between Writing to TCNTExy and Counter Clearing by Cycle Match	3153
30.14.2.8	Conflict between Setting and Clearing of Compare Match Status Flag	3154
30.14.2.9	Conflict between Setting of Cycle Match Status Flag and Writing 1 to the Status Clear Register.....	3155
30.14.2.10	Conflict between Writing 0 to TCNTExy and Cycle Match	3156
30.14.3	Load/Reload Conflict Operation.....	3157
30.14.3.1	Conflict between Data Transfer and Writing to Transfer Destination Register.....	3157
30.14.3.2	Conflict between Data Transfer and Writing to Transfer Source Register	3158
30.14.4	Counter Conflict Operation	3159
30.14.4.1	Conflict between Writing to Counter and Count-Up/Count-Down	3159
30.14.4.2	Conflict between Count-Up and Counter Clearing	3159
30.14.4.3	Conflict between Writing to Counter and Overflow.....	3160
30.14.4.4	Conflict between Setting and Clearing of Overflow Status Flag.....	3161
30.14.4.5	Conflict between Overflow and Counter Clearing by Compare Match	3162
30.14.5	Noise Canceler Conflict Operation.....	3163
30.14.5.1	Conflict between Writing to Noise Canceler Counter and Compare Match with Noise Canceler Register	3163
30.14.5.2	Conflict between Writing to Noise Canceler Register and Compare Match with Noise Canceler Counter.....	3164
30.14.6	Timer Down Counter Dxy Conflict Operation.....	3165
30.14.6.1	Conflict between Writing to DCNTDxy Counter and Count-Down.....	3165
30.14.6.2	Conflict between Writing to DCNTDxy Counter and Underflow	3165
30.14.6.3	Conflict between Writing to DCNTDxy Counter and Compare Match B (Counter Stop Trigger).....	3166
30.14.6.4	Conflict between Setting of Underflow Status Flag and Clearing by Writing 1 to Status Clear Register.....	3167
30.14.6.5	TODxyB Output upon Occurrence of Down Counter Start Trigger When Down Counter Value is 0000 0000 _H	3168
30.14.6.6	TODxyB Output at Simultaneous Occurrence of Start Trigger and Stop Trigger for Down Counter	3168
30.14.6.7	Conflict between Down Counter Start Trigger and Underflow.....	3168
30.14.7	Coordinated Operation of Timers A, B, and D	3169
30.14.7.1	Conflict Between Counter Clearing of TCNT1Dx and TCNT2Dx and Compare Match	3169
30.14.7.2	Conflict between Writing to TCNT1Dx/TCNT2Dx Counter and Counter Clearing by Timer B.....	3170
30.14.7.3	Conflict between TCNT1Dx/TCNT2Dx Counter Overflow and Counter Clearing by Timer B.....	3170
30.14.7.4	Conflict between TCNT1Dx Clearing by Clear Signal from Timer B and Input Capture to OSBRDx.....	3171

Section 31	Generic Timer Module (GTM)	3172
31.1	GTM Features	3172
31.1.1	IP Version	3172
31.1.2	Number of Sub-Units and Channels	3172
31.1.3	Register/RAM Base Address	3174
31.1.4	Clock Supply	3174
31.1.5	Interrupt Requests	3175
31.1.6	Reset Sources	3179
31.1.7	External Input/Output Signals	3179
31.2	Overview	3180
31.2.1	Functional Overview	3180
31.2.2	Block Diagram	3182
31.3	Registers	3183
31.3.1	List of Registers	3183
31.3.2	GTM_IRQ_SEL _{ij} — GTM Interrupt Selection Control Register ij	3191
31.3.3	GTM_IRQ_SEL _{8j} — GTM Interrupt Selection Control Register 8j	3192
31.3.4	GTM_IRQ_DPLL_SEL _j — GTM Interrupt DPLL Selection Register j	3193
31.3.5	GTM_DMA_SEL _{ij} — GTM DMA Selection Control Register ij	3196
31.3.6	GTM_DMA_SEL _{8j} — GTM DMA Selection Control Register 8j	3197
31.3.7	GTM_DMA_COM_SEL _j — GTM DMA Common Selection Function Register j	3198
31.3.8	GTM_ADCI_CHSEL _n — GTM ADCI channel selection register n	3199
31.4	Operation	3200
31.4.1	Peripheral Bus Access	3200
31.4.2	Reset	3204
31.4.3	Interconnections of Interrupt Factors	3205
31.4.4	Outputs to AD	3211
31.4.4.1	Outputs to ADCH	3211
31.4.4.2	Outputs to DSADC	3211
31.4.4.3	Outputs to CADC	3211
31.4.5	Outputs to ENCA	3212
31.4.6	Outputs to DFE	3212
31.4.7	Outputs to PSI5-S	3212
31.4.8	Peripheral Inputs	3213
31.4.8.1	Inputs from PORT	3213
31.4.8.2	Inputs from ADCH	3213
31.4.8.3	Inputs from DSADC	3213
31.4.8.4	Inputs from CADC	3213
31.4.8.5	Input from ENCA	3213
31.4.8.6	Inputs from DFE	3214
31.4.9	ADC Interface	3214
31.4.10	Safety Mechanism	3216
31.4.11	GTM Debug Function	3216

Section 32	Encoder Timer (ENCA)	3217
32.1	Features of ENCA	3217
32.1.1	Number of Units	3217
32.1.2	Register Base Addresses	3218
32.1.3	Clock Supply	3218
32.1.4	Interrupt Requests	3219
32.1.5	Reset Sources	3220
32.1.6	External Input/Output Signals	3220
32.2	Overview	3221
32.2.1	Functional Overview	3221
32.2.2	Block Diagram	3222
32.3	Registers	3223
32.3.1	List of Registers	3223
32.3.2	Details of Registers	3224
32.3.2.1	ENCAnCTL — ENCAAn Control Register	3224
32.3.2.2	ENCAnIOC0 — ENCAAn I/O Control Register 0	3226
32.3.2.3	ENCAnIOC1 — ENCAAn I/O Control Register 1	3227
32.3.2.4	ENCAnFLG — ENCAAn Status Flag Register	3229
32.3.2.5	ENCAnFGC — ENCAAn Status Flag Clear Register	3230
32.3.2.6	ENCAnCCR0 — ENCAAn Capture/Compare Register 0	3231
32.3.2.7	ENCAnCCR1 — ENCAAn Capture/Compare Register 1	3232
32.3.2.8	ENCAnCNT — ENCAAn Counter Register	3233
32.3.2.9	ENCAnTE — ENCAAn Timer Enable Status Register	3234
32.3.2.10	ENCAnTS — ENCAAn Timer Start Trigger Register	3235
32.3.2.11	ENCAnTT — ENCAAn Timer Stop Trigger Register	3236
32.3.2.12	ENCAnEMU — ENCAAn Emulation Register	3237
32.4	Operation	3238
32.4.1	Timer Counter Operation	3238
32.4.2	Up/Down Control of Timer Counter	3239
32.4.2.1	When the ENCAAnUDS1/ENCAAnUDS0 Bits in the ENCAAnCTL Register = 00 _B	3240
32.4.2.2	When the ENCAAnUDS1/ENCAAnUDS0 Bits in the ENCAAnCTL Register = 01 _B	3241
32.4.2.3	When the ENCAAnUDS1 and ENCAAnUDS0 Bits in the ENCAAnCTL Register = 10 _B	3242
32.4.2.4	When ENCAAnUDS1/ENCAAnUDS0 Bits in the ENCAAnCTL Register = 11 _B	3243
32.4.3	Timer Counter Clear Control by Encoder Input	3244
32.4.3.1	Clearing Method when ENCAAnSCE = 0	3244
32.4.3.2	Clearing Method when ENCAAnSCE = 1	3244
32.4.4	Functions of ENCAAnCCR0	3245
32.4.4.1	Compare Function	3245
32.4.4.2	Capture Function	3245
32.4.5	Functions of ENCAAnCCR1	3246
32.4.5.1	Compare Function	3246
32.4.5.2	Capture Function	3247
32.4.5.3	Timer Counter Clearing upon Compare Register Match	3248
32.5	ENCAAn Setting Sequences	3249
32.5.1	ENCAAn Setting Procedure	3249
32.5.1.1	Initial Setting Procedure for the Counter	3250

32.5.1.2	Initial Setting Procedure for Counter Clearing	3250
32.5.1.3	Setting Procedure for ENCA _n CCR0 Register	3251
32.5.1.4	Setting Procedure for ENCA _n CCR1 Register	3251
32.6	Timing Chart	3252
32.6.1	Overflow Occurrence and Overflow Flag Clear Operation	3252
32.6.2	Underflow Occurrence and Underflow Flag Clear Operation	3253
32.6.3	Count Clearing and Capture Operation by Encoder Clear Input (ENCA _n EC Pin)	3254
32.6.4	Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCA _n EC Pin).....	3255
32.6.5	Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCA _n EC Pin).....	3256
32.6.6	Overflow Operation Immediately after Startup.....	3257
32.6.7	Underflow Operation Immediately after Startup.....	3258
32.6.8	Using the ENCA _n LDE Function Immediately after Startup.....	3259
32.6.9	ENCA _n LDE Function (Loading Count Value).....	3260
32.6.10	Conflict between ENCA _n LDE Function (Loading Count Value) and Rewrite of ENCA _n CCR0 Register	3262
32.6.11	Conflict between ENCA _n LDE Function (Loading Count Value) and Clear Operation by Encoder Clear Input (ENCA _n EC Pin)	3263
32.6.12	Up-count after Conflict between ENCA _n LDE Function (Loading Count Value) and Encoder Clearing	3265
32.6.13	Capture Operation between Count Clocks (ENCA _n CCR1).....	3266
32.6.14	Capture Operation between Count Clocks (ENCA _n CCR0).....	3267
32.6.15	Encoder Operation when Compare Match Clear Control is Enabled and ENCA _n CTS = 0	3268
32.6.16	Encoder Operation when Compare Match Clear Control is Enabled and ENCA _n CTS = 1	3269
32.6.17	Encoder Operation when Compare Match Clear Control is Disabled	3270
32.6.18	Capture Operation Performed upon Clearing by ENCA _n EC, ENCA _n E0, or ENCA _n E1 when ENCA _n SCE = 1	3271
32.6.18.1	Accompanying Capture Operation.....	3271
32.6.18.2	When the Timing of the ENCA _n EC Input is Later than that of the ENCA _n E1 Input during Up-count (When ENCA _n ACL = 1, ENCA _n BCL = 0, ENCA _n ZCL = 1, and ENCA _n UDS[1:0] = 11 _B)	3272
32.6.18.3	When the Timing of the ENCA _n EC Input is the Same as that of the ENCA _n E1 Input during Up-count (When ENCA _n ACL = 1, ENCA _n BCL = 0, ENCA _n ZCL = 1, and ENCA _n UDS[1:0] = 11 _B)	3273
32.6.18.4	When the Timing of the ENCA _n EC Input is Earlier than that of the ENCA _n E1 Input during Up-count (When ENCA _n ACL = 1, ENCA _n BCL = 0, ENCA _n ZCL = 1, and ENCA _n UDS[1:0] = 11 _B)	3273
32.6.18.5	When the Timing of the ENCA _n EC Input is Later than that of the ENCA _n E1 Input during Down-count (When ENCA _n ACL = 1, ENCA _n BCL = 0, ENCA _n ZCL = 1, and ENCA _n UDS[1:0] = 11 _B)	3274
32.6.19	Capture Operation Performed upon Clearing by ENCA _n EC when ENCA _n SCE = 0.....	3275

Section 33	Peripheral Interconnection (PIC)	3276
33.1	Features of PIC	3276
33.1.1	Number of Units	3276
33.1.2	Register Base Addresses	3276
33.1.3	Clock Supply	3276
33.1.4	Reset Sources	3277
33.1.5	External Input/Output Signals	3277
33.2	Timer Synchronization Function (PIC1)	3278
33.2.1	Overview	3278
33.2.1.1	Functional Overview	3278
33.2.1.2	Block Diagram	3278
33.2.2	Registers	3279
33.2.2.1	List of Registers	3279
33.2.2.2	PIC1SST — Simultaneous Start Trigger Control Register	3279
33.2.2.3	PIC1SSER0 — Simultaneous Start Control Register 0	3280
33.2.3	Operation	3281
33.2.3.1	Timer Synchronization	3281
33.3	Trigger Selection Function (PIC2)	3282
33.3.1	Overview	3282
33.3.1.1	Functional Overview	3282
33.3.1.2	Block Diagrams	3283
33.3.2	Registers	3285
33.3.2.1	List of Registers	3285
33.3.2.2	PIC2ADCHnTSELj — AD Converter n Trigger Selection Control Register j (n = 0 to 3 [*] ; j = 0 to 4)	3291
33.3.2.3	PIC2ADCHnEDGSEL — AD Converter n [*] Trigger Edge Selection Control Register (n = 0 to 3 [*])	3292
33.3.2.4	PIC2ADTEN5nj — AD Converter Trigger Output Control Register 5nj (n = 0 to 3 [*] ; j = 0 to 4)	3293
33.3.2.5	PIC2ADTEN6nj — AD Converter Trigger Output Control Register 6nj (n = 0 to 3 [*] ; j = 0 to 4)	3296
33.3.2.6	PIC2ADTEN7nj — AD Converter Trigger Output Control Register 7nj (n = 0 to 3 [*] ; j = 0 to 4)	3298
33.3.2.7	PIC2ADTEN8nj — AD Converter Trigger Output Control Register 8nj (n = 0 to 3 [*] ; j = 0 to 4)	3300
33.3.2.8	PIC2DSADTEN0n0 — DSADC Start Trigger Output Control Register n0 (n = 0 to 5)	3303
33.3.2.9	PIC2DSADTEN0n1 — DSADC Start Trigger Output Control Register n1 (n = 0 to 5)	3304
33.3.2.10	PIC2DSADTEN0n2 — DSADC Start Trigger Output Control Register n2 (n = 0 to 5)	3306
33.3.2.11	PIC2DSADTEN0n3 — DSADC Start Trigger Output Control Register n3 (n = 0 to 5)	3308
33.3.2.12	PIC2DSADTEN1n0 — DSADC Stop Trigger Output Control Register n0 (n = 0 to 5)	3310
33.3.2.13	PIC2DSADTEN1n1 — DSADC Stop Trigger Output Control Register n1 (n = 0 to 5)	3311
33.3.2.14	PIC2DSADTEN1n2 — DSADC Stop Trigger Output Control Register n2 (n = 0 to 5)	3313

33.3.2.15	PIC2DSADTEN1n3 — DSADC Stop Trigger Output Control Register n3 (n = 0 to 5).....	3315
33.3.2.16	PIC2DSADCATSEL0 — DSADC Trigger Selection Control Register 0	3317
33.3.2.17	PIC2DSADCATSEL1 — DSADC Trigger Selection Control Register 1	3319
33.3.2.18	PIC2ENCAnTSELj — Encoder Timer n Trigger Selection Control Register j (n = 0 to 1; j = 0 to 1).....	3321
33.3.2.19	PIC2CADTEN000 — CADC Start Trigger Output Control Register 00	3323
33.3.2.20	PIC2CADTEN001 — CADC Start Trigger Output Control Register 01	3324
33.3.2.21	PIC2CADTEN002 — CADC Start Trigger Output Control Register 02	3326
33.3.2.22	PIC2CADTEN003 — CADC Start Trigger Output Control Register 03	3328
33.3.2.23	PIC2CADTEN100 — CADC Stop Trigger Output Control Register 00	3330
33.3.2.24	PIC2CADTEN101 — CADC Stop Trigger Output Control Register 01	3331
33.3.2.25	PIC2CADTEN102 — CADC Stop Trigger Output Control Register 02	3333
33.3.2.26	PIC2CADTEN103 — CADC Stop Trigger Output Control Register 03	3335
33.3.2.27	PIC2CADCATSEL0 — CADC Trigger Selection Control Register 0	3337
33.3.2.28	PIC2CADCATSEL1 — CADC Trigger Selection Control Register 1	3338
33.3.2.29	PIC2ADTCFGx — A/D Converter Trigger Output Configuration Register x (x = 0 to 4).....	3339
33.3.2.30	PIC2DSADTCFGx — DSADC Start / Stop Trigger Output Configuration Register x (x = 0 to 2)	3340
33.3.2.31	PIC2CADTCFG0 — CADC Start / Stop Trigger Output Configuration Register 0.....	3341
33.3.2.32	PIC2DSADTSENx — DSADC Read Gate Trigger Selection Control Register x (x = 0 to 1).....	3342
33.3.2.33	PIC2CADTSEN0 — CADC Read Gate Trigger Selection Control Register 0.....	3344
33.3.2.34	PIC2DFETSENx — DFE Timer Trigger Selection Control Register x (x = 0 to 11).....	3345
33.3.2.35	PIC2DFETEDGSEL0 — DFE Timer Trigger Edge Selection Control Register 0.	3347
33.3.2.36	PIC2DFETEDGSEL1 — DFE Timer Trigger Edge Selection Control Register 1.	3349
33.3.2.37	PIC2DFETEDGSEL2 — DFE Timer Trigger Edge Selection Control Register 2.	3351
33.3.2.38	PIC2PSI5SENx — PSI5-S Sync Output Control Register x (x = 0 to 3)	3352
33.3.2.39	PIC2GTMINENx — GTM Timer Input Module (TIM) Source Select Register x (x = 0, 2, 4, 6, 8, 10, 12, 14).....	3354
33.3.2.40	PIC2GTMINENx — GTM Timer Input Module (TIM) Source Select Register x (x = 1, 3, 5, 7, 9, 11, 13, 15).....	3357
33.3.2.41	PIC2ENCATCFG0 — Encoder Timer Trigger Output Configuration Register 0 ..	3360
33.3.2.42	PIC2ENCATCFG1 — Encoder Timer Trigger Output Configuration Register 1 ..	3361
33.3.2.43	PIC2ENCAISEN0 — Encoder Input Selection Control Register 0	3362
33.3.3	Function	3363
33.3.3.1	ADCH Trigger Selection Function.....	3363
33.3.3.2	DSADC Trigger Selection Function	3365
33.3.3.3	CADC Trigger Selection Function.....	3367
33.3.3.4	ENCA Trigger Selection Function.....	3369
33.3.3.5	DFE Trigger Selection Function.....	3370
33.3.3.6	PSI5-S Timestamp and the Sync Pulse Signal Selection Function.....	3371
33.3.3.7	GTM Timer Input (TIM) Selection Function	3372
33.3.3.8	ENCA Encoder Input Selection.....	3373
Section 34 Analog to Digital Converter (ADCH).....		3374
34.1	Features.....	3374
34.1.1	Number of Units and Channels.....	3374
34.1.2	Register Base Address	3377

34.1.3	Clock Supply	3377
34.1.4	Interrupt Requests.....	3378
34.1.5	Reset Source	3389
34.1.6	External Input/Output Signals	3390
34.2	Overview.....	3398
34.2.1	Functional Overview.....	3398
34.2.2	Block Diagram.....	3401
34.3	Registers.....	3405
34.3.1	List of Registers	3405
34.3.2	ADCH Common Registers	3408
34.3.2.1	ADCH0ADSYNSTCR — A/D Synchronization Start Control Register	3408
34.3.2.2	ADCH0ADTSYNSTCR — A/D Timer Synchronization Start Control Register	3409
34.3.2.3	ADCH0VMONVDCR1 — Voltage Monitor Voltage Divider Control Register 1	3410
34.3.2.4	ADCH0VMONVDCR2 — Voltage Monitor Voltage Divider Control Register 2	3411
34.3.3	ADCH Specific Registers (Virtual Channel, Control, Safety-related).....	3412
34.3.3.1	ADCHnVCRj — Virtual Channel Register j.....	3412
34.3.3.2	ADCHnDRj — Data Register j	3415
34.3.3.3	ADCHnDIRj — Data Supplementary Information Register j.....	3419
34.3.3.4	ADCHnADHALTR — A/D Halt Register	3420
34.3.3.5	ADCHnADCR1 — A/D Control Register 1.....	3421
34.3.3.6	ADCHnMPXCURCR — MPX Current Control Register	3422
34.3.3.7	ADCHnMPXINTER — MPX Interrupt Enable Register	3423
34.3.3.8	ADCHnMPXCURR — MPX Current Register	3424
34.3.3.9	ADCHnMPXOWR — MPX Optional Wait register.....	3425
34.3.3.10	ADCHnMPXCMDR — MPX Command Information Register	3426
34.3.3.11	ADCHnADCR2 — A/D Control Register 2.....	3427
34.3.3.12	ADCHnDFASENTSGER — DFE/ASF Entry Scan Group Enable Register	3428
34.3.3.13	ADCHnADENDP — A/D Conversion Monitor Virtual Channel Pointer	3429
34.3.3.14	ADCHnSFTCR — Safety Control Register.....	3430
34.3.3.15	ADCHnTDCR — Pin-Level Self-Diagnosis Control Register.....	3431
34.3.3.16	ADCHnODCR — Wiring-break Detection Control Register.....	3432
34.3.3.17	ADCHnECR — Error Clear Register	3433
34.3.3.18	ADCHnOWER — Overwrite Error Register	3434
34.3.3.19	ADCHnPER — Parity Error Register.....	3435
34.3.3.20	ADCHnVCULLMTBR0 to 6 — Virtual Channel Upper/Lower Limit Table Register 0 to 6.....	3436
34.3.3.21	ADCHnVCLMINTER1 — Virtual Channel Upper/Lower Limit Error Interrupt Enable Register 1	3437
34.3.3.22	ADCHnVCLMINTER2 — Virtual Channel Upper/Lower Limit Error Interrupt Enable Register 2	3438
34.3.3.23	ADCHnVCLMSR1 — Virtual Channel Upper/Lower Limit Excess Status Register 1.....	3439
34.3.3.24	ADCHnVCLMSR2 — Virtual Channel Upper/Lower Limit Excess Status Register 2.....	3440
34.3.3.25	ADCHnVCLMSCR1 — Virtual Channel Upper/Lower Limit Excess Status Clear Register 1	3441
34.3.3.26	ADCHnVCLMSCR2 — Virtual Channel Upper/Lower Limit Excess Status Clear Register 2	3442
34.3.4	Scan Group Specific Registers	3443
34.3.4.1	ADCHnSGSTCRx — Scan Group x Start Control Register	3443

34.3.4.2	ADCHnSGSTPCR _x — Scan Group x Stop Control Register	3444
34.3.4.3	ADCHnADTSTCR _y — A/D Timer y Start Control Register	3445
34.3.4.4	ADCHnADTENDCR _y — A/D Timer y End Control Register.....	3446
34.3.4.5	ADCHnSGCR _x — Scan Group x Control Register.....	3447
34.3.4.6	ADCHnSGVCSP _x — Scan Group x Start Virtual Channel Pointer	3449
34.3.4.7	ADCHnSGVCEP _x — Scan Group x End Virtual Channel Pointer	3450
34.3.4.8	ADCHnSGMCYCR _x — Scan Group x Multicycle Register	3451
34.3.4.9	ADCHnSGSR _x — Scan Group x Status Register	3452
34.3.4.10	ADCHnADTIPR _y — A/D Timer y Initial Phase Register.....	3453
34.3.4.11	ADCHnADTPRR _y — A/D Timer y Cycle Register.....	3454
34.3.4.12	ADCHnSGVCPR _x — Scan Group x Virtual Channel Pointer Register	3455
34.3.4.13	ADCHnSGVCOWR _x — Scan Group x Virtual Channel Optional Waiting Times Register.....	3456
34.4	Operation	3457
34.4.1	Setting Procedure	3457
34.4.1.1	Initial Settings.....	3457
34.4.1.2	Startup Method	3459
34.4.1.3	Terminating Procedure	3460
34.4.1.4	Limitations of Setting Update.....	3464
34.4.2	Scan Mode.....	3465
34.4.2.1	Multicycle Scan Mode.....	3465
34.4.2.2	Continuous Scan Mode.....	3466
34.4.3	Suspend Method.....	3467
34.4.3.1	Synchronous Suspend.....	3468
34.4.3.2	Asynchronous Suspend	3469
34.4.3.3	Synchronous/Asynchronous Mixture Type Suspend.....	3470
34.4.4	Conversion Class.....	3471
34.4.4.1	Normal A/D Conversion	3471
34.4.4.2	A/D Converter Self-Diagnosis.....	3471
34.4.4.3	Example of Addition Mode A/D Conversion Operation.....	3472
34.4.4.4	Normal A/D Conversion with the MPX and Normal A/D Conversion with the MPX in Addition Mode	3473
34.4.5	Trigger Functions	3476
34.4.5.1	S/W Trigger.....	3476
34.4.5.2	H/W Trigger.....	3478
34.4.5.3	A/D Timer Trigger	3480
34.4.6	Interrupt Request Functions.....	3481
34.4.6.1	Scan End Interrupt Request	3481
34.4.6.2	A/D Error Interrupt Request	3483
34.4.6.3	Overwrite Error.....	3484
34.4.6.4	ID Error	3484
34.4.6.5	Parity Error Interrupt Request.....	3486
34.4.6.6	MPX Interrupt Request	3487
34.4.6.7	Virtual Channel Upper/Lower Limit Error Interrupt	3488
34.4.7	DFE/GTM/ASF Entry Function.....	3491
34.4.8	Monitoring Function Using the A/D Conversion Monitor Pin	3492
34.4.9	Virtual Channel Upper/Lower Limit Excess Notice to ADC VMON Secondary Error Generator (AVSEG), or to ADC Boundary Flag Generator (ABFG).....	3493
34.4.9.1	Setting Registers.....	3493
34.4.9.2	Table Registers.....	3494

34.4.9.3	Interface Signals	3495
34.4.9.4	Operating Timing	3496
34.4.10	Wiring-Break Detection Function	3498
34.4.11	Self-Diagnostic Functions	3499
34.4.11.1	Pin-Level Self-Diagnosis.....	3499
34.4.11.2	A/D Converter Self-Diagnosis.....	3500
34.4.11.3	Wiring-Break Detection Self-Diagnosis.....	3501
34.4.11.4	Secondary Power Supply Voltage Monitor Self-Diagnosis.....	3502
34.4.12	Timing Regulations	3503
34.4.13	Optional Waiting Times Insertion Function of Between Virtual Channels	3504
34.4.13.1	Waiting Times Insertion in Normal A/D Conversion	3504
34.4.13.2	Waiting Time Insertion in Synchronous Suspend Movement.....	3506
34.4.13.3	Waiting Times Insertion in Asynchronous Suspend Movement	3507
34.4.14	Example of Scan Group Stop	3508
34.4.14.1	Example of All Scan Group Stop (A/D Halt)	3508
34.4.14.2	Example of Scan Group x Stop	3509
34.4.15	Secondary Power Supply Voltage Monitor Function	3511
34.5	Definition of A/D Conversion Accuracy.....	3514
34.6	Usage Notes	3515
34.6.1	Notes on Using an External Analog Multiplexer	3515
34.6.2	Notes on Using Analog Input Pins	3516
34.6.3	Module Standby Function	3517
34.7	Integer/Floating-Point Conversion Module (IFC).....	3518
34.7.1	Overview	3518
34.7.1.1	Function Overview	3518
34.7.1.2	Block Diagram.....	3518
34.7.2	Registers	3520
34.7.2.1	List of Registers	3520
34.7.2.2	IFC Specific Registers	3521
34.8	ADC Summation Function (ASF).....	3522
34.8.1	Overview	3522
34.8.1.1	Function Overview	3522
34.8.1.2	Block Diagram.....	3522
34.8.2	Registers	3523
34.8.2.1	List of Registers	3523
34.8.2.2	ASF Specific Registers	3525
34.8.3	Operation	3534
34.8.3.1	Example of Accumulation Processing Operation	3536
34.9	ADC VMON Secondary Error Generator (AVSEG).....	3540
34.9.1	Overview	3540
34.9.1.1	Function Overview	3540
34.9.1.2	Block Diagram.....	3541
34.9.2	Registers	3542
34.9.2.1	List of Registers	3542
34.9.2.2	AVSEG Specific Registers.....	3543
34.9.3	Operation	3551
34.9.3.1	Noise Count Method	3551

34.9.3.2	Upper/Lower Error Pulse Output	3555
34.9.4	Use Method	3556
34.10	ADC Boundary Flag Generator (ABFG)	3557
34.10.1	Overview	3557
34.10.1.1	Function Overview	3557
34.10.1.2	Block Diagram	3558
34.10.2	Registers	3560
34.10.2.1	List of Registers	3560
34.10.2.2	ABFG Specific Registers	3561
34.10.3	Operation	3567
34.10.3.1	Noise Count Method	3567
34.10.3.2	Boundary Flag Output, Boundary Flag Pulse Output	3571
34.10.4	Use Method	3572
34.11	ADC Interrupt Router (AIR).....	3574
34.11.1	Overview	3574
34.11.1.1	Function Overview	3574
34.11.1.2	Block Diagram	3574
34.11.2	Registers	3575
34.11.2.1	List of Registers	3575
34.11.2.2	AIR Specific Registers	3576
Section 35	Delta-Sigma Analog to Digital Converter (DSADC).....	3586
35.1	Features.....	3586
35.1.1	Number of Units	3586
35.1.2	Register Base Addresses	3587
35.1.3	Clock Supply	3587
35.1.4	Interrupt Requests.....	3588
35.1.5	Reset Source	3588
35.1.6	External Input/Output Signals	3589
35.2	Overview	3590
35.2.1	Function Overview	3590
35.2.2	Block Diagram	3592
35.3	Registers.....	3593
35.3.1	List of Registers	3593
35.4	Delta-Sigma ADC Common Registers	3595
35.4.1	DSADCSYNSTCR — AD Synchronization Start Control Register	3595
35.4.2	DSADCADGCR — AD Global Control Register	3596
35.4.3	DSADCTDCR — Pin Level Self-Diagnosis Control Register.....	3597
35.5	Delta-Sigma ADC Unit-Specific Registers.....	3598
35.5.1	DSADCnVCRj — Virtual Channel Register j	3598
35.5.2	DSADCnDIRj — Data Supplementary Information Register j.....	3602
35.5.3	DSADCnADSTCR — AD Start Control Register	3605
35.5.4	DSADCnADENDCR — AD Stop Control Register	3606
35.5.5	DSADCnCLBSTCR — Calibration Start Control Register	3607

35.5.6	DSADCnCLBEDCR — Calibration Stop Control Register	3608
35.5.7	DSADCnADTCR — AD Conversion Trigger Control Register.....	3609
35.5.8	DSADCnUCR — Unit Control Register.....	3610
35.5.9	DSADCnVCPTRR — Virtual Channel Pointer Control Register.....	3612
35.5.10	DSADCnADSR — AD Conversion Status Register.....	3613
35.5.11	DSADCnUDPTRR — Unit Data Pointer Register.....	3614
35.5.12	DSADCnUDIR — Unit Data Supplementary Information Register	3615
35.5.13	DSADCnTSVAL — Timestamp Register	3616
35.5.14	DSADCnSFTCR — Safety Control Register	3617
35.5.15	DSADCnECR — Error Clear Register	3618
35.5.16	DSADCnER — Error Register	3619
35.5.17	DSADCnTDLVR — Pin Level Self-Diagnosis Level Register.....	3621
35.5.18	DSADCnULTBR0/1/2/3 — Upper-Limit/Lower-Limit Table Register 0/1/2/3	3622
35.6	Operation	3623
35.6.1	A/D Conversion Flow	3623
35.6.2	Filter Type Setting.....	3624
35.6.3	A/D Conversion Time	3625
35.6.4	Starting an A/D Conversion by an External Trigger.....	3629
35.6.5	Terminating an A/D Conversion by an External Trigger	3630
35.6.6	Starting an A/D Conversion by a Timer Trigger.....	3630
35.6.7	Terminating an A/D Conversion by a Timer Trigger	3630
35.6.8	Switching Virtual Channels via Start and End Triggers	3631
35.6.9	Calculating Correction Values by Using Calibration	3633
35.6.9.1	Calibration Time.....	3633
35.6.10	A/D Conversion Result Error Correction Processing.....	3634
35.6.11	A/D Conversion Monitoring Function Using Monitor Pins.....	3634
35.6.12	DMA Request Source	3635
35.6.13	AD Error Interrupt Request and AD Parity Error Interrupt Request.....	3635
35.6.13.1	Upper-Limit/Lower-Limit Errors.....	3636
35.6.13.2	Overwrite Error.....	3636
35.6.13.3	ID Error	3637
35.6.14	Boundary Flag Generating Signal Output Function	3638
35.6.15	Timestamp Function.....	3639
35.6.16	Wiring-Break Detection Function	3641
35.6.17	Self-Diagnosis	3642
35.6.17.1	Self-Diagnosis of Wiring-Break Detection Function.....	3642
35.6.17.2	A/D Self-Diagnosis Function.....	3643
35.6.17.3	Pin Level Self-Diagnosis Function	3643
35.6.18	DFE Control	3645
35.7	Usage Notes	3646
35.7.1	Notes on Using Analog Input Pins	3646
35.7.2	Module Standby Function	3646

Section 36	Cyclic Analog to Digital Converter (CADC)	3647
36.1	Features	3647
36.1.1	Number of Units and Channels	3647
36.1.2	Register Base Addresses	3648
36.1.3	Clock Supply	3648
36.1.4	Interrupt Requests	3648
36.1.5	Reset Sources	3648
36.1.6	External Input/Output Signals	3649
36.2	Overview	3650
36.2.1	Function Overview	3650
36.2.2	Block Diagram	3652
36.3	Registers	3653
36.3.1	List of Registers	3653
36.4	Cyclic ADC Common Registers	3654
36.4.1	CADCADGCR — AD Control Register	3654
36.4.2	CADCTDCR — Pin Level Self-Diagnosis Control Register	3655
36.5	Cyclic ADC Unit-Specific Registers	3656
36.5.1	CADCnVCRj — Virtual Channel Register j	3656
36.5.2	CADCnDIRj — Data Supplementary Information Register j	3659
36.5.3	CADCnADSTCR — AD Start Control Register	3662
36.5.4	CADCnADENDCR — AD Stop Control Register	3663
36.5.5	CADCnCLBSTCR — Calibration Start Control Register	3664
36.5.6	CADCnCLBEDCR — Calibration Stop Control Register	3665
36.5.7	CADCnADTCR — AD Conversion Trigger Control Register	3666
36.5.8	CADCnUCR — Unit Control Register	3667
36.5.9	CADCnVCPTRR — Virtual Channel Pointer Register	3669
36.5.10	CADCnADSR — AD Conversion Status Register	3670
36.5.11	CADCnUDPTRR — Unit Data Pointer Register	3671
36.5.12	CADCnUDIR — Unit Data Supplementary Information Register	3672
36.5.13	CADCnSMPCR — Sampling Rate Control Register	3673
36.5.14	CADCnSFTCR — Safety Control Register	3674
36.5.15	CADCnECR — Error Clear Register	3675
36.5.16	CADCnER — Error Register	3676
36.5.17	CADCnTDLVR — Pin Level Self-Diagnosis Level Register	3678
36.5.18	CADCnULTBR0/1/2/3 — Upper-Limit/Lower-Limit Table Register 0/1/2/3	3679
36.6	Operation	3680
36.6.1	A/D Conversion Flow	3680
36.6.2	A/D Conversion time	3681
36.6.3	Starting an A/D Conversion by External Triggers	3684
36.6.4	Terminating an A/D Conversion via an External Trigger	3685
36.6.5	Starting an A/D Conversion via a Timer Trigger	3685
36.6.6	Terminating an A/D Conversion via a Timer Trigger	3685

36.6.7	Switching Virtual Channels via Start and End Triggers	3686
36.6.8	A/D Conversion Monitoring Function Using Monitor Pins	3688
36.6.9	DMA Request Sources	3689
36.6.10	AD Error Interrupt Request and AD Parity Error Interrupt Request	3690
36.6.11	Boundary Flag Output Function	3692
36.6.12	Read Gate Function	3693
36.6.13	Wiring-break Detection Function	3694
36.6.14	Self-Diagnosis Function	3695
36.6.14.1	Self-Diagnosis of Wiring-break Detection Function	3695
36.6.14.2	A/D Converter Self-Diagnosis Function	3696
36.6.14.3	Pin Level Self-Diagnosis Function	3696
36.6.15	DFE Control	3698
36.7	Usage Notes	3699
36.7.1	Notes on Using Analog Input Pins	3699
36.7.2	Module Standby Function	3699
Section 37	Digital Filter Engine (DFE)	3700
37.1	Features	3700
37.1.1	Units and Channels	3700
37.1.2	Register Base Address	3701
37.1.3	Clock Supply	3701
37.1.4	Interrupt Requests	3702
37.1.5	Reset Sources	3704
37.1.6	External Input/Output Signals	3704
37.2	Overview	3705
37.2.1	Features	3705
37.2.2	Overall Configuration	3706
37.2.3	Overall Configuration of the FIFO Macro	3707
37.2.4	Data Format	3708
37.2.5	Filtering Circuit	3709
37.2.5.1	FIR Filter	3709
37.2.5.2	IIR Filter	3709
37.2.6	Dispatch Circuit	3710
37.2.7	Accumulation Circuit	3711
37.2.8	Subtraction Circuit	3711
37.2.9	Peak-Hold (PH) Circuit	3711
37.2.10	Output Circuit	3713
37.2.11	Intermediate Values Output Circuit	3713
37.2.12	Capture Circuit	3713
37.2.13	Buffer A Circuit	3713
37.2.14	Buffer B Circuit	3713
37.2.15	Terms	3714
37.3	Control Registers	3717
37.3.1	List of Registers for DFE Macro	3717

37.3.2	List of Registers for the FIFO Macro.....	3725
37.3.3	DFEjCTLACHn — Control Register A n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1).....	3726
37.3.4	DFEjCTLBCHn — Control Register B n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1).....	3729
37.3.5	DFEjDOCHn — Output Data Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1).....	3734
37.3.6	DFEjPHCHn — Peak-Hold Result Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1).....	3735
37.3.7	DFEjPHINDCHn — Peak-Hold Index Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1).....	3736
37.3.8	DFEjSTCHn — Status Register n (n = 0 to 11, for DFE0, n = 0 to 3 for DFE1).....	3738
37.3.9	DFEjCLRSTCHn — Clear Status Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1).....	3742
37.3.10	DFEjERMCHn — Error Mask Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1)....	3744
37.3.11	DFEjTRGCHn — Trigger Setting Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1).....	3745
37.3.12	DFEjTMRGCHn — Timer Trigger Setting Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1).....	3748
37.3.13	DFEjTRHCHn — Trigger History Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1).....	3751
37.3.14	DFEjCPA to DFEjCPD — Comparison Value Setting Register.....	3753
37.3.15	DFEjPHIA to DFEjPHID — Peak-Hold Initial Value Setting Register.....	3754
37.3.16	DFEjACA to DFEjACD — Accumulation/Decimation Count Setting Register.....	3755
37.3.17	DFEjDI — Software Input Data Register.....	3756
37.3.18	DFEjTRG — Software Trigger Register.....	3757
37.3.19	DFEjST — DFE Status Register.....	3758
37.3.20	DFEjEST — DFE Error Status Register.....	3759
37.3.21	DFEjPMITRG0 — Peak-Hold Mask-Start/End Timer Trigger Select Register 0.....	3760
37.3.22	DFEjPMITRG1 — Peak-Hold Mask-Start/End Timer Trigger Select Register 1.....	3762
37.3.23	DFEjPITRG — Peak-Hold Initialization/End Timer Trigger Select Register.....	3763
37.3.24	DFEjPITRG1 — Peak-Hold Initialization/End Timer Trigger Select Register 1.....	3765
37.3.25	DFEjMITRG — Accumulation/Decimation Initialization/Prohibition Timer Trigger Select Register.....	3766
37.3.26	DFEjMITRG1 — Accumulation/Decimation Initialization/Prohibition Timer Trigger Select Register 1.....	3768
37.3.27	DFEjFITRG — Filter Initialization Timer Trigger Select Register.....	3769
37.3.28	DFEjFITRG1 — Filter Initialization Timer Trigger Select Register 1.....	3771
37.3.29	DFEjPHUPDCn — Peak-Hold Update Notification Setting Register n (n = 0 to 3).....	3772
37.3.30	DFEjHOLCHn — Intermediate Value Output Register L n (n = even channels).....	3773
37.3.31	DFEjHOHCHn — Intermediate Value Output Register H n (n = even channels).....	3774
37.3.32	DFEjHOLMCHn — Intermediate Value Output Mirror Register L n (n = even channels).....	3775
37.3.33	DFEjGAINCHn — IIR Filter Gain Setting Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1).....	3776
37.3.34	DFEjCPOFSTn — Comparison Offset Value Setting Register n (n = 0 to 2).....	3778

37.3.35	DFEjPH23CCTL0 — Peak-Hold 23 Common Control Register 0	3779
37.3.36	DFEjPH2CTL0 — Peak-Hold 2 Control Register 0.....	3780
37.3.37	DFEjPH3CTL0 — Peak-Hold 3 Control Register 0.....	3782
37.3.38	DFEjPH20 — Peak-Hold 2 Result Register 0.....	3784
37.3.39	DFEjPH30 — Peak-Hold 3 Result Register 0.....	3785
37.3.40	DFEjPH2IND0 — Peak-Hold 2 Index Register 0	3786
37.3.41	DFEjPH3IND0 — Peak-Hold 3 Index Register 0	3788
37.3.42	DFEjSUBCTLCHn — Subtraction Control Register n (n = 0 to 2).....	3790
37.3.43	DFEjSUBDOCHn — Subtraction Result Register n (n = 0 to 2).....	3792
37.3.44	DFEjSUBSTCHn — Subtraction Status Register n (n = 0 to 2).....	3793
37.3.45	DFEjSUBCLRSTCHn — Subtraction Clear Status Register n (n = 0 to 2).....	3795
37.3.46	DFEjSUBERMCHn — Subtraction Error Mask Register n (n = 0 to 2).....	3796
37.3.47	DFEjSUBTRGCHn — Subtraction Trigger Setting Register n (n = 0 to 2)	3797
37.3.48	DFEjSUBTRHCHn — Subtraction Trigger History Register n (n = 0 to 2)	3799
37.3.49	DFEjSUBTRG0 — Subtraction Start/End Timer Trigger Select Register 0.....	3800
37.3.50	DFEjCAPCTLCHn — Capture Control Register n (n = 0 to 2)	3801
37.3.51	DFEjCAPDOCHn — Capture Result Register n (n = 0 to 2)	3802
37.3.52	DFEjCAPSTCHn — Capture Status Register n (n = 0 to 2).....	3803
37.3.53	DFEjCAPCLRSTCHn — Capture Clear Status Register n (n = 0 to 2)	3804
37.3.54	DFEjCAPERMCHn — Capture Error Mask Register n (n = 0 to 2)	3805
37.3.55	DFBFACCTL — Buffer A Common Control Register	3806
37.3.56	DFBFACCTLCHn — Buffer A Control Register n (n = 0 to 5).....	3807
37.3.57	DFBFADOCHn — Buffer A Output Register n (n = 0 to 5).....	3808
37.3.58	DFBFACLRLR — Buffer A Clear Register	3811
37.3.59	DFBFACST — Buffer A Common Status Register	3812
37.3.60	DFBFBCTL — Buffer B Control Register.....	3813
37.3.61	DFBFBDO — Buffer B Output Register	3814
37.3.62	DFBFBCLR — Buffer B Clear Register	3815
37.3.63	DFBFBST — Buffer B Status Register	3816
37.4	Operation	3817
37.4.1	Overview of Operation	3817
37.4.1.1	DFE Macro.....	3817
37.4.1.2	FIFO Macro.....	3819
37.4.2	Operating Procedures.....	3821
37.4.2.1	Channel Start Procedure	3821
37.4.2.2	Channel Stop Procedure	3824
37.4.2.3	Error Processing Procedure	3826
37.5	Details.....	3831
37.5.1	Data Flow	3831
37.5.1.1	DFE Macro.....	3831
37.5.1.2	FIFO Macro.....	3832
37.5.2	Address Map	3833
37.5.2.1	DFE Macro.....	3833
37.5.2.2	FIFO Macro.....	3833

37.5.3	Memory	3834
37.5.4	Format of Coefficient Memory.....	3835
37.5.5	Format of Data Memory	3836
37.5.6	Setting Control Registers	3837
37.5.6.1	FIR (Fixed-Point).....	3837
37.5.6.2	FIR (Integer).....	3838
37.5.6.3	IIR (Fixed-Point).....	3839
37.5.7	Trigger Control	3840
37.5.7.1	Timer Trigger Input	3840
37.5.7.2	Trigger Flag Functions	3844
37.5.7.3	Trigger Flag Functions and Processing	3848
37.5.7.4	Automatic Initialization of the Channel.....	3849
37.5.8	Input Data Control.....	3851
37.5.8.1	AD Input	3851
37.5.8.2	Cascade Input.....	3851
37.5.8.3	Software Input.....	3853
37.5.8.4	Dispatch	3853
37.5.9	Filtering Circuit	3855
37.5.9.1	FIR (8, 16, 24, 32 Taps).....	3855
37.5.9.2	FIR (64 Taps).....	3857
37.5.9.3	One-Stage IIR	3858
37.5.9.4	Two-Stage IIR	3860
37.5.9.5	Three-Stage IIR	3862
37.5.9.6	Errors in the Filtering Circuit	3864
37.5.10	Accumulation Circuit	3864
37.5.10.1	Absolute Value Calculation	3864
37.5.10.2	Summary of Accumulation Processing	3864
37.5.10.3	Operation of Accumulation Processing.....	3865
37.5.10.4	Operation of Accumulation Processing (with Trigger Flag)	3867
37.5.10.5	Summary of Decimation Processing.....	3869
37.5.10.6	Operation for Decimation Processing	3869
37.5.10.7	Operation of Decimation Processing (with Trigger Flag).....	3871
37.5.11	Subtraction Circuit.....	3873
37.5.11.1	Summary of the Subtraction Circuit	3874
37.5.11.2	Operation of the Subtraction Circuit.....	3875
37.5.11.3	Operation of Subtraction Circuit (with Subtraction Start Trigger)	3880
37.5.11.4	Operation of Subtraction Circuit (with Subtraction End Trigger)	3881
37.5.11.5	Operation of Subtraction Circuit (Stop by DFEjSUBCTLCHn.SEN).....	3882
37.5.12	Peak-Hold Circuit	3884
37.5.12.1	Summary of Peak-Hold Processing	3884
37.5.12.2	Operation of Peak-Hold Processing	3885
37.5.12.3	Operation of Peak-Hold Processing (Trigger Flag is Enabled).....	3887
37.5.12.4	Operation of Peak-Hold Mask Processing (Trigger Flag is Enable).....	3889
37.5.12.5	Operation of Comparison Processing	3891
37.5.12.6	Operation of the Peak-Hold Index Register	3892
37.5.12.7	Reading of the Peak-Hold Index Register and Operation	3894
37.5.12.8	Operation When Both Peak-Hold Processing and Comparison Processing Are Executed	3896
37.5.12.9	Summary of Peak-Hold-23 Processing.....	3898
37.5.12.10	Operation of Peak-Hold-23 Function	3899
37.5.12.11	Operation of the Peak-Hold Index Register When Peak-Hold-23 is Enabled	3906

37.5.12.12	Status of Peak-Hold-23 Function	3906
37.5.12.13	Operation of Peak-Hold-23 Function (Trigger Flag is enable)	3907
37.5.12.14	Setting Example of Peak-Hold-23 Function.....	3908
37.5.13	Output Circuit	3910
37.5.13.1	Floating-Point Conversion Circuit	3910
37.5.13.2	Output Register Circuit.....	3910
37.5.13.3	Floating Point Conversion Circuits for the Intermediate Value Output Registers L and H	3910
37.5.13.4	Intermediate Value Output Register Circuits	3910
37.5.14	Capture Circuit	3911
37.5.14.1	Summary of Capture Circuit	3911
37.5.15	Buffer A Circuit.....	3912
37.5.15.1	Summary of Buffer A Circuit	3912
37.5.15.2	Operation of Output Data Register Enable Flag and Buffer A Data Update Flag	3913
37.5.15.3	Initialization of the Buffer A Circuit.....	3914
37.5.15.4	Overwriting of FIFO When There is an Overflow.....	3914
37.5.16	Buffer B Circuit	3915
37.5.16.1	Summary of Buffer B Circuit	3915
37.5.16.2	Operation of Output Data Register Enable Flag.....	3916
37.5.16.3	Initialization for Buffer B Circuit.....	3916
37.5.17	Status Register.....	3917
37.5.17.1	Conditions for Setting and Clearing the DOEN Flag	3918
37.5.18	Error Mask Registers	3920
37.5.19	Subtraction Status Register	3920
37.5.20	Subtraction Error Mask Registers	3921
37.5.21	Capture Status Register.....	3921
37.5.22	Capture Error Mask Register	3921
37.5.23	Buffer A Status Register	3922
37.5.24	Buffer A Clear Register	3922
37.5.25	Buffer B Status Register	3922
37.5.26	Buffer B Clear Register	3922
37.5.27	Interrupt Requests.....	3923
37.5.27.1	DFE Macro.....	3923
37.5.27.2	FIFO Macro.....	3930
37.6	Notes and Restrictions.....	3932
37.6.1	Processing Time	3932
37.6.1.1	Processing Time for One Channel.....	3932
37.6.1.2	Termination Processing Time	3932
37.6.2	DFE Activation Input Interval	3932
37.6.2.1	DFE Processing Time	3933
37.6.3	Trigger Input.....	3934
37.6.4	Channel Tag and AD Tag	3935
37.6.5	Input Data Format and Calculation Restrictions	3935
37.6.6	Restrictions on Cascade Processing	3935
37.6.7	Operation when the Given Channel is Disabled	3936
37.6.7.1	Channel Enable Bits and Invalidation	3936

37.6.8	Restrictions on FIR 64TAP	3937
37.6.9	Restrictions on Memory Access.....	3937
37.6.10	Restrictions on Trigger Setting Registers	3937
37.6.11	Setting of Peak-Hold Processing Disable Bit and Accumulation/Decimation Processing Disable Bit	3937
37.6.12	Controlling Control Registers A and B	3937
37.6.13	Controlling the Peak-Hold Index Registers	3938
37.6.14	Restrictions on the Intermediate Value Output Registers.....	3938
37.6.15	Restrictions on Subtraction Circuit.....	3938
37.6.16	Restrictions on Buffer A Circuit	3939
37.6.16.1	Restrictions on FIFO Channels In Use	3939
37.6.16.2	Notes about Read Registers.....	3939
37.6.16.3	Notes about Buffer A Capture Interrupt Request.....	3939
37.6.17	Restrictions on Buffer B Circuit	3940
37.6.17.1	Notes Concerning the Buffer B Capture Interrupt Request	3940
37.6.17.2	Restrictions on Buffer B Circuit when both Buffer A and Buffer B are in Use	3940
37.6.18	Note on Using Automatic Channel Initialization	3940
37.6.19	Restrictions on Using Peak-Hold Mask Function.....	3940
37.6.20	Restrictions on Using Peak-Hold Function	3941

Section 38 Functional Safety 3942

38.1	Overview	3942
38.2	Reset Sources	3943
38.3	ECC and EDC.....	3944
38.3.1	Overview	3944
38.3.1.1	ECC.....	3944
38.3.1.2	Address Parity.....	3945
38.3.1.3	Address Feedback.....	3945
38.3.2	Key Code Protection for ECC Control Registers	3946
38.3.2.1	Overview	3946
38.3.2.2	List of Registers	3946
38.3.2.3	KCPROT — ECC Control Key Code Protection Register	3946
38.3.3	Code Flash ECC and Address Parity.....	3947
38.3.3.1	Overview	3947
38.3.3.2	Register Base Address	3948
38.3.3.3	List of Registers	3949
38.3.3.4	CFPECCCTL — ECC Control Register	3950
38.3.3.5	CFPECCTSTCTL — ECC Test Control Register	3951
38.3.3.6	CFCAPCTL — Address Parity Control Register.....	3952
38.3.3.7	CFSECCCTL — ECC Control Register	3953
38.3.3.8	CFSECCTSTCTL — ECC Test Control Register	3954
38.3.3.9	CF_ERRINT — Error Notification Control Register	3955
38.3.3.10	CF_SSTCLR — 1-bit Error Status Clear Register.....	3956
38.3.3.11	CF_DSTCLR — Fatal Error Status Clear Register.....	3957
38.3.3.12	CF_OVFCLR — 1-bit Error Overflow Status Clear Register	3958
38.3.3.13	CF_SERSTR — 1-bit Error Status Register	3959
38.3.3.14	CF_DERSTR — Fatal Error Status Register.....	3960
38.3.3.15	CF_OVFSTR — 1-bit Error Overflow Status Register.....	3961

38.3.3.16	CF_SERINF — 1-bit Error Location Information Register	3962
38.3.3.17	CF_nSEADR — n-th 1-bit Error Address Register (n = 00 to 03)	3963
38.3.3.18	CF_00DEADR — 1st Fatal Error Address Register	3964
38.3.3.19	Test Function	3965
38.3.4	Data Flash ECC	3966
38.3.4.1	Overview	3966
38.3.4.2	List of Registers	3966
38.3.4.3	DFECCCTL — Data Flash ECC Control Register	3967
38.3.4.4	DFERSTR — Data Flash Error Status Register	3968
38.3.4.5	DFERSTC — Data Flash Error Status Clear Register	3969
38.3.4.6	DFOVFSTR — Data Flash Error Overflow Status Register	3970
38.3.4.7	DFOVFSTC — Data Flash Error Overflow Status Clear Register.....	3971
38.3.4.8	DFERRINT — Data Flash Error Notification Control Register.....	3972
38.3.4.9	DFEADR — Data Flash 1st Error Address Register	3973
38.3.4.10	DFTSTCTL — Data Flash Test Control Register	3974
38.3.4.11	DFKCPROT — Data Flash ECC Key Code Protection Register.....	3975
38.3.4.12	Test Function	3976
38.3.5	Local RAM ECC and Address Feedback.....	3977
38.3.5.1	Overview	3977
38.3.5.2	Register Base Address	3979
38.3.5.3	List of Registers	3979
38.3.5.4	LRECCCTL — ECC Control Register.....	3981
38.3.5.5	LRECCTSTCTL — ECC Test Control Register.....	3982
38.3.5.6	LRAFINV — Address Feedback Test Control Register.....	3983
38.3.5.7	LRTDATBFECF — ECC Read Buffer Register	3984
38.3.5.8	LR0_ERRINT — Error Notification Control Register	3985
38.3.5.9	LR0_SSTCLR — 1-bit Error Status Clear Register	3986
38.3.5.10	LR0_DSTCLR — Fatal Error Status Clear Register	3987
38.3.5.11	LR0_OVFCLR — 1-bit Error Overflow Status Clear Register	3988
38.3.5.12	LR0_SERSTR — 1-bit Error Status Register	3989
38.3.5.13	LR0_DERSTR — Fatal Error Status Register	3990
38.3.5.14	LR0_OVFSTR — 1-bit Error Overflow Status Register	3991
38.3.5.15	LR0_SERINF — 1-bit Error Location Information Register	3992
38.3.5.16	LR0_nSEADR — n-th 1-bit Error Address Register (n = 00 to 07).....	3993
38.3.5.17	LR0_00DEADR — 1st Fatal Error Address Register	3994
38.3.5.18	LR1_ERRINT — Error Notification Control Register	3995
38.3.5.19	LR1_SSTCLR — 1-bit Error Status Clear Register	3996
38.3.5.20	LR1_DSTCLR — Fatal Error Status Clear Register	3997
38.3.5.21	LR1_OVFCLR — 1-bit Error Overflow Status Clear Register	3998
38.3.5.22	LR1_SERSTR — 1-bit Error Status Register	3999
38.3.5.23	LR1_DERSTR — Fatal Error Status Register	4000
38.3.5.24	LR1_OVFSTR — 1-bit Error Overflow Status Register	4001
38.3.5.25	LR1_SERINF — 1-bit Error Location Information Register	4002
38.3.5.26	LR1_nSEADR — n-th 1-bit Error Address Register (n = 00 to 07).....	4003
38.3.5.27	LR1_00DEADR — 1st Fatal Error Address Register	4004
38.3.5.28	Test Function	4005
38.3.6	Cluster RAM ECC and Address Feedback.....	4007
38.3.6.1	Overview	4007
38.3.6.2	Register Base Address	4008
38.3.6.3	List of Registers	4009
38.3.6.4	CRCAFCTL — Address Feedback Control Register	4010
38.3.6.5	CRCECCTSTCTL — ECC Test Control Register	4011

38.3.6.6	CRCAFINV — Address Feedback Test Control Register.....	4012
38.3.6.7	CRCTDATABFECCF — ECC Read Buffer Register	4013
38.3.6.8	CRAECCCTL — ECC Control Register.....	4014
38.3.6.9	CR_ERRINT — Error Notification Control Register.....	4015
38.3.6.10	CR_SSTCLR — 1-bit Error Status Clear Register	4016
38.3.6.11	CR_DSTCLR — Fatal Error Status Clear Register	4017
38.3.6.12	CR_OVFCLR — 1-bit Error Overflow Status Clear Register.....	4018
38.3.6.13	CR_SERSTR — 1-bit Error Status Register.....	4019
38.3.6.14	CR_DERSTR — Fatal Error Status Register	4020
38.3.6.15	CR_OVFSTR — 1-bit Error Overflow Status Register	4021
38.3.6.16	CR_SERINF — 1-bit Error Location Information Register.....	4022
38.3.6.17	CR_nSEADR — n-th 1-bit Error Address Register (n = 00 to 07).....	4023
38.3.6.18	CR_00DEADR — 1st Fatal Error Address Register	4025
38.3.6.19	Test Function	4027
38.3.7	Instruction Cache EDC and Address Feedback	4029
38.3.7.1	Overview	4029
38.3.7.2	Register Base Address	4030
38.3.7.3	List of Registers	4031
38.3.7.4	ITECCCTL — ECC Control Register.....	4032
38.3.7.5	ITAFINV — Address Feedback Test Control Register	4033
38.3.7.6	IDeccCTL — ECC Control Register.....	4034
38.3.7.7	IDAFINV — Address Feedback Test Control Register.....	4035
38.3.7.8	IT_ERRINT — Error Notification Control Register.....	4036
38.3.7.9	IT_SSTCLR — 1-bit Error Status Clear Register	4037
38.3.7.10	IT_DSTCLR — Fatal Error Status Clear Register	4038
38.3.7.11	IT_OVFCLR — 1-bit Error Overflow Status Clear Register.....	4039
38.3.7.12	IT_SERSTR — 1-bit Error Status Register.....	4040
38.3.7.13	IT_DERSTR — Fatal Error Status Register.....	4041
38.3.7.14	IT_OVFSTR — 1-bit Error Overflow Status Register	4042
38.3.7.15	IT_SERINF — 1-bit Error Location Information Register.....	4043
38.3.7.16	IT_00SEADR — 1st 1-bit Error Address Register	4044
38.3.7.17	IT_00DEADR — 1st Fatal Error Address Register	4045
38.3.7.18	ID_ERRINT — Error Notification Control Register	4046
38.3.7.19	ID_SSTCLR — 1-bit Error Status Clear Register	4047
38.3.7.20	ID_DSTCLR — Fatal Error Status Clear Register.....	4048
38.3.7.21	ID_OVFCLR — 1-bit Error Overflow Status Clear Register	4049
38.3.7.22	ID_SERSTR — 1-bit Error Status Register	4050
38.3.7.23	ID_DERSTR — Fatal Error Status Register	4051
38.3.7.24	ID_OVFSTR — 1-bit Error Overflow Status Register.....	4052
38.3.7.25	ID_SERINF — 1-bit Error Location Information Register	4053
38.3.7.26	ID_00SEADR — 1st 1-bit Error Address Register.....	4054
38.3.7.27	ID_00DEADR — 1st Fatal Error Address Register	4055
38.3.7.28	Test Function	4056
38.3.8	sDMAC/DTS RAM ECC and Address Feedback.....	4057
38.3.8.1	Overview	4057
38.3.8.2	Register Base Address	4059
38.3.8.3	List of Registers	4059
38.3.8.4	DRECCCTL — ECC Control Register.....	4061
38.3.8.5	DRECCTSTCTL — ECC Test Control Register.....	4062
38.3.8.6	DRAFINV — Address Feedback Test Control Register	4063
38.3.8.7	DRECCTSTDIN0 — Test Data Input Register	4064
38.3.8.8	DRECCTSTEIN — Test ECC Input Register.....	4065

38.3.8.9	DRTDATBFDATAF0 — Data Read Buffer Register	4066
38.3.8.10	DRTDATBFECCECF — ECC Read Buffer Register	4067
38.3.8.11	DEECCCTL — ECC Control Register	4068
38.3.8.12	DEECCSTCTL — ECC Test Control Register	4069
38.3.8.13	DEECCSTDIN0 — Test Data Input Register	4070
38.3.8.14	DEECCSTEIN — Test ECC Input Register	4071
38.3.8.15	DETDATBFDATAF0 — Data Read Buffer Register	4072
38.3.8.16	DETDATBFECCECF — ECC Read Buffer Register	4073
38.3.8.17	DAECCCTL — ECC Control Register	4074
38.3.8.18	DAECCSTCTL — ECC Test Control Register	4075
38.3.8.19	DAECCSTDIN0 — Test Lower Data Input Register	4076
38.3.8.20	DAECCSTDIN1 — Test Upper Data Input Register	4077
38.3.8.21	DAECCSTEIN — Test ECC Input Register	4078
38.3.8.22	DATDATBFDATAF0 — Lower Data Read Buffer Register	4079
38.3.8.23	DATDATBFDATAF1 — Upper Data Read Buffer Register	4080
38.3.8.24	DATDATBFECCECF — ECC Read Buffer Register	4081
38.3.8.25	DR_ERRINT — Error Notification Control Register	4082
38.3.8.26	DR_SSTCLR — 1-bit Error Status Clear Register	4083
38.3.8.27	DR_DSTCLR — Fatal Error Status Clear Register	4084
38.3.8.28	DR_OVFCLR — 1-bit Error Overflow Status Clear Register	4085
38.3.8.29	DR_SERSTR — 1-bit Error Status Register	4086
38.3.8.30	DR_DERSTR — Fatal Error Status Register	4087
38.3.8.31	DR_OVFSTR — 1-bit Error Overflow Status Register	4088
38.3.8.32	DR_SERINF — 1-bit Error Location Information Register	4089
38.3.8.33	DR_00SEADR — 1st 1-bit Error Address Register	4090
38.3.8.34	DR_00DEADR — 1st Fatal Error Address Register	4091
38.3.8.35	DE_ERRINT — Error Notification Control Register	4092
38.3.8.36	DE_SSTCLR — 1-bit Error Status Clear Register	4093
38.3.8.37	DE_DSTCLR — Fatal Error Status Clear Register	4094
38.3.8.38	DE_OVFCLR — 1-bit Error Overflow Status Clear Register	4095
38.3.8.39	DE_SERSTR — 1-bit Error Status Register	4096
38.3.8.40	DE_DERSTR — Fatal Error Status Register	4097
38.3.8.41	DE_OVFSTR — 1-bit Error Overflow Status Register	4098
38.3.8.42	DE_SERINF — 1-bit Error Location Information Register	4099
38.3.8.43	DE_00SEADR — 1st 1-bit Error Address Register	4100
38.3.8.44	DE_00DEADR — 1st Fatal Error Address Register	4101
38.3.8.45	DA_ERRINT — Error Notification Control Register	4102
38.3.8.46	DA_SSTCLR — 1-bit Error Status Clear Register	4103
38.3.8.47	DA_DSTCLR — Fatal Error Status Clear Register	4104
38.3.8.48	DA_OVFCLR — 1-bit Error Overflow Status Clear Register	4105
38.3.8.49	DA_SERSTR — 1-bit Error Status Register	4106
38.3.8.50	DA_DERSTR — Fatal Error Status Register	4107
38.3.8.51	DA_OVFSTR — 1-bit Error Overflow Status Register	4108
38.3.8.52	DA_SERINF — 1-bit Error Location Information Register	4109
38.3.8.53	DA_00SEADR — 1st 1-bit Error Address Register	4110
38.3.8.54	DA_00DEADR — 1st Fatal Error Address Register	4111
38.3.8.55	Test Function	4112
38.3.9	ECC for Peripheral RAM (32 Bits)	4113
38.3.9.1	Overview	4113
38.3.9.2	List of Registers	4115
38.3.9.3	E710CTL — ECC Control Register	4117
38.3.9.4	E710TMC — ECC Test Mode Control Register	4119
38.3.9.5	E710TED — ECC Encoder and Decoder Data Test Register	4121

38.3.9.6	E710TRC — ECC Redundant Bit Data Control Test Register	4122
38.3.9.7	ECSYND — ECC Decoder Syndrome Data Register	4123
38.3.9.8	ECHORD — ECC 7-Bit Redundant Data Holding Test Register	4124
38.3.9.9	ECECRD — ECC Encoder Test Register.....	4125
38.3.9.10	ECERDB — ECC Redundant Bit Input and Output Substitution Buffer Register.....	4126
38.3.9.11	E710EAD — ECC Error Address Register	4127
38.3.9.12	ETCTXECCCTL— Ethernet TX RAM ECC Control Register	4128
38.3.9.13	ETCRXECCCTL— Ethernet RX RAM ECC Control Register	4129
38.3.9.14	Notification to ECM	4130
38.3.9.15	Test Function	4130
38.3.10	ECC for Peripheral RAM (16 Bits)	4131
38.3.10.1	Overview	4131
38.3.10.2	List of Registers	4132
38.3.10.3	E610CTL — ECC Control Register	4133
38.3.10.4	E610TMC — ECC Test Mode Control Register	4135
38.3.10.5	E610TED — ECC Encoder and Decoder Data Test Register.....	4137
38.3.10.6	ECEDB — ECC Encoder and Decoder Data Input and Output Substitution Buffer Register	4138
38.3.10.7	ECRIDM — ECC Redundant Bit Input Data Monitoring Register	4138
38.3.10.8	ECERDB — ECC Redundant Bit Input and Output Substitution Buffer Register.....	4139
38.3.10.9	E610TRC — ECC Redundant Bit Data Control Test Register	4139
38.3.10.10	ECSYND — ECC Decoder Syndrome Data Register	4140
38.3.10.11	ECHORD — ECC 6-Bit Redundant Data Holding Test Register	4140
38.3.10.12	ECECRD — ECC Encoder Test Register.....	4141
38.3.10.13	ECRODM — ECC Redundant Bit Output Data Monitoring Register.....	4141
38.3.10.14	E610EAD — ECC Error Address Register	4142
38.3.10.15	Notification to ECM	4143
38.3.10.16	Test Function	4143
38.3.11	Safety Mechanism on Data Transfer Path.....	4144
38.3.11.1	Overview	4144
38.3.11.2	Register Base Address	4145
38.3.11.3	List of Registers	4148
38.3.11.4	BUSAECCTL — ECC Control Register	4156
38.3.11.5	BUSAECCTSTCTL — ECC Test Control Register	4157
38.3.11.6	BUSAECCTSTDIN0 — Test Data Input Register.....	4158
38.3.11.7	BUSAECCTSTEIN — Test ECC Input Register.....	4159
38.3.11.8	BUSDVCECCCTL — ECC Control Register	4160
38.3.11.9	BUSDVCECCTSTCTL — ECC Test Control Register	4161
38.3.11.10	BUSDVCECCTSTDIN0 — Test Data Input Register	4162
38.3.11.11	BUSDVCECCTSTEIN — Test ECC Input Register.....	4163
38.3.11.12	BUSDV1ECCCTL — ECC Control Register.....	4164
38.3.11.13	BUSDV1ECCCTSTCTL — ECC Test Control Register.....	4165
38.3.11.14	BUSDV1ECCCTSTDIN0 — Test Data Input Register	4166
38.3.11.15	BUSDV1ECCCTSTEIN — Test ECC Input Register	4167
38.3.11.16	BUSDEMECCCTL — ECC Control Register.....	4168
38.3.11.17	BUSDEMECCTSTCTL — ECC Test Control Register.....	4169
38.3.11.18	BUSDEMECCTSTDIN0 — Test Data Input Register	4170
38.3.11.19	BUSDEMECCTSTEIN — Test ECC Input Register	4171
38.3.11.20	BUSDDEECCCTL — ECC Control Register	4172
38.3.11.21	BUSDDEECCTSTCTL — ECC Test Control Register	4173
38.3.11.22	BUSDDEECCTSTDIN0 — Test Data Input Register	4174
38.3.11.23	BUSDDEECCTSTDIN1 — Test Data Input Register	4175

38.3.11.24	BUSDDEECCTSTEIN — Test ECC Input Register	4176
38.3.11.25	BUSDECCCTL — ECC Control Register	4177
38.3.11.26	LR_BUSERRINT — Error Notification Control Register	4178
38.3.11.27	LR_BUSASSTCLR — Address 1-bit Error Status Clear Register	4179
38.3.11.28	LR_BUSADSTCLR — Address 2-bit Error Status Clear Register	4180
38.3.11.29	LR_BUSDSSTCLR — Data 1-bit Error Status Clear Register	4181
38.3.11.30	LR_BUSDDSTCLR — Data 2-bit Error Status Clear Register	4182
38.3.11.31	LR_BUSASERSTR — Address 1-bit Error Status Register	4183
38.3.11.32	LR_BUSADERSTR — Address 2-bit Error Status Register	4184
38.3.11.33	LR_BUSDSERSTR — Data 1-bit Error Status Register	4185
38.3.11.34	LR_BUSDDERSTR — Data 2-bit Error Status Register	4186
38.3.11.35	CR_BUSERRINT — Error Notification Control Register	4187
38.3.11.36	CR_BUSASSTCLR — Address 1-bit Error Status Clear Register	4188
38.3.11.37	CR_BUSADSTCLR — Address 2-bit Error Status Clear Register	4189
38.3.11.38	CR_BUSDSSTCLR — Data 1-bit Error Status Clear Register	4190
38.3.11.39	CR_BUSDDSTCLR — Data 2-bit Error Status Clear Register	4191
38.3.11.40	CR_BUSASERSTR — Address 1-bit Error Status Register	4192
38.3.11.41	CR_BUSADERSTR — Address 2-bit Error Status Register	4193
38.3.11.42	CR_BUSDSERSTR — Data 1-bit Error Status Register	4194
38.3.11.43	CR_BUSDDERSTR — Data 2-bit Error Status Register	4195
38.3.11.44	CF_BUSERRINT — Error Notification Control Register	4196
38.3.11.45	CF_BUSASSTCLR — Address 1-bit Error Status Clear Register	4197
38.3.11.46	CF_BUSADSTCLR — Address 2-bit Error Status Clear Register	4198
38.3.11.47	CF_BUSASERSTR — Address 1-bit Error Status Register	4199
38.3.11.48	CF_BUSADERSTR — Address 2-bit Error Status Register	4200
38.3.11.49	PH_BUSERRINT — Error Notification Control Register	4201
38.3.11.50	PH_BUSASSTCLR — Address 1-bit Error Status Clear Register	4202
38.3.11.51	PH_BUSADSTCLR — Address 2-bit Error Status Clear Register	4203
38.3.11.52	PH_BUSDSSTCLR — Data 1-bit Error Status Clear Register	4204
38.3.11.53	PH_BUSDDSTCLR — Data 2-bit Error Status Clear Register	4205
38.3.11.54	PH_BUSASERSTR — Address 1-bit Error Status Register	4206
38.3.11.55	PH_BUSADERSTR — Address 2-bit Error Status Register	4207
38.3.11.56	PH_BUSDSERSTR — Data 1-bit Error Status Register	4208
38.3.11.57	PH_BUSDDERSTR — Data 2-bit Error Status Register	4209
38.3.11.58	DM_BUSERRINT — Error Notification Control Register	4210
38.3.11.59	DM_BUSDSSTCLR — Data 1-bit Error Status Clear Register	4211
38.3.11.60	DM_BUSDDSTCLR — Data 2-bit Error Status Clear Register	4212
38.3.11.61	DM_BUSDSERSTR — Data 1-bit Error Status Register	4213
38.3.11.62	DM_BUSDDERSTR — Data 2-bit Error Status Register	4214
38.3.11.63	EM_BUSERRINT — Error Notification Control Register	4215
38.3.11.64	EM_BUSASSTCLR — Address 1-bit Error Status Clear Register	4216
38.3.11.65	EM_BUSADSTCLR — Address 2-bit Error Status Clear Register	4217
38.3.11.66	EM_BUSDSSTCLR — Data 1-bit Error Status Clear Register	4218
38.3.11.67	EM_BUSDDSTCLR — Data 2-bit Error Status Clear Register	4219
38.3.11.68	EM_BUSASERSTR — Address 1-bit Error Status Register	4220
38.3.11.69	EM_BUSADERSTR — Address 2-bit Error Status Register	4221
38.3.11.70	EM_BUSDSERSTR — Data 1-bit Error Status Register	4222
38.3.11.71	EM_BUSDDERSTR — Data 2-bit Error Status Register	4223
38.3.11.72	APECECCCTL — ECC Control Register	4224
38.3.11.73	APECERRINT — Error Notification Control Register	4225
38.3.11.74	APECSTCLR — Error Status Clear Register	4226
38.3.11.75	APEC1STERSTR — Error Status Register	4227
38.3.11.76	APECTSTCTL — ECC Test Control Register	4228

38.3.11.77	APECTSTDIN0 — Test Data Input Register	4229
38.3.11.78	APECTSTEIN — Test ECC Input Register.....	4230
38.3.11.79	APECKCPROT — Key Code Protection Register.....	4231
38.3.11.80	PHC_BUSERRINT — Error Notification Control Register.....	4232
38.3.11.81	PHC_BUSASSTCLR — Address 1-bit Error Status Clear Register.....	4233
38.3.11.82	PHC_BUSADSTCLR — Address 2-bit Error Status Clear Register.....	4234
38.3.11.83	PHC_BUSDSSTCLR — Data 1-bit Error Status Clear Register	4235
38.3.11.84	PHC_BUSDDSTCLR — Data 2-bit Error Status Clear Register	4236
38.3.11.85	PHC_BUSASERSTR — Address 1-bit Error Status Register	4237
38.3.11.86	PHC_BUSADERSTR — Address 2-bit Error Status Register	4238
38.3.11.87	PHC_BUSDSERSTR — Data 1-bit Error Status Register.....	4239
38.3.11.88	PHC_BUSDDERSTR — Data 2-bit Error Status Register	4240
38.3.11.89	HBECECCCTL — ECC Control Register	4241
38.3.11.90	HBECERRINT — Error Notification Control Register.....	4242
38.3.11.91	HBECSTCLR — Error Status Clear Register	4243
38.3.11.92	HBEC1STERSTR — Error Status Register.....	4244
38.3.11.93	HBECTSTCTL — ECC Test Control Register	4245
38.3.11.94	HBECTSTDIN0 — Test Data Input Register	4246
38.3.11.95	HBECTSTEIN — Test ECC Input Register	4247
38.3.11.96	HBECKCPROT — Key Code Protection Register.....	4248
38.3.11.97	Test Function	4249
38.3.12	Error Injection for ECC Function	4250
38.3.12.1	Overview	4250
38.3.12.2	List of Registers	4250
38.3.12.3	EGDATORGn — Error Data Register.....	4251
38.3.12.4	EGECCCALn — 7-bit ECC Calculation Register	4252
38.3.12.5	EGDATINVn — Error Injection Register	4253
38.3.12.6	EGECCINVn — Error Injection Register.....	4254
38.3.12.7	EGECC8CAL — 8-bit ECC Calculation Register	4255
38.3.12.8	Usage of ERRGEN	4256
38.4	Hardware Redundancy	4258
38.4.1	Overview	4258
38.4.2	List of Registers	4258
38.4.3	DTS_COMP_CNTRL — Error Injection Control Register.....	4259
38.4.4	Usage Note	4260
38.5	Memory Protection.....	4261
38.5.1	Overview	4261
38.5.2	SPID	4263
38.5.2.1	Overview	4263
38.5.2.2	List of Registers	4266
38.5.2.3	BMnSPID — BusMaster n SPID Register	4268
38.5.2.4	BMnSPIDMSK — BusMaster n SPID Mask Register.....	4269
38.5.2.5	BMnSPIDMSKLOCK — BusMaster n SPID Mask Lock Register	4270
38.5.2.6	SPIDKCPROT — SPID Key Code Protect Register.....	4271
38.5.2.7	Usage.....	4272
38.5.3	PEG.....	4273
38.5.3.1	Overview	4273
38.5.3.2	Register Base Address	4273
38.5.3.3	List of Registers	4274
38.5.3.4	PEGKCPROT — Key Code Protection Register	4275

38.5.3.5	PEGPROTm — Channel Protection Control Register	4276
38.5.3.6	PEGSPIDm — Channel SPID Setting Register.....	4277
38.5.3.7	PEGBADm — Channel Base Address Setting Register.....	4278
38.5.3.8	PEGADVm — Channel Valid Bit Setting Register	4279
38.5.3.9	PEGERRCLR — Guard Error Clear Register.....	4280
38.5.3.10	PEGERRSTAT — Guard Error Status Register	4281
38.5.3.11	PEGERRADDR — Guard Error Address Register	4282
38.5.3.12	PEGERRTYPE — Guard Error Access Information Register	4283
38.5.4	CRG	4284
38.5.4.1	Overview	4284
38.5.4.2	Register Base Address	4284
38.5.4.3	List of Registers	4285
38.5.4.4	CRGKCPROT — Key Code Protection Register.....	4286
38.5.4.5	CRGPROTm — Channel Protection Control Register	4287
38.5.4.6	CRGSPIDm — Channel SPID Setting Register	4288
38.5.4.7	CRGBADm — Channel Base Address Setting Register	4289
38.5.4.8	CRGADVm — Channel Valid Bit Setting Register	4290
38.5.4.9	CRGERRCLR — Guard Error Clear Register	4291
38.5.4.10	CRGERRSTAT — Guard Error Status Register.....	4292
38.5.4.11	CRGERRADDR — Guard Error Address Register.....	4293
38.5.4.12	CRGERRTYPE — Guard Error Access Information Register	4294
38.5.5	INTC2 Guard.....	4295
38.5.5.1	Overview	4295
38.5.5.2	List of Registers	4295
38.5.5.3	INTC2GERRCLR — Guard Error Clear Register	4296
38.5.5.4	INTC2GERRSTAT — Guard Error Status Register	4297
38.5.5.5	INTC2GERRADDR — Guard Error Address Register	4298
38.5.5.6	INTC2GERRTYPE — Guard Error Access Information Register.....	4299
38.5.5.7	INTC2GKCPROT — Key Code Protection Register	4300
38.5.5.8	INTC2GMPIDn — SPID Setting Register	4301
38.5.5.9	INTC2GPROT_m/GR — Channel/INTC2 Protection Control Register	4302
38.5.6	DTS Guard	4303
38.5.6.1	Overview	4303
38.5.6.2	List of Registers	4303
38.5.6.3	DTSGERRCLR — Guard Error Clear Register	4304
38.5.6.4	DTSGERRSTAT — Guard Error Status Register.....	4305
38.5.6.5	DTSGERRADDR — Guard Error Address Register.....	4306
38.5.6.6	DTSGERRTYPE — Guard Error Access Information Register	4307
38.5.6.7	DTSGKCPROT — Key Code Protection Register.....	4308
38.5.6.8	DTSGMPIDn — SPID Setting Register	4309
38.5.6.9	DTSGPROT_m/GR — Channel/DTS Protection Control Register.....	4310
38.5.7	sDMAC Guard.....	4311
38.5.7.1	Overview	4311
38.5.7.2	Register Base Address	4311
38.5.7.3	List of Registers	4311
38.5.7.4	DMAGERRCLR — Guard Error Clear Register	4312
38.5.7.5	DMAGERRSTAT — Guard Error Status Register	4313
38.5.7.6	DMAGERRADDR — Guard Error Address Register	4314
38.5.7.7	DMAGERRTYPE — Guard Error Access Information Register	4315
38.5.7.8	DMAGKCPROT — Key Code Protection Register	4316
38.5.7.9	DMAGMPIDn — SPID Setting Register	4317

38.5.7.10	DMAGPROT_m/GR/DP — Channel/DMA/DMA Descriptor Protection Control Register	4318
38.5.8	IBG	4319
38.5.8.1	Overview	4319
38.5.8.2	Register Base Address	4319
38.5.8.3	List of Registers	4319
38.5.8.4	IPIGERRCLR — Guard Error Clear Register	4321
38.5.8.5	IPIGERRSTAT — Guard Error Status Register	4322
38.5.8.6	IPIGERRADDR — Guard Error Address Register	4323
38.5.8.7	IPIGERRTYPE — Guard Error Access Information Register	4324
38.5.8.8	IPIGKCPROT — Key Code Protection Register	4325
38.5.8.9	IPIGPROT0_n/4 — Channel/Common Protection Control Register (n = 0 to 3) ..	4326
38.5.8.10	IPIGPROT1_n/4 — Channel/Common SPID Setting Register (n = 0 to 3)	4327
38.5.8.11	MEGERRCLR — Guard Error Clear Register	4328
38.5.8.12	MEGERRSTAT — Guard Error Status Register	4329
38.5.8.13	MEGERRADDR — Guard Error Address Register	4330
38.5.8.14	MEGERRTYPE — Guard Error Access Information Register	4331
38.5.8.15	MEGKCPROT — Key Code Protection Register	4332
38.5.8.16	MEGPROT0_n — Channel Protection Control Register	4333
38.5.8.17	MEGPROT1_n — Channel SPID Setting Register	4334
38.5.8.18	BRGERRCLR — Guard Error Clear Register	4335
38.5.8.19	BRGERRSTAT — Guard Error Status Register	4336
38.5.8.20	BRGERRADDR — Guard Error Address Register	4337
38.5.8.21	BRGERRTYPE — Guard Error Access Information Register	4338
38.5.8.22	BRGKCPROT — Key Code Protection Register	4339
38.5.8.23	BRGPROT0_n/16 — Channel/Common Protection Control Register (n = 0 to 15)	4340
38.5.8.24	BRGPROT1_n/16 — Channel/Common SPID Setting Register (n = 0 to 15)	4341
38.5.8.25	TPTGERRCLR — Guard Error Clear Register	4342
38.5.8.26	TPTGERRSTAT — Guard Error Status Register	4343
38.5.8.27	TPTGERRADDR — Guard Error Address Register	4344
38.5.8.28	TPTGERRTYPE — Guard Error Access Information Register	4345
38.5.8.29	TPTGKCPROT — Key Code Protection Register	4346
38.5.8.30	TPTGPROT0_n — Channel Protection Control Register	4347
38.5.8.31	TPTGPROT1_n — Channel SPID Setting Register	4348
38.5.9	PBG	4349
38.5.9.1	Overview	4349
38.5.9.2	Register Base Address	4354
38.5.9.3	List of Registers	4354
38.5.9.4	PBGERRCLR — Guard Error Clear Register	4355
38.5.9.5	PBGERRSTAT — Guard Error Status Register	4356
38.5.9.6	PBGERRADDR — Guard Error Address Register	4357
38.5.9.7	PBGERRTYPE — Guard Error Access Information Register	4358
38.5.9.8	PBGKCPROT — Key Code Protection Register	4359
38.5.9.9	PBGPROT0_m — Channel Protection Control Register	4360
38.5.9.10	PBGPROT1_m — Channel SPID Setting Register	4361
38.5.10	HBG	4362
38.5.10.1	Overview	4362
38.5.10.2	Register Base Address	4362
38.5.10.3	List of Registers	4362
38.5.10.4	HBGERRCLR — Guard Error Clear Register	4363
38.5.10.5	HBGERRSTAT — Guard Error Status Register	4364

38.5.10.6	HBGERRADDR — Guard Error Address Register	4365
38.5.10.7	HBGERRTYPE — Guard Error Access Information Register	4366
38.5.10.8	HBGKCPROT — Key Code Protection Register	4367
38.5.10.9	HBGPROT0 — Channel Protection Control Register	4368
38.5.10.10	HBGPROT1 — Channel SPID Setting Register	4369
38.6	BIST	4370
38.6.1	Overview	4370
38.6.2	Registers	4370
38.6.2.1	Register Protection	4370
38.6.2.2	List of Registers	4371
38.6.2.3	Reset of Registers	4372
38.6.2.4	LBISTREF1 — Logic BIST Reference Value Register 1	4373
38.6.2.5	LBISTREF2 — Logic BIST Reference Value Register 2	4374
38.6.2.6	MBISTREF1 — Memory BIST Reference Value Register 1	4375
38.6.2.7	LBISTSIG1 — Logic BIST Signature Value Register 1	4376
38.6.2.8	LBISTSIG2 — Logic BIST Signature Value Register 2	4377
38.6.2.9	MBISTSIG1 — Memory BIST Signature Value Register 1	4378
38.6.2.10	MBIST1FTAG0 — Memory BIST1 FTAG Signature Value Register 0	4379
38.6.2.11	MBIST1FTAG1 — Memory BIST1 FTAG Signature Value Register 1	4380
38.6.2.12	MBIST1FTAG2 — Memory BIST1 FTAG Signature Value Register 2	4381
38.6.2.13	BSEQ0ST — BIST Sequencer Status Register	4382
38.6.2.14	BSEQ0STB — BIST Sequencer Inverted Status Register	4383
38.6.2.15	BISTST — BIST Result Register	4384
38.6.2.16	BSEQ0SEL — BIST Scenario Select Register	4385
38.6.2.17	BSEQ0CTL — BIST Skip Control Register	4387
38.6.3	Functions	4388
38.6.3.1	BIST Execution Condition	4388
38.6.3.2	BIST Result Confirmation	4389
38.6.4	Operation	4389
38.7	MISG	4390
38.7.1	Overview	4390
38.7.2	Block Diagram	4391
38.7.2.1	MISG	4391
38.7.2.2	Signature Generation	4392
38.7.2.3	Signature Compare Unit	4393
38.7.3	Functional Specifications	4394
38.7.3.1	Conditions for Signature Generation	4394
38.7.3.2	Automatic Signature Comparison	4396
38.7.3.3	Data Counter	4396
38.7.3.4	Error Notification	4396
38.7.4	Register Specifications	4397
38.7.4.1	Register Map	4397
38.7.4.2	MISR1 — Multi Input Signature Register 1	4398
38.7.4.3	MISR2 — Multi Input Signature Register 2	4399
38.7.4.4	MISRCDR — Calculation Data Register	4400
38.7.4.5	MISRCR — Control Register	4401
38.7.4.6	MISRBASEADR — Monitoring Base Address Register	4402
38.7.4.7	MISRADRMSK — Monitoring Address Mask Register	4403
38.7.4.8	MISRDCNTCTL — Data Counter Control Register	4404
38.7.4.9	MISRDCNT — Data Counter Register	4405

38.7.4.10	MISRCMPCTL — Comparator Control Register	4406
38.7.4.11	MISRCMPERSTR — Compare Error Status Register	4407
38.7.4.12	MISRCMPERRSTC — Compare Error Status Clear Register	4408
38.7.4.13	MISRERRCTL — Error Notification Control Register	4409
38.7.4.14	MISRCMPDSEL — Compare Data Select Register	4410
38.7.4.15	MISRCMPDCNT — Compare Data Counter Register	4411
38.7.5	Usage	4412
38.7.5.1	Usage Example 1	4412
38.7.5.2	Usage Example 2	4412
38.7.5.3	Recommended Clear Flow of Error Flag	4413
38.8	ECM	4414
38.9	Voltage Monitor	4414
38.10	Clock Monitor	4414
38.11	Data CRC Function	4414
Section 39	Error Control Module (ECM)	4415
39.1	Features	4415
39.1.1	Number of Units and Channels	4415
39.1.2	Register Base Address	4415
39.1.3	Clock Supply	4416
39.1.4	Interrupt Requests	4416
39.1.5	Reset Sources	4417
39.1.6	External Input and Output Signals	4417
39.2	Overview	4418
39.2.1	Specification Overview	4418
39.2.2	Block Diagram	4419
39.2.3	Error Input	4420
39.2.4	Operations for Error Output	4441
39.2.5	<u>ERROROUT_M</u> and <u>ERROROUT_C</u> Behavior at Reset	4442
39.2.5.1	Dynamic Mode Enable	4442
39.2.5.2	Dynamic Mode Disable	4442
39.2.6	Loop-Back Function	4442
39.2.7	Pseudo Error Generation	4443
39.2.8	Error Status	4443
39.2.9	Register Protection	4443
39.2.10	Timeout Function for Interrupt Processing	4443
39.2.11	Masking of “Error Clear Trigger Registers”	4443
39.2.12	DCLS Error Interrupt (EI Level) and EI Level Interrupt	4444
39.3	Register Specification	4445
39.3.1	List of Registers	4445
39.3.2	ECMmESET — ECM Master/Checker Error Set Trigger Register (m = M/C)	4447
39.3.3	ECMmECLR — ECM Master/Checker Error Clear Trigger Register (m = M/C)	4448
39.3.4	ECMmESSTR0 to ECMmESSTR9 — ECM Master/Checker Error Source Status Register 0 to 9 (m = M/C)	4449

39.3.5	ECMEPCFG — ECM Error Pulse Configuration Register	4452
39.3.6	ECMMICFG0 to ECMMICFG9 — ECM Maskable Interrupt Configuration Register 0 to 9	4453
39.3.7	ECNMICFG0 to ECNMICFG9 — ECM FE Level Interrupt Configuration Register 0 to 9	4456
39.3.8	ECMIRCFG0 to ECMIRCFG9 — ECM Internal Reset Configuration Register 0 to 9	4459
39.3.9	ECMEMK0 to ECMEMK9 — ECM Error Mask Register 0 to 9	4462
39.3.10	ECMESSTC0 to ECMESSTC9 — ECM Error Source Status Clear Trigger Register 0 to 9	4465
39.3.11	ECMKCPROT — ECM Key Code Protection Register	4468
39.3.12	ECMPE0 to ECMPE9 — ECM Pseudo Error Trigger Register 0 to 9	4469
39.3.13	ECMDTMCTL — ECM Delay Timer Control Register	4472
39.3.14	ECMDTMR — ECM Delay Timer Register	4473
39.3.15	ECMDTMCMP — ECM Delay Timer Compare Register	4474
39.3.16	ECMMIDTMCFG0 to ECMMIDTMCFG9 — ECM Maskable Interrupt Delay Timer Configuration Register 0 to 9	4475
39.3.17	ECNMIDTMCFG0 to ECNMIDTMCFG9 — ECM FE Level Interrupt Delay Timer Configuration Register 0 to 9	4478
39.3.18	ECMEOCCFG — ECM Error Output Clear Invalidation Configuration Register	4481
39.3.19	ECMPEM — ECM Pseudo Error Mask Register	4482
Section 40	Data CRC Function B (DCRB)	4483
40.1	Features	4483
40.1.1	Number of Units	4483
40.1.2	Register Base Addresses	4484
40.1.3	Clock Supply	4484
40.1.4	Interrupt Requests	4484
40.1.5	Reset Sources	4484
40.1.6	External Input/Output Signals	4484
40.2	Overview	4485
40.2.1	Functional Overview	4485
40.2.2	Block Diagram	4486
40.2.3	Configuration of CRC Calculation	4487
40.3	Registers	4489
40.3.1	List of Registers	4489
40.3.2	DCRBnCIN — DCRB Input Register	4490
40.3.3	DCRBnCOUT — DCRB Data Register	4491
40.3.4	DCRBnCTL — DCRB Control Register	4492
40.4	Operation	4494
Section 41	Flash Memory	4495
41.1	Features	4495
41.2	Structure of Memory	4497

41.2.1	Mapping of Code Flash Memory	4497
41.2.2	Mapping of Data Flash Memory	4499
41.2.3	Mapping of Flash Extra Area	4500
41.2.4	Examples of the Chip Writing.....	4501
41.3	Operating Modes Associated with Flash Memory	4502
41.4	Functional Overview	4503
41.5	Serial Programming	4507
41.6	Communication Modes	4507
41.6.1	Asynchronous Flash Programming Interface — 2-Wire UART	4507
41.6.2	Synchronous Flash Programming Interface CSI	4507
41.6.3	CAN-FD/CAN Bootstrap	4508
41.6.4	Selection of Communication Method	4509
41.7	Self-Programming.....	4510
41.7.1	Outline	4510
41.7.2	Background Operation	4511
41.7.3	Dual Operation	4511
41.8	Reading Flash Memory.....	4512
41.8.1	Reading Code Flash Memory	4512
41.8.2	Parallel Access to the Code Flash memory.....	4512
41.8.3	Reading Data Flash Memory	4512
41.9	Description of Registers.....	4513
41.9.1	Register Base Address	4513
41.9.2	Registers Related to Write and Erase Protection of Flash Memory	4513
41.9.2.1	FHVE15 — FHVE15 Control Register	4514
41.9.2.2	FHVE3 — FHVE3 Control Register	4515
41.9.3	Registers Related to Product Information	4516
41.9.3.1	PRDNAME _n ; n = 1 to 4 — Product Name Storage Register (n = 1 to 4)	4517
41.9.3.2	PRDNUM — Product Number / Product Version Storage Register	4518
41.10	Configuration Setting Area (Option Bytes, Reset Vector)	4519
41.10.1	Reset Vector PE.....	4520
41.10.2	OPBT0 — Option Byte 3 to 0 Bit Arrangement.....	4521
41.10.3	OPBT1 — Option Byte 7 to 4 Bit Arrangement.....	4522
41.10.4	OPBT2 — Option Byte 11 to 8 Bits Arrangement.....	4523
41.10.5	OPBT3 — Option Byte 15 to 12 Bits Arrangement.....	4524
41.10.6	OPBT4 — Option Byte 19 to 16 Bit Arrangement.....	4526
41.10.7	OPBT5 — Option Byte 23 to 20 Bit Arrangement.....	4529
41.10.8	OPBT6 — Option Byte 27 to 24 Bit Arrangement.....	4530
41.10.9	OPBT7 — Option Byte 31 to 28 Bit Arrangement.....	4531
41.10.10	OPBT8 — Option Byte 35 to 32 Bit Arrangement.....	4532
41.11	Block Protection Area	4533
41.12	Usage Notes	4534

Section 42	Basic Hardware Protection (BHP)	4536
42.1	Features	4536
42.2	Security Functions in Serial Programming Mode	4537
42.3	Security Functions in Debug Interfaces	4537
42.3.1	Security Levels and State of Restricting the Connection of Debug Interfaces	4538
42.3.2	The Case of CAN BootStrap Mode	4540
42.4	Security Functions in Connection of RHSIF Link Partner	4541
42.5	Code Flash / Data Flash Protection	4542
42.5.1	Block Protection Function	4544
42.5.2	OTP (One Time Programmable) Function	4546
42.5.3	Test Mode Entry Protection	4547
42.6	Security Settings Area	4548
42.6.1	S_OPBT0 — OCD ID Related Option Bytes	4549
42.6.2	S_OPBT1 — TEST ID /C-TEST ID Related Option Bytes	4550
42.6.3	S_OPBT2	4551
42.6.4	S_OPBT3 — Customer ID Related Option Bytes	4552
42.6.5	S_OPBT4 — Data Flash ID Related Option Bytes	4553
42.6.6	S_OPBT5 — Serial Programmer ID Related Option Bytes	4554
42.6.7	S_OPBT6 — RHSIF ID Related Option Bytes	4555
42.6.8	S_OPBT7	4556
42.6.9	OCDIDn — OCD ID (n = 0 to 7)	4557
42.6.10	SPIDn — Serial Programmer ID (n = 0 to 7)	4558
42.6.11	CUSTOMERIDn — Customer ID (n = 0 to 7)	4559
42.6.12	DATAFLASH IDn — Data Flash ID (n = 0 to 7)	4560
42.6.13	CTESTIDn — C-Test ID (n = 0 to 7)	4561
42.6.14	RHSIFIDn — RHSIF ID (n = 0 to 7)	4562
Section 43	Intelligent Cryptographic Unit/Master (ICUMD)	4563
Section 44	Secure Watchdog Timer (SWDT)	4564
Section 45	Debugging and Calibration	4565
45.1	Debug Interface	4565
45.2	Run Control Functions	4567
45.3	Calibration Functions	4569
45.4	Trace Control Functions	4570
45.5	Performance Measurement Function	4571
45.6	Debug Support Function	4572
45.7	Peripheral Break Control	4573
45.8	Debugging-Related Registers	4574
45.8.1	Registers equipped according to the product specifications	4574
45.8.1.1	Register Base Address	4574

45.8.1.2	Clock Supply	4574
45.8.1.3	Reset Source	4574
45.8.1.4	List of Registers	4574
45.8.1.5	Product Specification Registers	4575
45.9	Aurora Trace Interface	4576
45.9.1	Overview	4576
45.9.2	Summary of Specifications	4576
45.10	Cautions on Using On-Chip Debugger	4577
Section 46	Boundary Scan	4578
46.1	Features of Boundary Scan	4578
46.1.1	Products that Incorporate Boundary Scan	4578
46.1.2	Register Base Address	4578
46.1.3	Clock Supply	4578
46.1.4	Interrupt Requests	4578
46.1.5	Reset Sources	4578
46.1.6	External Input/Output Signals	4579
46.2	Overview	4580
46.2.1	Functional Overview	4580
46.2.2	Block Diagram	4581
46.3	Registers	4582
46.3.1	List of Registers	4582
46.3.2	Details of Registers	4583
46.3.2.1	SDIR — Instruction Register	4583
46.3.2.2	SDID — ID Register	4583
46.3.2.3	SDBPR — Bypass Register	4583
46.3.2.4	SDBSR — Boundary Scan Register	4583
46.4	Operation	4584
46.4.1	TAP Controller	4584
46.4.2	Supported Instructions	4585
46.4.2.1	BYPASS	4585
46.4.2.2	SAMPLE/PRELOAD	4585
46.4.2.3	EXTEST	4585
46.4.2.4	CLAMP	4585
46.4.2.5	HIGHZ	4585
46.4.2.6	IDCODE	4586
46.4.3	Supported Pins	4587
46.4.3.1	Supported Pin Information	4587
46.4.3.2	Usage Note	4588
Section 47	Electrical Characteristics	4589
Appendix A	List of Registers	4590
Appendix B	Package	4591

Section 1 Overview

The RH850/E2x-FCC1, RH850/E2M are products of the single-chip microcontroller RH850 series from Renesas Electronics.

This section gives an overview of the RH850/E2x-FCC1 and RH850/E2M.

1.1 Outline

This product is a 32-bit single-chip microcontroller that incorporates multiple CPUs of the RH850 architecture, code flash, data flash, RAM, DMA controllers, high-speed communication interfaces including RHSB (microsecond bus), CAN, FlexRay, Ethernet, RHSIF, LIN, SENT, PSI5-S, three types (sequential conversion type, delta-sigma type and cyclic type) of A/D converters, peripheral functions including digital filter processing, an Advance Timer Unit (ATU-V) and GTM that are best suited for high-speed accurate power train control. This product also conforms to the Automotive Safety Integrity Level (ASIL) that is currently in high demand in the automotive field.

(1) Includes multiple RH850 cores

This product contains multi RH850G4MH cores (each CPU is referred to as CPU0 and CPU1 hereafter). CPU0 and CPU1 support RISC-type instruction sets and have significantly improved the instruction execution speed with basic instructions (one clock cycle per instruction) and optimized 10-stage pipeline configurations. Furthermore, this product also supports multiplication instructions using a 32-bit hardware multiplier, saturated product-sum operation instructions, and bit manipulation instructions as instructions best suited for various applications such as automobile power train control.

Two-byte basic instructions and high-level language instructions improve object code efficiency for the C compiler and reduce the program size. Furthermore, this product is suited for advanced real-time control applications by offering a high-speed interrupt response time including the processing time of the on-chip interrupt controller.

(2) On-chip code flash and data flash

This product incorporates an 8-MB code flash allowing high-speed access, which enables each CPU to access this flash memory efficiently. This memory can be reprogrammed while placed in an application system. This can shorten the system development period and significantly improve serviceability after the system is delivered.

This product also has a 224-KB data flash that is available for storing EEPROM data.

(3) A variety of peripheral functions

This product incorporates timers, a Digital Filter Engine (DFE), three types of A/D converters (SAR-ADC, DS-ADC, and Cyclic ADC) and other functions that are best suited for reducing hardware and software control loads in the automobile power train control (e.g. engine, transmission). In addition, this product incorporates standard peripheral functions (UART, CAN, and RHSB) for automotive applications. It also has the global standard on-chip NEXUS JTAG as a debug interface. This allows construction of systems without the need to provide these functions externally, which reduces cost, quantity of components and PCB footprint.

(4) Functional safety support

This product provides with several dedicated functionalities including a Lock-Step Dual Core configuration for the CPU, memory protection with ECC, bus protection with ECC/EDC, peripheral module protection, and voltage/clock monitors to support the functional safety standard (ISO26262) required in automotive applications.

(5) Security support

This product provides various security features and utilizes the Intelligent Cryptographic Unit/Master (ICUMD) as an on-chip Hardware Security Module (HSM).

1.2 Features

CPU0/1 core	RH850G4MH: 2 units (for high-speed operation and control)
CPU0/1 cache memory	16 KB
Minimum CPU0/1 instruction execution time	2.5 ns (during internal 400 MHz operation)
General CPU0/1 registers	Thirty-two 32-bit registers
FP-SIMD	<p>CPU0 and CPU1 have their own respective FP-SIMD units</p> <p>32-bit x 1/2/4-way operations</p> <p>Thirty-two 128-bit dedicated registers independent from CPU/FPU</p> <p>Supports single-precision (32-bit) only</p> <p>Supports IEEE754 compliant data types and exceptions</p> <p>Exceptions can be enabled/disabled individually</p> <p>Rounding modes: To nearest/Toward 0/Toward + ∞/Toward - ∞</p> <p>Handling of denormalized numbers: Truncation to zero / IEEE754 compliant exception</p>
CPU0/1 instruction sets	<p>Signed multiplication (32 bits × 32 bits → 64 bits): 1 to 2 CPU clocks</p> <p>Saturated operation instructions (with overflow/underflow detection function)</p> <p>32-bit arithmetic/logical shift instructions: 1 CPU clock</p> <p>Bit manipulation instructions</p> <p>Load/store instructions with long/short formats</p> <p>Signed load instructions</p>
Memory space	4-GB address space (common to program and data)
Code flash	<p>Two types of memory area</p> <ul style="list-style-type: none"> • User area: 8 MB (common to CPU0, CPU1, ICUMD) • User boot area: 64 KB high-speed reading through cache enabled
Data flash	224 KB (common to CPU0, CPU1 and ICUMD)
RAM	<p>Local RAM: 32 KB (CPU0/1)</p> <p>Cluster RAM: 704 KB (common to CPU0 and CPU1) (64 KB: Retention RAM)</p> <p>Cluster Emulation RAM: 1 MB (E2x-FCC1 only)</p> <p>Global Emulation RAM: 1.5 MB (E2x-FCC1 only)</p>
Interrupts/exceptions	<p>1 nonmaskable interrupt (NMI pin)</p> <p>2 FE level interrupts</p> <p>512 maskable interrupts (high-speed: 32, low-speed: 480)</p> <p>Simultaneous distribution of interrupt sources to multiple cores (CPU0/CPU1)</p> <ul style="list-style-type: none"> • Applicable sources: non-maskable interrupt (NMI pin), FE level interrupts, 32 high-speed maskable interrupts <p>External interrupt input function (IRQ pins)</p> <p>Software interrupt function (SINT)</p> <p>Inter-processor interrupt function (IPIR)</p> <p>16-level priority specifiable for maskable interrupts</p> <p>For RH850 G4MH exceptions, see Section 3, CPU system.</p>
sDMAC controller	<p>32 channels incorporated (16 channels × 2 units)</p> <p>Transfer data length: 1 byte, 2 bytes, 4 bytes, 8 bytes, 16 bytes, 32 bytes, 64 bytes</p> <p>Parallel reads and writes (fly-by)</p> <p>Address mode: dual address mode</p> <p>Transfer requests: auto request, peripheral hardware request</p> <p>Bus modes: normal speed mode, slow speed mode</p> <p>Arbitration modes: fixed priority mode, round-robin mode</p> <p>Interrupt requests: termination of descriptor step, termination of data transfer, occurrence of address error</p> <p>Descriptor memory: 8 KB (shared among all channels)</p> <p>Scatter-gather transfer</p> <p>Transfer target: On-chip memory, on-chip peripheral modules (excluding the DTS and sDMAC)</p>

DTS controller	<p>128 channels incorporated</p> <p>Transfer unit: 8 bits/16 bits/32 bits/64 bits/128 bits</p> <p>64-bit × 2-burst transfer</p> <p>Dual-address transfer mode</p> <p>Address reloading function</p> <p>Chain transfer function</p> <p>Three transfer modes: Single transfer, block transfer 1 (specified by number of transfer times), and block transfer 2 (specified by address count)</p> <p>Transfer target: On-chip memory, on-chip peripheral modules (excluding the DTS and sDMAC)</p> <p>Transfer requests can be set by interrupt sources and software</p>
I/O	<p>Output driving ability of specific input/output pins is selectable</p> <p>Inversion or non-inversion of output values of specific input/output pins is selectable</p> <p>Pull-down or pull-down off of specific input/output pins is selectable</p>
Safety functions	<p>Flash memory ECC error detection function</p> <p>RAM ECC error detection function</p> <p>Peripheral module RAM ECC error detection function (e.g. DTS, CAN, FlexRay, DFE)</p> <p>Clock monitor</p> <p>Error Control Module (ECM)</p> <p>Duplexing of modules (e.g. CPU0/1, ECM, error output pins)</p> <p>Automatic Power-on BIST execution after reset</p> <p>Shut-down BIST execution selection after System Reset 2</p>
Error Control Module (ECM)	<p>Collects information for each error check system and safety function and indicates error status</p> <p>When an error is detected, an error signal can be output from the error pin to an outside</p> <p>Interrupts and internal reset signals can be generated upon detection of an error</p> <p>Provided with a function to generate a pseudo-error for debugging and self-diagnosis</p>
Data CRC function B (DCRB)	<p>The data CRC (Cyclic Redundancy Check) function can verify or generate data streams protected by a CRC with various lengths and different bit widths</p>
Multi-Input Signature Generator (MISG)	<p>Monitors write access to specific addresses by the respective CPUs, and generate a 64-bit signature using the write data</p>
Window Watchdog Timer (WDTB)	<p>2 units incorporated</p> <p>Can generate a signal to the ECM when a counter overflows (timer expires)</p> <p>Can generate an interrupt at 75% of the counter overflow value</p> <p>An interrupt request can be generated at any function of the counter value</p> <p>A window open period can be set to any function of the counter value</p>
Advanced Timer Unit V (ATU-V)	<p>Timer A: 32-bit input capture × 8 channels</p> <p>Timer B: Angle clock generating timer × 1 channel</p> <p>Timer C: 32-bit input capture/output compare × 44 channels</p>
(ATU and GTM cannot be used at the same time)	<p>Timer D: 32-bit one-shot pulse × 36 channels 32-bit input capture × 36 channels 32-bit output compare × 36 channels</p> <p>Timer E: 24-bit PWM × 36 channels</p> <p>Timer F: 32-bit event counter × 16 channels</p> <p>Timer G: 32-bit interval timer × 10 channels</p>
OS Timer (OSTM)	<p>Three units incorporated</p> <ul style="list-style-type: none"> • A 32-bit timer assuming use by an OS • Interval timer mode or free-running timer mode selectable • Synchronous start between units available
Peripheral Interconnection function (PIC1)	<p>One unit incorporated</p> <ul style="list-style-type: none"> • Synchronous operation between the OSTM timer and other timer input/outputs can be connected
Peripheral Interconnection function (PIC2)	<p>Two units incorporated</p> <ul style="list-style-type: none"> • Maps peripheral IP outputs to triggers
Serial Communication Interface 3 (SCI3)	<p>4 channels incorporated</p> <ul style="list-style-type: none"> • Clock synchronization or start-stop system selectable • Full-duplex communication enabled • Arbitrary bit rate selectable by the on-chip baud rate generator • LSB first or MSB first selectable

Clocked Serial Interface H (CSIH)	<p>6 channels incorporated</p> <ul style="list-style-type: none"> • Three-wire serial synchronous data transfer • Master mode or slave mode selectable • Six or four chip select output signals are selectable • Arbitrary bit rate is selectable by the on-chip baud rate generator
CANFD interface (RS-CANFD)	<p>5 channels incorporated</p> <ul style="list-style-type: none"> • RS-CAN mode (RS-CAN software compatibility mode) <ul style="list-style-type: none"> – Conforming to CAN ISO-11898-1 (2003) – Transfer speed up to 1 Mbps – A total of 400 message buffers provided for 5 channels – Reception filtering • CAN-FD mode <ul style="list-style-type: none"> – Conforming to CAN-FD ISO 11898-1 (2015) – Transfer speed up to 8 Mbps – A total of 400 message buffers provided for 5 channels – Reception filtering
FlexRay (FLXA)	<p>1 unit incorporated</p> <ul style="list-style-type: none"> • Conforming to Protocol Specification v2.1 • Buffer size: A 8-KB space is divided into up to 128 sections (for transmission, reception, and receive FIFO) • Message filtering: ID filter, channel filter, cycle counter filter • Bit rate: 10 Mbps
LIN/UART interface (RLIN3)	<p>6 channels incorporated</p> <ul style="list-style-type: none"> • Conforming to LIN Protocol Spec versions 1.3, 2.0, 2.1, 2.2, and SAE J2602 • Three operating modes <ul style="list-style-type: none"> – LIN Master mode – LIN Slave mode – UART mode (half-duplex, full-duplex) • Arbitrary bit rate is selectable by the on-chip baud rate generator • LIN Self-test mode with internal data loop back
Renesas High-Speed Bus (RHSB)	<p>2 channels incorporated</p> <p>Communication module supporting Micro-Second Bus channels</p> <ul style="list-style-type: none"> • Down-stream communication • Up to two slaves selectable and configurable individually • Functions for emergencies
Single Edge Nibble Transmission (RSENT)	<p>17 channels incorporated</p> <ul style="list-style-type: none"> • Conforming to the SENT (Single Edge Nibble Transmission) protocol specified in the SAE J2716_201604 standard and the SPC (Short PWM Code) extension to the SENT specification • Unidirectional or bidirectional transfer is possible through a single pin • Bidirectional transfer is possible through two pins • Data transfer protected by a CRC is possible
High Speed Serial Peripheral Interface (HS-SPI)	<p>1 channel incorporated</p> <ul style="list-style-type: none"> • Three-wire serial synchronous data transfer • Master mode or Slave mode selectable • Built in DMA for communication data transfer
Peripheral Sensor Interface 5 serial communication module (PSI5-S)	<p>1 channel incorporated</p> <ul style="list-style-type: none"> • Support the UART based communication for PSI5 transceiver • Conformance with PSI5 protocol specification V2.2 • Generate a PSI5 message from the UART transfer data • The bit rate of the UART can be set by the built-in baud rate generator
RHSIF	<p>1 channel incorporated</p> <ul style="list-style-type: none"> • Asynchronous high speed LVDS interface based on IEEE 1596.3-1996 reduced range link • Asynchronous high speed LVDS interface supporting maximum data rates of 80 Mbps and 160 Mbps • Four channels, including one channel with data streaming capability • Bus master interface which is used by a target node to access shared memory

Ethernet Controller (ETNC)	<p>1 channel incorporated</p> <ul style="list-style-type: none"> • Conformance with the IEEE 802.3 MAC layer standard • PHY interface: MII (Media Independent Interface) and RMI (Reduced Media Independent Interface) • Supports 10 Mbps and 100 Mbps • Supports full-duplex and half-duplex modes • Built-in DMA transfer function
Analog to Digital Converter (ADCH)	<p>80 channels incorporated</p> <ul style="list-style-type: none"> • A/D conversion method: Successive approximation • Configuration of analog input pins <ul style="list-style-type: none"> ADCH0/ADCH1 dedicated inputs: 20/12 ADCH0/ADCH1 shared inputs: 8 ADCH2/ADCH3 dedicated inputs: 20/12 ADCH2/ADCH3 shared inputs: 8 • Resolution: 12-bit/10-bit • Conversion speed: 1.0 μs • Scan groups for five systems for each converter • Two scan modes (multicycle scan mode and continuous scan mode) • ADCH0: Up to 40 virtual channels • ADCH2: Up to 40 virtual channels • ADCH1: Up to 40 virtual channels • ADCH3: Up to 40 virtual channels • Two types of A/D conversion and addition functions incorporated • Converts the converted results to floating-point format • Can enter data directly to the digital filter and the ADC summation function and the Generic Timer Module • Safety functions • Supports an upper/lower-limit-excess-notice-function for the ADC Voltage Monitor Secondary Error Generator or ADC Boundary Flag Generator in each virtual channel
Delta-Sigma Analog to Digital Converter (DSADC)	<p>26 channels incorporated</p> <ul style="list-style-type: none"> • A/D conversion method: Delta-Sigma modulation method • Configuration of analog input pins <ul style="list-style-type: none"> DSADC00: 8 single-ended inputs/4 differential inputs DSADC10: 4 single-ended inputs/2 differential inputs DSADC20: 2 single-ended inputs/1 differential inputs DSADC12: 4 single-ended inputs/2 differential inputs DSADC13: 4 single-ended inputs/2 differential inputs DSADC11: 4 single-ended inputs/2 differential inputs • Input type: Single-end input and differential input • Input gain function (PGA): $\times 1$, $\times 2$, $\times 4$, or $\times 8$ selectable • Calibration function • Can enter data directly to the digital filter and the Generic Timer Module. • Timestamp function • Supporting a boundary flag generating signal to the ADC Boundary Flag Generator • Safety functions
Cyclic Analog to Digital Converter (CADC)	<p>8 channels incorporated</p> <ul style="list-style-type: none"> • A/D conversion method: Cyclic conversion method • Configuration of analog input pins <ul style="list-style-type: none"> CADC00: 8 single-ended inputs /4 differential inputs • Input type: Single-ended input and differential input • Calibration function • Can enter data directly to the digital filter and the Generic Timer Module • Supporting a boundary flag generating signal to the ADC Boundary Flag Generator • Safety functions
Power supply	<p>E2x supports both the 5 V and 3.3 V power supplies with the exception of the core supply. The Power on/off sequence has no constraints</p>

Power supply voltage monitor	<ul style="list-style-type: none"> • The power supply voltage monitor is used for monitoring power domains E0VCC, VCC and VDD • The power supply voltage monitor has High-side (HDET) and Low-side (LDET) voltage detectors that detect if the monitored voltage is higher or lower than the specified voltage • Power supply voltage monitors have two types of detection functions: Primary detection and Secondary detection • The Primary detection function is performed by Voltage Monitor. The Secondary detection function is performed by the SAR-ADC • The Delay Monitor (DMON) assists the VMON, which detects the Low-side voltage of VDD • The Primary power supply voltage monitor can control on or off for the VMON high level and low level detection of VDD, and the high level detection of VCC and E0VCC • The Primary detection voltage value is fixed, the secondary detection voltage value can be set by the SAR-ADC • If the Secondary power supply voltage monitor detects voltage failures, it notifies the error to the ECM. Based on the settings in the ECM, it is possible to generate an interrupt or reset
Digital Filter Engine (DFE)	<p>16-channel digital filtering (FIR/IIR) functions</p> <ul style="list-style-type: none"> • FIR with up to 64 taps selectable • Up to a 6-order IIR is selectable • Filter coefficients and data are stored in the RAM to be connected • For filtered data: <ul style="list-style-type: none"> Accumulation processing or decimation processing can be enabled Peak-hold processing or comparison processing can be enabled • Directly inputs converted data from the SAR A/D converter or Delta-Sigma A/D converter or cyclic A/D converter to perform automatic filtering
Secure Watchdog Timer (SWDT)	<p>1 unit incorporated</p> <p>Can generate an error signal for the Reset Controller in response to an error</p> <p>Confirmation of matching with a specified Program Counter (PC) value of the CPU0</p>
Intelligent Cryptographic Unit Master (ICUMD)	<p>The ICUMD is an on-chip Hardware Security Module (HSM)</p> <p>The ICUMD supports user-defined security services to the overall system based on cryptographic primitives</p>
On-Chip Debugging unit (OCD)	<p>NEXUS JTAG: One channel incorporated</p> <p>LPD (4-pin): One channel incorporated</p> <p>CAN Debug interface: One channel incorporated</p> <p>Aurora Trace Interface: One channel incorporated (E2x-FCC1 only)</p>
Boundary scan	<p>Supports boundary scan conforming to the IEEE1149.1 standard</p>
Clock controller	<p>The user is able to select the crystal resonance frequency (20 MHz or 40 MHz)</p> <p>Incorporates a crystal resonation circuit (Main OSC), which is used as a reference clock for the PLL</p> <p>Incorporates an internal oscillator, which is used as the start-up clock and backup clock</p> <p>Incorporates a PLL circuit to generate high speed internal clocks by multiplying the Main OSC input</p> <p>Generates clock pulses used inside the chip from the internal oscillator, main oscillator and PLL</p> <p>Software configurable external clock output</p>
Operating modes	<p>Operating modes</p> <ul style="list-style-type: none"> • Normal Operating Mode • User Boot Mode • Serial Programming Mode
Standby controller	<p>This product supports various power-down modes</p> <p>The power consumption of the device can be reduced by selecting one of the following modes:</p> <ul style="list-style-type: none"> • Power off standby mode • HALT Mode • Emulation/Instrumentation RAM and Aurora retention Mode (E2x-FCC1 only) <p>Module standby</p>

Reset controller	<p>6 reset functions</p> <ul style="list-style-type: none"> – Power Up Reset – System Reset 1 – System Reset 2 – Application Reset – Module Reset – JTAG Reset <p>External reset output pin: $\overline{\text{RES_OUT}}$ Automatic RAM initialization after reset</p>
Clock monitor	<ul style="list-style-type: none"> • Up to 7 clock monitors depending on the device configuration • Detects clock disturbances that results in a frequency lower or higher than the target frequency, and sends an error notification to the ECM <p>Supports the self-diagnosis function</p>
Temperature sensor	<p>1 sensor incorporated</p> <ul style="list-style-type: none"> • Out of range detection of temperature • Operating modes <ul style="list-style-type: none"> – Single measurement mode – Continuous measurement mode • Interrupt generation <ul style="list-style-type: none"> – Temperature Measurement End Interrupt (INTOTSOTI) – Temperature Rise/Drop Interrupt (INTOTSOTULI) – Temperature Alarm Error (INTOTSOTABE) – Temperature Sensor Error (INTOTSOTE)
Generic Timer Module (GTM) (ATU and GTM cannot be used at the same time)	<p>GTM v3.1.5.1 is a modular timer unit and consists of the following submodules.</p> <ul style="list-style-type: none"> • Advanced Routing Unit (ARU) • Broadcast Module (BRC) • Parameter Storage Module (PSM) • Clock Management Unit (CMU) • Cluster Configuration Module (CCM) • Time Base Unit (TBU) • Timer Input Module (TIM) • Timer Output Module (TOM) • ARU-connected Timer Output Module (ATOM) • Dead Time Module (DTM) • Multi Channel Sequencer (MCS) • Memory Configuration (MCFG) (E2x-FCC1 only) • TIM0 Input Mapping Module (MAP) • Digital PLL (DPLL) • Sensor Pattern Evaluation (SPE) • Interrupt Concentrator Module (ICM) • Output Compare Unit (CMP) • Monitoring Unit (MON)
Encoder Timer (ENCA)	<ul style="list-style-type: none"> • Generation of the counter control signal from the encoder input signal, and count operation • Capture function for capturing the counter value with an external trigger signal • Compare function for compare match judgment with the count value • Two capture compare registers that can be set separately for capture operation and for compare operation • Interrupt mask function for masking the interrupt request signal output as a result of the compare match judgment during compare operation • Function for loading the value of the capture compare register to the counter upon underflow occurrence • The Encoder input signal can be applied to the timer counter clear condition • Edge or level can be selected for clearing the encoder input signal of the timer counter clear condition • Detection of counter overflow and underflow and output of error flags and error occurrence interrupts • Five interrupts: two capture compare interrupts, one counter clear interrupt, one overflow interrupt, and one underflow interrupt

Package	[E2x-FCC1] 373-pin plastic FBGA (0.8 mm ball pitch) (21mm × 21mm package size) 292-pin plastic FBGA (0.8 mm ball pitch) (17mm × 17mm package size)
	[E2M] 373-pin plastic FBGA (0.8 mm ball pitch) (21mm × 21mm package size) 292-pin plastic FBGA (0.8 mm ball pitch) (17mm × 17mm package size)

1.3 Application Fields

- Automotive field (including engine control system and transmission control system)

1.4 Ordering Information

Table 1.1 Product Name List

Product Name	Package	On-Chip ROM	Operating Temperature (Tj)	External Oscillator	Maximum Operating Frequency
R7F702002AEABA (RH850/E2M)	Plastic FBGA-373 0.8-mm ball pitch 21 mm × 21 mm	8 MB	max. 150°C	20/40 MHz	400 MHz
R7F702002AEABG (RH850/E2M)	Plastic FBGA-292 0.8-mm ball pitch 17 mm × 17 mm	8 MB	max. 150°C	20/40 MHz	400 MHz

FBGA is hereafter referred to as BGA unless the complete abbreviation is required.

The RH850/E2x-FCC1 has 2 variants. Select the proper product according to the table below.

Table 1.2 RH850/E2x-FCC1 List

Frequency	Package	
	FBGA-373	FBGA-292
400 MHz	R7F702Z04CEDBA (for E2M)	R7F702Z02CEDBG (for E2M)

1.5 Differences in the Specifications of FCC Products

The table below lists the differences in the specifications of FCC products.

Product		RH850/E2x-FCC1		
		For E2M	RH850/E2M	
CPU	Frequency		400	400
	Main Core		2	2
	Lockstep		2	2
	FPU		2	2
	Cache (per Main core)	instruction	16 KB (PBS 4w)	16 KB (PBS 4w)
		data	4 line (256 bit/line)	4 line (256 bit/line)
	FP-SIMD		2	2
	MPU regions per core		24	24
	RAM	Total RAM (Local RAM + Cluster RAM)		768 KB
Local RAM per core		32 KB	32 KB	
Cluster RAM (CRAM)		Total Size (Retention RAM included)	704 KB	704 KB
		Retention RAM (included in CRAM)	64 KB	64 KB
Cluster Emulation RAM		1 MB	No	
Global Emulation RAM		1.5 MB	No	
Instrumentation RAM		64 KB	No	
Trace RAM		64 KB	No	
Flash		Code Flash		8 MB
	User boot Area		64 KB	64 KB
	Data Flash	Total Size	224 KB	224 KB
		Dedicated ICUMD	32 KB	32 KB
DMA	sDMAC/DTS		32/128	32/128
Timers	ATU-V	Timer A	8	8
		Timer B	1	1
		Timer C	External: 40 Internal: 4	External: 40 Internal: 4
		Timer D	External: 36 Internal: 0	External: 36 Internal: 0
		Timer E	36	36
		Timer F	16	16
		Timer G	10	10
	ENCA		2	2
	OSTM	units	3	3
	WDTB	units	2	2
	SWDT	unit	1	1
	GTM		Yes	Yes
	GTM RAM		Yes	Yes

Product		RH850/E2x-FCC1		
		For E2M	RH850/E2M	
Comms	SCI3	channels	4	4
	HS-SPI	channel	1	1
	RLIN3	channels	6	6
	CSIH	channels	6	6
	RHSIF	channel	1	1
	RHSB	channels	2	2
	RS-CANFD	channels	5	5
		total buffers	400	400
	FlexRay	unit	1	1
	Ethernet	channel	1	1
		MII	Yes	Yes
		RMI	Yes	Yes
	RSENT	channels	17	17
PSI5-S	channel	1	1	
Safety	ASIL level		D	D
	ECM		Yes	Yes
	CRC(DCRB)		3	3
	MISG		Yes	Yes
	Clock Monitor		Yes	Yes
	Error injection for self-diagnosis of several safety mechanisms		Yes	Yes
	LBIST		Yes	Yes
	MBIST		Yes	Yes
	Bus ECC		Yes	Yes
DFE	units		2	2
	channels		12 + 4	12 + 4
ADC	Delta Sigma ADC	units	6	6
		units with 8 inputs each	1	1
		units with 4 inputs each	4	4
		units with 2 inputs each	1	1
		total inputs	26	26
	Cyclic ADC	units	1	1
		inputs per unit	8	8
		total inputs	8	8
	SAR ADC	units	4	4
		ADCH0/1/2/3 (Dedicated inputs)	20/12/20/12	20/12/20/12
		ADCH0/1 (Shared inputs)	8	8
		ADCH2/3 (Shared inputs)	8	8
		total inputs	80	80
		virtual channels per unit	40	40
		ADC timers	2	2

Product		RH850/E2x-FCC1	
		For E2M	RH850/E2M
Debug	NEXUS-JTAG	Yes	Yes
	AURORA	Yes	No
	LDU	Yes	Yes
	On-chip trace module	Yes	No
	Boundary scan	Yes	Yes
Security	BHP	Yes	Yes
	ICUMD	Yes	Yes
	Secure RAM	64 KB	64 KB
System Control	Temp Sensor	Yes	Yes
	Voltage Monitor	Yes	Yes
	Power Sequence Free	Yes	Yes

1.6 Pin Connection Diagram (Top View)

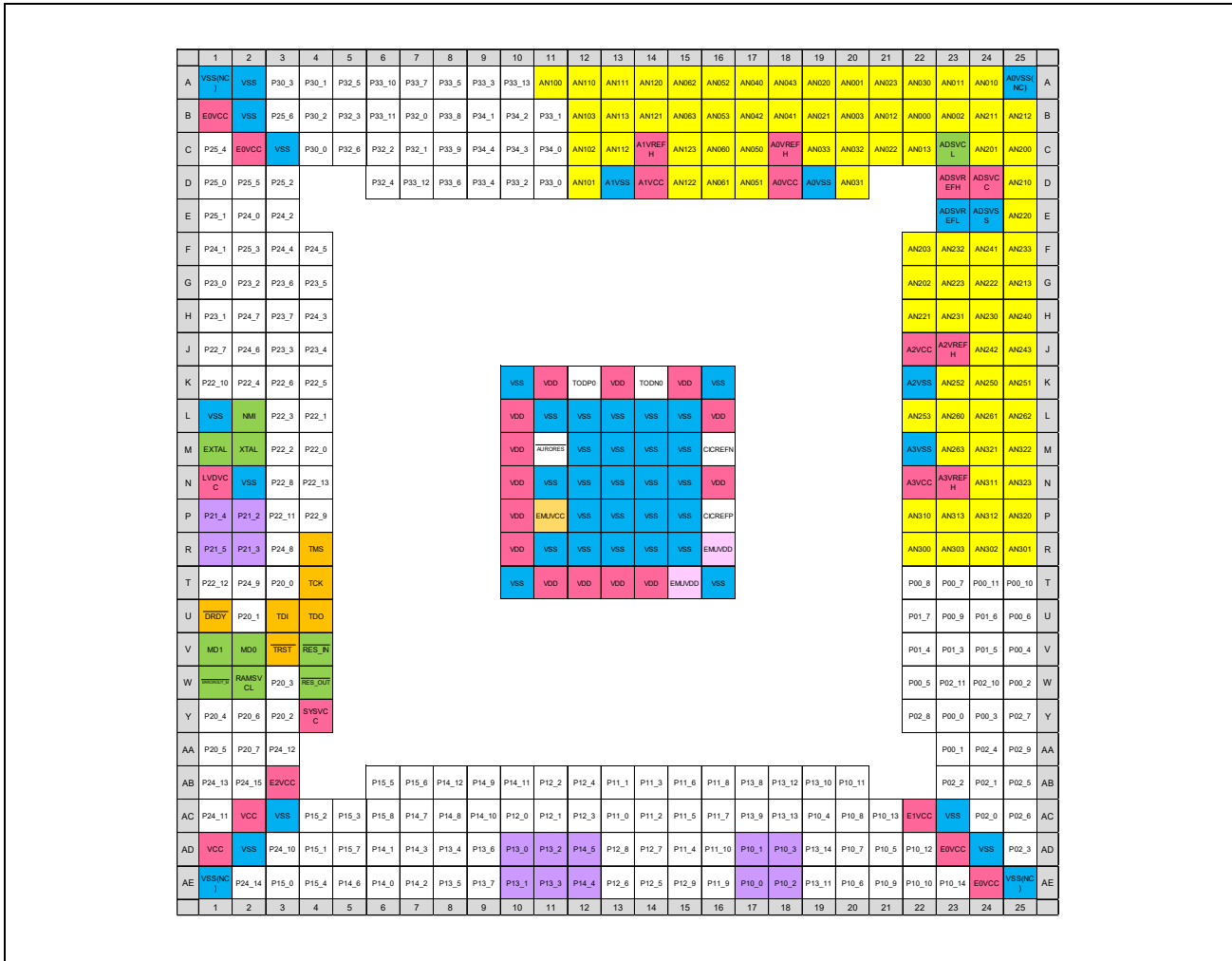


Figure 1.1 Pin Connection Diagram (FCC1-BGA373 E2M)

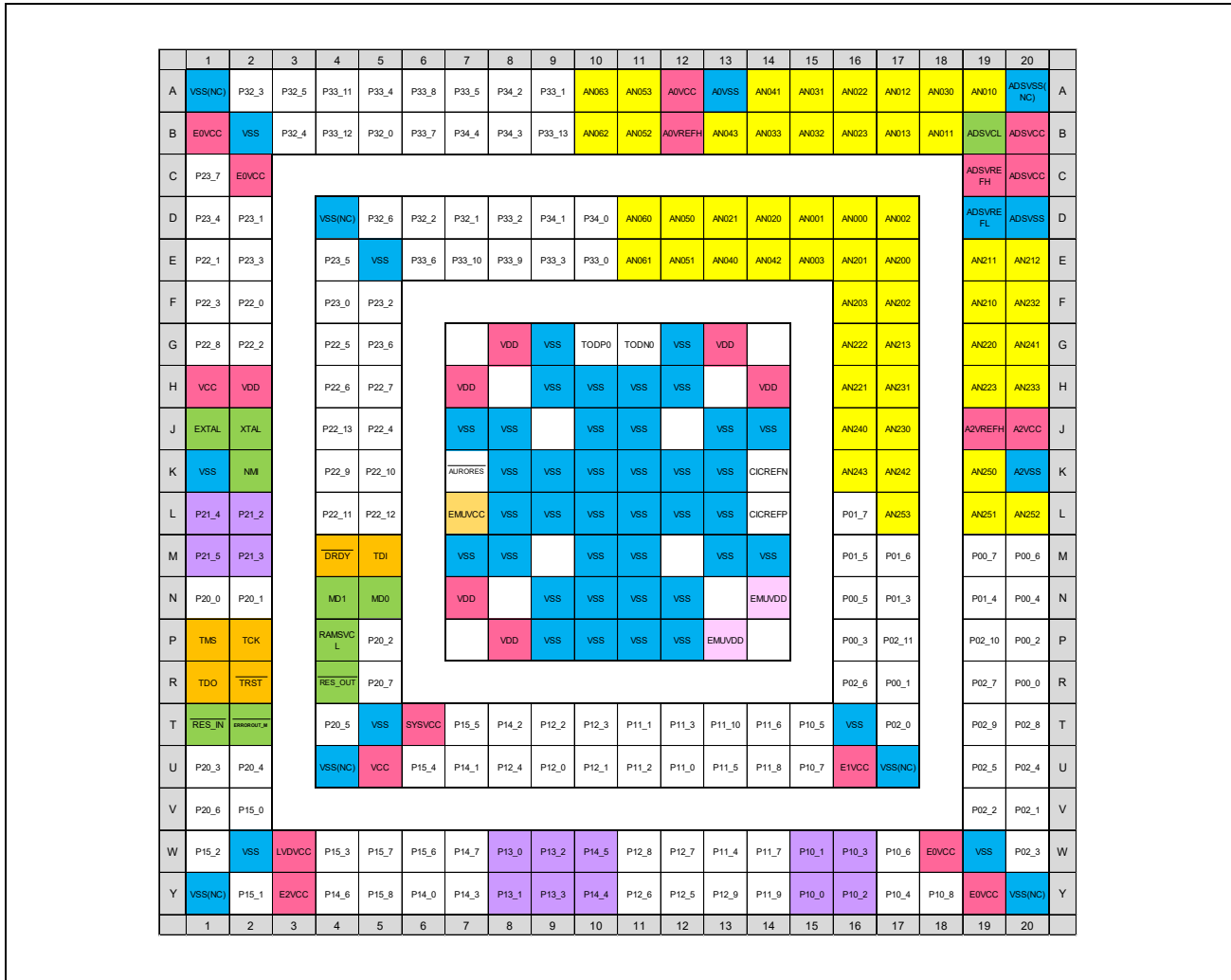


Figure 1.2 Pin Connection Diagram (FCC1-BGA292 E2M)

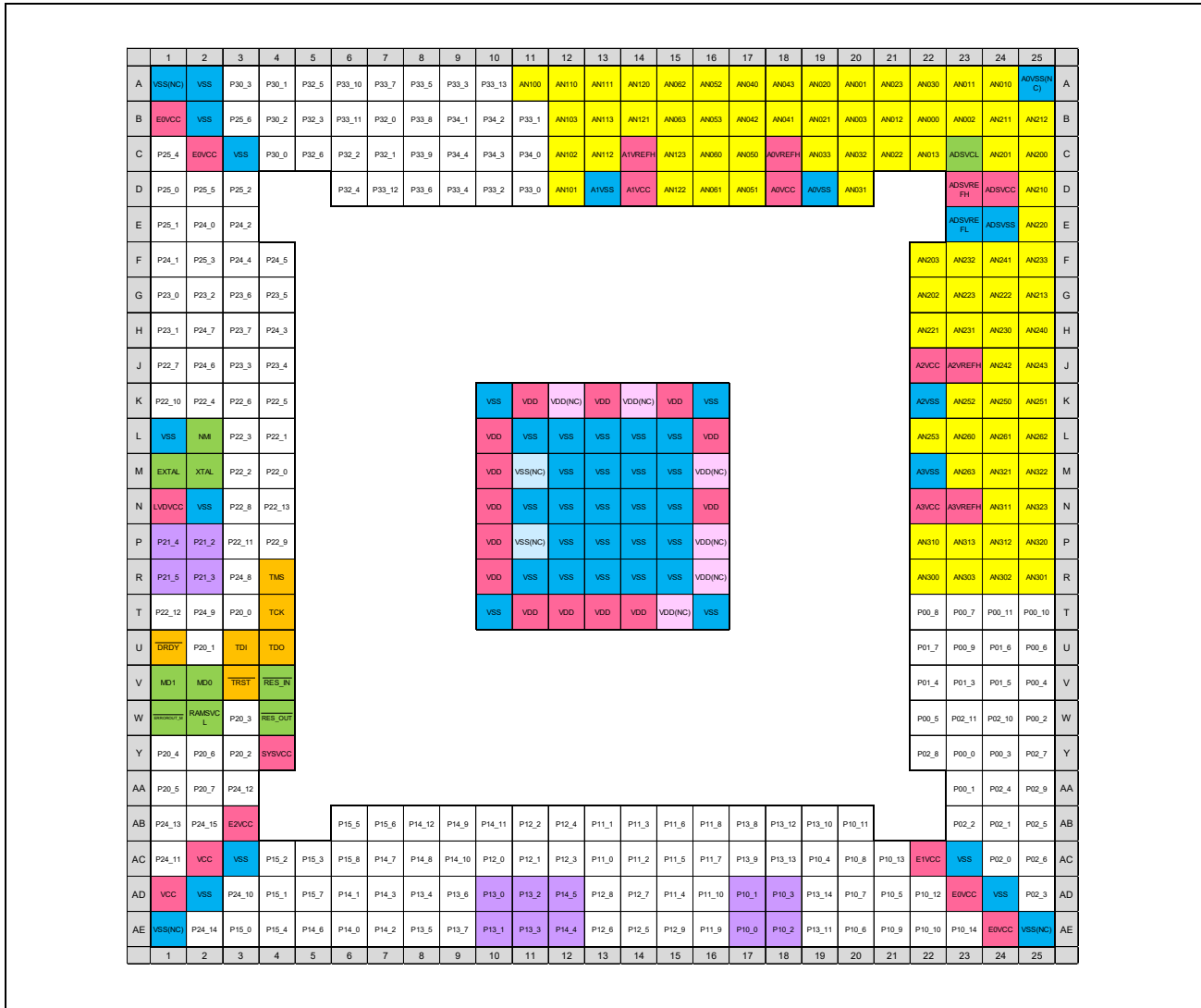


Figure 1.3 Pin Connection Diagram (BGA373 E2M)

1.7 Functional Block Configuration

Internal Block Diagram

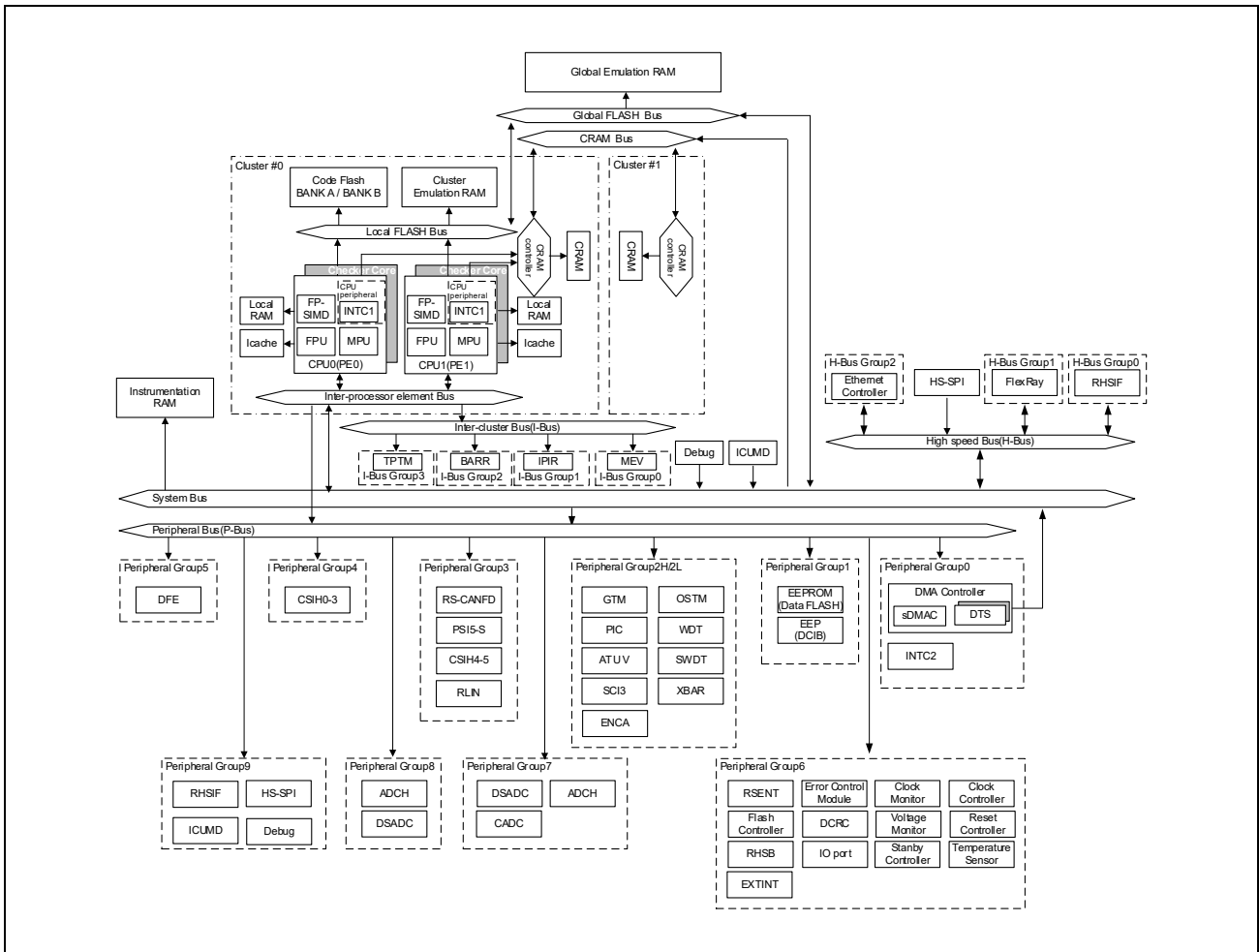


Figure 1.5 Internal Block Diagram (E2x-FCC1)

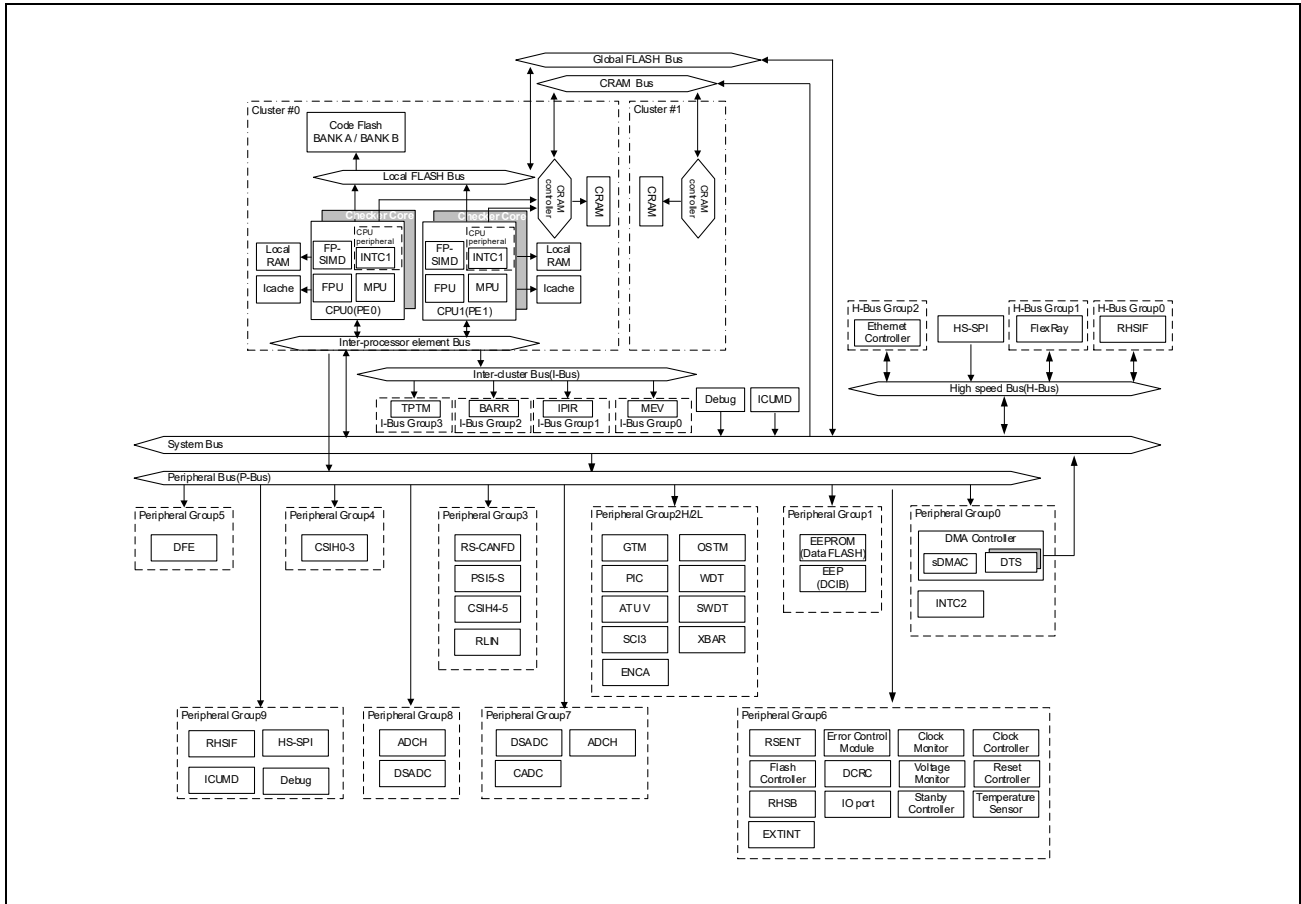


Figure 1.6 Internal Block Diagram (E2M)

CPU0 and CPU1 each have their own set of CPU peripherals. These CPU peripherals are assigned to the common address: CPU peripheral (self), separately from their respective address. Access by CPU0 to a register of a CPU peripheral (self) is to the register in the CPU peripheral of CPU0; access by CPU1 to a register of a CPU peripheral (self) is to the register in the CPU peripheral of CPU1.

Section 2 Pin Function

This section contains a generic description of the Pin Function Controller (the unit name is defined as PORT in this section). The first part of this section describes the features specific to RH850/E2x-FCC1 microcontrollers, including register base addresses and input/output signals. The ensuing sections describe the functions and registers of PORT.

CAUTION

Some pin functions may be absent from other mass-produced products even if the PKG is the same shape as the FCC1. For information regarding which pin functions are and are not present, see the user's manual for the product you are using.

2.1 Features of Pin Function

2.1.1 Port Group

This product has the following numbers of port groups.

Table 2.1 Indices

Index	Description
n	Throughout this section, each port group is identified by “n” (n = 00 to 57). For example, PMCN indicates the port mode control register of the Pn port.
m	Throughout this section, each port pin is identified by the index “m” (m = 0 to 15). For example, P00_7 indicates pin 7 of port group 00.

Table 2.2 Port Groups in This Product

Product Name	E2x-FCC1		E2M	
	BGA373	BGA292(E2M)	BGA373	BGA292
Number of Port Groups	22	19	22	19
Port Group Name	P00_m	m = 0 to 11	m = 0 to 11	m = 0 to 7
	P01_m	m = 3 to 7	m = 3 to 7	m = 3 to 7
	P02_m	m = 0 to 11	m = 0 to 11	m = 0 to 11
	P10_m	m = 0 to 14	m = 0 to 8	m = 0 to 8
	P11_m	m = 0 to 10	m = 0 to 10	m = 0 to 10
	P12_m	m = 0 to 9	m = 0 to 9	m = 0 to 9
	P13_m	m = 0 to 14	m = 0 to 3	m = 0 to 3
	P14_m	m = 0 to 12	m = 0 to 7	m = 0 to 7
	P15_m	m = 0 to 8	m = 0 to 8	m = 0 to 8
	P20_m	m = 0 to 7	m = 0 to 7	m = 0 to 7
	P21_m	m = 2 to 5	m = 2 to 5	m = 2 to 5
	P22_m	m = 0 to 13	m = 0 to 13	m = 0 to 13
	P23_m	m = 0 to 7	m = 0 to 7	m = 0 to 7
	P24_m	m = 0 to 15	—	—
	P25_m	m = 0 to 6	—	—
	P27_m	m = 0* ¹	m = 0* ¹	m = 0* ¹
	P30_m	m = 0 to 3	—	—
	P32_m	m = 0 to 6	m = 0 to 6	m = 0 to 6
	P33_m	m = 0 to 13	m = 0 to 13	m = 0 to 13
	P34_m	m = 0 to 4	m = 0 to 4	m = 0 to 4
	P40_m	m = 0 to 15	m = 0 to 15	m = 0 to 15
	JP0_m	m = 0 to 3, 5* ²	m = 0 to 3, 5* ²	m = 0 to 3, 5* ²

Note 1. The P27 port group does not have an input function and can only be used when the Option Byte PFC_RESO_CFG = 1 (See **Section 2.6.8, RES_OUT Function** for details). In addition, this port can only be used for debugger handshaking.

Note 2. TCK does not support port output; it only has a port input function.

If TDI/JP0_0 pin is used as debug interface, JPM0_0 and JPIBC0_0 must be set value after reset (JPM0_0 = 1, JPIBC0_0 = 1).

Table 2.3 Virtual Port Groups in This Product

Product Name		E2x-FCC1		E2M	
		BGA373	BGA292(E2M)	BGA373	BGA292
Number of Port Groups		11	11	11	11
Port Group Name	P41_m	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15
	P42_m	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15
	P43_m	m = 0 to 7	m = 0 to 7	m = 0 to 7	m = 0 to 7
	P50_m	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15
	P51_m	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15
	P52_m	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15
	P53_m	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15
	P54_m	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15
	P55_m	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15
	P56_m	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15
P57_m	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15	

Note: A virtual port is a port group that is not connected to an external pin; but is instead connected to an internal function. (See **Section 2.6.3, Virtual Port Function via RHSB (XBAR)**, **Section 2.6.4, Virtual Port Function via HSSPI** and **Section 2.6.5, Virtual Port Function via Analog** for details.)

2.1.2 Register Base Addresses

The PORT base addresses are shown in the following table. All port register addresses are expressed as offsets of the base address.

Table 2.4 Register Base Addresses

Base Address Name	Base Address	Bus Group
<PORT0_Base>	FF61 0000 _H	Peripheral Group 6
<JPORT0_Base>	FF62 0000 _H	Peripheral Group 6
<PORT1_Base>	FFC1 0000 _H	Peripheral Group 6
<JPORT1_Base>	FFC2 0000 _H	Peripheral Group 6

2.1.3 Clock Supplies

The PORT clock supplies are shown in the following table.

Table 2.5 Clock Supplies

Unit Name	Unit Clock Name	Supply Clock Name
PORT0	PCLK	CLK_LSB
JPORT0	PCLK	CLK_LSB
PORT1	PCLK	CLK_LSB
JPORT1	PCLK	CLK_LSB

Note: For details of port unit names refer to **Section 2.4, Registers**.

2.1.4 Interrupt Requests

The PORT interrupt requests from external pins are shown in the following table.

Table 2.6 Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
INTP00	IRQ0	External interrupt 0	504	206	114
INTP01	IRQ1	External interrupt 1	505	207	115
INTP02	IRQ2	External interrupt 2	506	208	116
INTP03	IRQ3	External interrupt 3	507	209	117
INTP04	IRQ4	External interrupt 4	508	210	118
INTP05	IRQ5	External interrupt 5	509	211	119
INTP06	IRQ6	External interrupt 6	510	212	120
INTP07	IRQ7	External interrupt 7	511	213	121

2.1.5 Reset Sources

The PORT reset sources are shown below.

PORT is initialized by the following reset sources.

Table 2.7 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
PORT0	All registers	✓	✓	✓	✓	—	—
JPORT0	All registers	✓	✓	✓	✓	—	—
PORT1	All registers	✓	✓	✓	✓	—	—
JPORT1	All registers	✓	✓	✓	✓	—	—

2.2 Overview

This product has various pins for input/output (I/O) functions, known as ports. The ports are organized into port groups. This product also has several control registers to allocate functions other than general-purpose I/O functions to the corresponding pins. See below for definitions of pin, port, and port group.

2.2.1 Functional Overview

2.2.1.1 Terms

The terms described in this section are defined as follows.

- Port group:

One port can have up to 16 pins, and the number of pins differs according to the port. All pins of a port share the same control registers.

- Port mode/Port:

In port mode, a pin functions as a general-purpose I/O pin.

- Alternative mode:

In alternative mode, a pin functions as an I/O pin of a peripheral function. Multiple peripheral functions may be multiplexed on a single pin, and control registers select the peripheral function to be used.

2.2.1.2 Overview of Pin Functions

Pins can operate in the following four different modes:

- Port mode (PMn.PMn_m = 0)

The pin operates as a general purpose I/O port in port mode. PMn.PMn_m selects input or output.

- S/W I/O control alternative mode (PMn.PMn_m = 1, PIPn.PIPn_m = 0)

The pin is operated by an alternative function in S/W I/O control alternative mode.

The selection between input and output is made by S/W via the PMn.PMn_m control bits.

- Direct I/O control alternative mode (PMn.PMn_m = 1, PIPn.PIPn_m = 1)

The pin is operated by an alternative function in direct I/O control alternative mode.

In contrast to S/W I/O control alternative mode, input/output is directly controlled by the alternative function in this mode.

- When the input buffer is disabled (PMn.PMn_m = 0, PMn.PMn_m = 1, PIBn.PIBn_m = 0)

A shoot-through current does not flow even if the pin level is in the Hi-Z state. Thus the pin does not need to be fixed to a high or low level externally.

Table 2.8 Pin Function Configuration (Outline)

Unit Name	Bit			I/O
	PMC	PM	PIPC	
Port mode	0	0	0/1	Output mode
		1	0/1	Input mode*1
S/W I/O control alternative mode	1	0	0	Output mode
		1	0	Input mode*2
Direct I/O control alternative mode		0/1	1	Controlled by alternative function*2

Note 1. The input buffer should be enabled (PIBCn_m = 1).

Note 2. When used as an input pin in alternative mode, be sure to set PIBCN_m = 0.

When a pin is operated in alternative mode (PMcn.PMCn_m = 1), one of many different alternative functions (up to 16) can be selected by the PFCn, PFCEn, PFCAEn, and PFCEAEn registers as shown in the following table.

- S/W I/O control alternative mode (PIPCn.PIPCn_m = 0):

Outputs (PMn_m = 0): ALT-OUT1 to ALT-OUT16

Inputs (PMn_m = 1): ALT-IN1 to ALT-IN16

- Direct I/O control alternative mode (PIPCn.PIPCn_m = 1):

Input/Output of ALT-OUT1 to ALT-IN16 is directly selected by the alternative function.

Table 2.9 Outline of Alternative Mode Selection (PM_{Cn}.PM_{Cn_m} = 1)

Bit					Function
PFCEAE*2	PFAE*2	PFCE*2	PFC*2	PM*1	
0	0	0	0	0	Alternative output mode 1 (ALT-OUT1)
				1	Alternative input mode 1 (ALT-IN1)
			1	0	Alternative output mode 2 (ALT-OUT2)
				1	Alternative input mode 2 (ALT-IN2)
		1	0	0	Alternative output mode 3 (ALT-OUT3)
				1	Alternative input mode 3 (ALT-IN3)
			1	0	Alternative output mode 4 (ALT-OUT4)
				1	Alternative input mode 4 (ALT-IN4)
	1	0	0	0	Alternative output mode 5 (ALT-OUT5)
				1	Alternative input mode 5 (ALT-IN5)
			1	0	Alternative output mode 6 (ALT-OUT6)
				1	Alternative input mode 6 (ALT-IN6)
		1	0	0	Alternative output mode 7 (ALT-OUT7)
				1	Alternative input mode 7 (ALT-IN7)
			1	0	Alternative output mode 8 (ALT-OUT8)
				1	Alternative input mode 8 (ALT-IN8)
1	0	0	0	0	Alternative output mode 9 (ALT-OUT9)
				1	Alternative input mode 9 (ALT-IN9)
			1	0	Alternative output mode 10 (ALT-OUT10)
				1	Alternative input mode 10 (ALT-IN10)
		1	0	0	Alternative output mode 11 (ALT-OUT11)
				1	Alternative input mode 11 (ALT-IN11)
			1	0	Alternative output mode 12 (ALT-OUT12)
				1	Alternative input mode 12 (ALT-IN12)
	1	0	0	0	Alternative output mode 13 (ALT-OUT13)
				1	Alternative input mode 13 (ALT-IN13)
			1	0	Alternative output mode 14 (ALT-OUT14)
				1	Alternative input mode 14 (ALT-IN14)
		1	0	0	Alternative output mode 15 (ALT-OUT15)
				1	Alternative input mode 15 (ALT-IN15)
			1	0	Alternative output mode 16 (ALT-OUT16)
				1	Alternative input mode 16 (ALT-IN16)

Note 1. When PIP_{Cn}.PIP_{Cn_m} = 1, the I/O direction is directly controlled by the peripheral (alternative) function and PM is ignored

Note 2. When the bit is a reserved bit, it is defined as 0 in the above table.

2.2.1.3 Pin Data Input/Output

(1) Output Data

In port mode ($PMCn.PMCn_m = 0$), the value of $Pn.Pn_m$ is output from the Pn_m pin.

(2) Input Data

Reading the $PPRn$ register returns either the value of the Pn_m pin, the associated bit of port register $Pn.Pn_m$, or the data output by an alternative function.

The source of the data read via $PPRn$ depends on the pin mode and the setting of several control bits as shown in the following table.

Table 2.10 Mode Description and $PPRn_m$ Read Values for Each Mode

PMC n_m	PM n_m	Bit				Mode	PPR Read Value			
		PIBC n_m	PIPC n_m	PODCE n_m	PODC n_m					
0	1	0	X	X	X	Port input, input buffer disabled	$Pn.Pn_m$ bit			
		1		X	X	Port input, input buffer enabled	Pn_m pin			
	0	X		0	0	Port push-pull output	$Pn.Pn_m$ bit*1			
				1	1	Port N-ch open drain output				
				1	0	Port push-pull output				
				1	1	Port P-ch open drain output				
1	1	X	0	X	X	S/W I/O control alternative input	Pn_m pin			
				0	0	0	S/W I/O control alternative push-pull output	Alternative function internal output signal*1		
					1	1	S/W I/O control alternative N-ch open drain output			
					1	0	S/W I/O control alternative push-pull output			
	1		1	S/W I/O control alternative P-ch open drain output						
	X		1	1	1	0	0	Direct I/O control alternative push-pull enable	I/O port in alternative mode • Input: Pn_m pin • Output: Alternative function Internal output signal*1	
						1	1	Direct I/O control alternative N-ch open drain enable		
						1	0	0		Direct I/O control alternative push-pull enable
							1	1		Direct I/O control alternative P-ch open drain enable

Note 1. When $PBDCn_m = 1$, the Pn_m pin level is read via the $PPRn_m$ bit.

- $PMCn.PMCn_m$

This bit selects either port mode ($PMCn_m = 0$) or alternative function mode ($PMCn_m = 1$).

- $PMn.PMn_m$

This bit selects input ($PMn_m = 1$) or output ($PMn_m = 0$) in port mode ($PMCn_m = 0$) and S/W I/O control alternative mode ($PMCn_m = 1$, $PIPCn_m = 0$).

- PIBCn.PIBCn_m

This bit disables (PIBCn_m = 0) or enables (PIBCn_m = 1) the input buffer in input port mode (PMcn_m = 0 and PMn_m = 1). When the input buffer is disabled, PPRn_m reads the Pn.Pn_m bit; otherwise the Pn_m pin level is returned.

- PIPcn.PIPcn_m

This bit selects either the S/W or direct I/O control alternative mode.

- PBDCn.PBDCn_m

Setting this bit to 1 in output mode enables bidirectional mode for the pin. In bidirectional mode, the level on pin Pn_m can be read from PPPn.PPRn_m.

- PODCn.PODCn_m, PODCEn.PODCEn_m

This bit selects either push-pull output (PODCn_m = 0), N-ch open drain output (PODCn_m = 1, PODCEn_m = 0) or P-ch open drain output (PODCn_m = 1, PODCEn_m = 1).

(3) Writing to the Pn Register

The data to be output in port Pn_m in port mode (PMcn.PMCn_m = 0) is held in port register Pn.

Pn data can be rewritten in the following two different ways:

- Direct write to the Pn register

New data can be directly written to the Pn register.

- Indirect operation on the Pn bit (SET/RESET/NOT)

An indirect operation on Pn is possible by using the following two registers:

- Port set/reset register: PSRn

When PSRn.PSRn_(m + 16) = 1, the value of PSRn.PSRn_m determines the value of Pn.Pn_m. Thus Pn_m can be set/reset without a direct write to Pn.

- Port NOT register: PNOTn

Setting PNOTn.PNOTn_m = 1 inverts the Pn.Pn_m bit without a direct write to Pn.

The indirect Pn SET/RESET/NOT operation provides access to single bits (not limited to one bit) of the Pn register while leaving all other Pn bits untouched.

2.3 Port Types

2.3.1 Pin Configuration

The following figure shows the overall configuration of the pins.

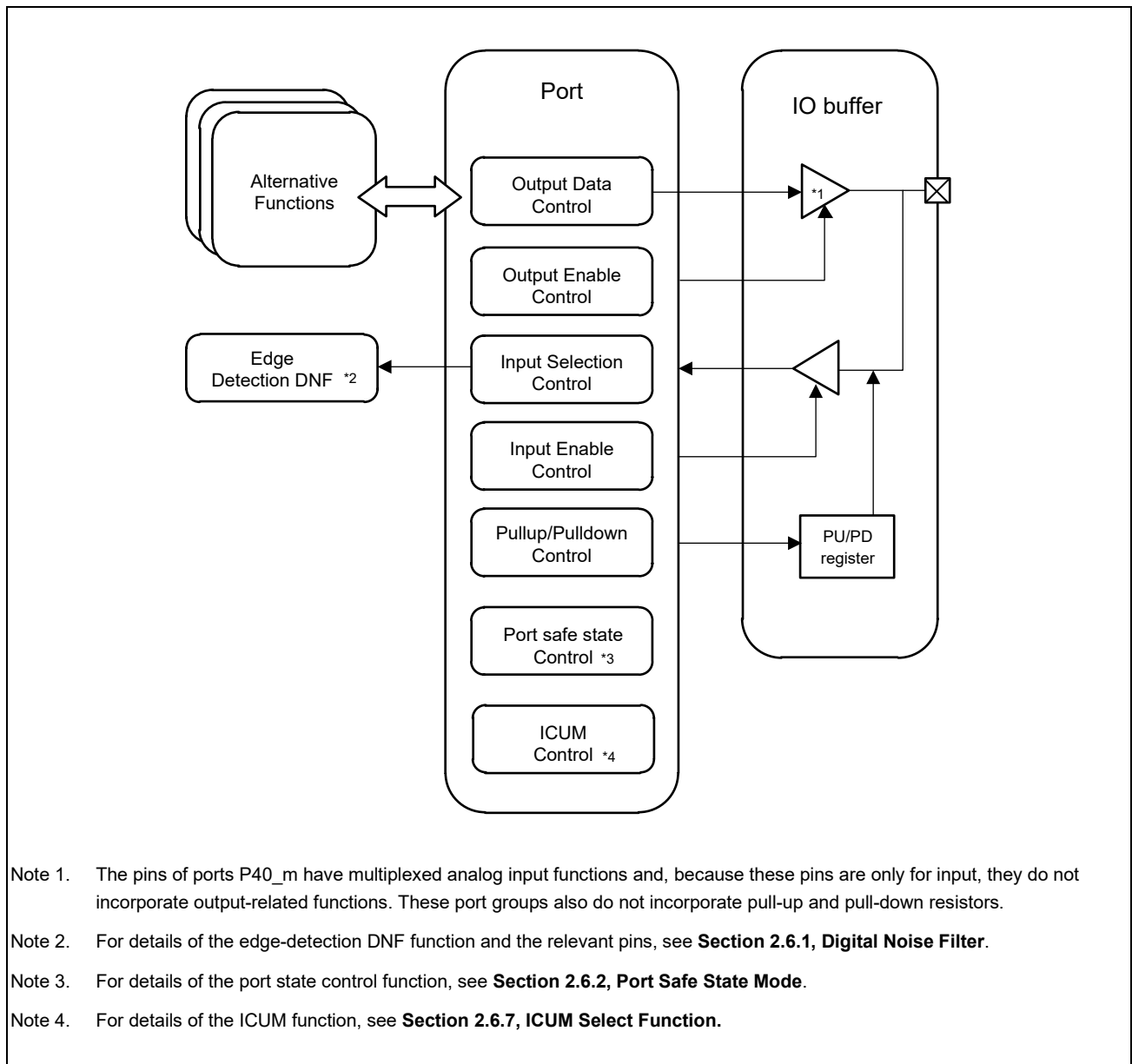


Figure 2.1 Block Diagram of Pin Configuration

2.3.2 Port Control Logic

The following figure shows the logical circuitry of the port control functions. The diagram is only a logical reference and does not show the real circuitry.

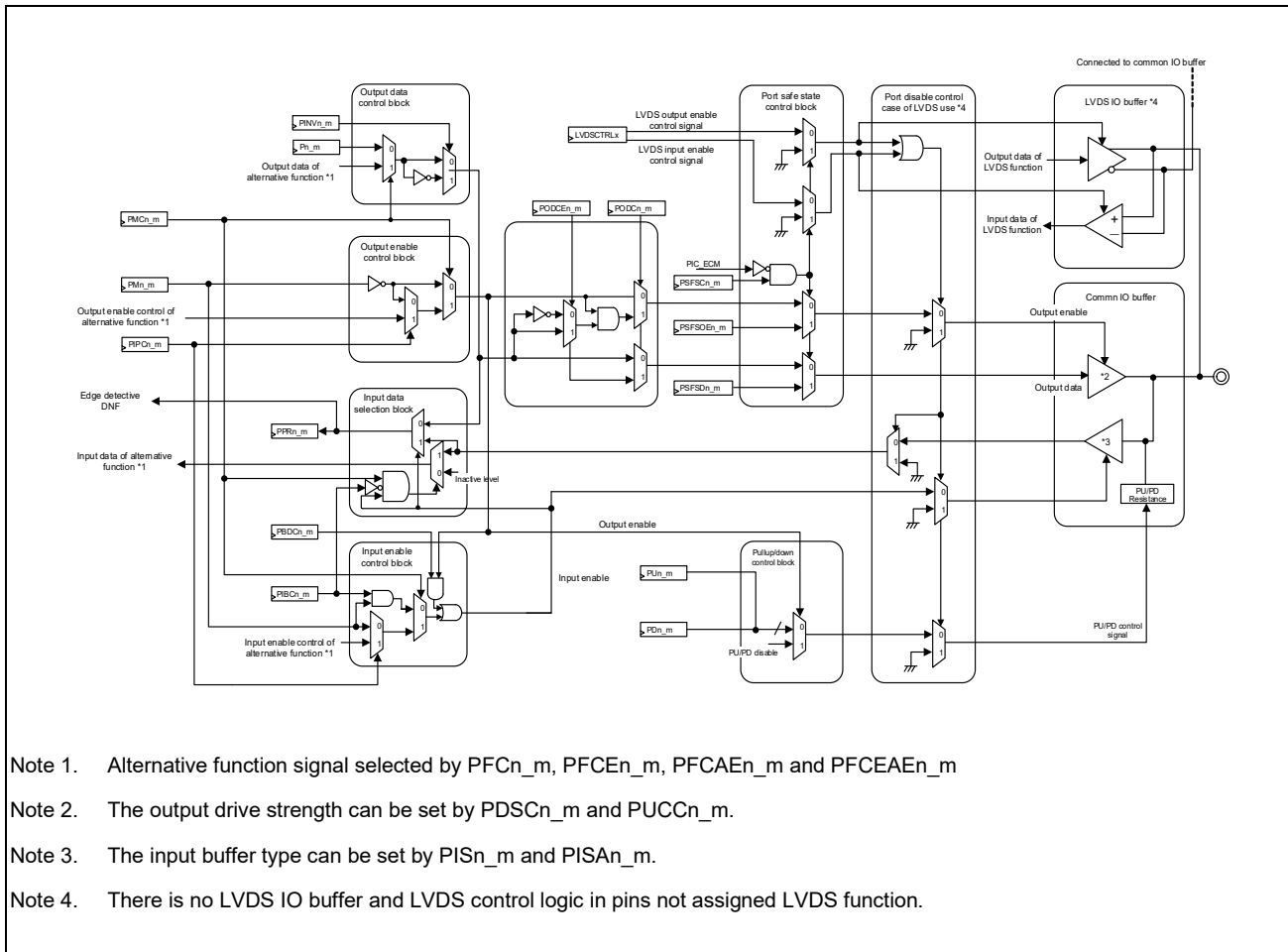


Figure 2.2 Block Diagram of Port Control Logic

2.4 Registers

2.4.1 List of Registers

The PORT registers are listed in the following table.

For details of the PORT base addresses, see **Section 2.1.2, Register Base Addresses**.

Table 2.11 List of Registers (1/2)

Module Name	Register Name	Symbol	Register Access Unit	Address	Access	Access Protection*1	
						Excluding P21, 27	P21, 27 only*2,*3
PORT0	Port register	Pn	Port Group	<PORT0_base> + 0000 _H + n × 40 _H	16	—	PWE
	Port set/reset register	PSRn	Port Group	<PORT0_base> + 0004 _H + n × 40 _H	32	—	PWE
	Port NOT register	PNOTn	Port Group	<PORT0_base> + 0008 _H + n × 40 _H	16	—	PWE
	Port pin read register	PPRn	Port Group	<PORT0_base> + 000C _H + n × 40 _H	16	—	—
	Port mode register	PMn	Port Group	<PORT0_base> + 0010 _H + n × 40 _H	16	—	PWE
	Port mode control register	PMcN	Port Group	<PORT0_base> + 0014 _H + n × 40 _H	16	—	PWE
	Port function control register	PFcN	Port Group	<PORT0_base> + 0018 _H + n × 40 _H	16	—	PWE
	Port function control expansion register	PFCEn	Port Group	<PORT0_base> + 001C _H + n × 40 _H	16	—	PWE
	Port mode set/reset register	PMSRn	Port Group	<PORT0_base> + 0020 _H + n × 40 _H	32	—	PWE
	Port mode control set/reset register	PMCSRn	Port Group	<PORT0_base> + 0024 _H + n × 40 _H	32	—	PWE
	Port function control additional expansion register	PFCAEn	Port Group	<PORT0_base> + 0028 _H + n × 40 _H	16	—	PWE
	Port function control extra additional expansion register	PFCEAEn	Port Group	<PORT0_base> + 002C _H + n × 40 _H	16	—	PWE
	Port output level inversion register	PINVn	Port Group	<PORT0_base> + 0030 _H + n × 40 _H	16	PWE	PWE
	Port safe state control register	PSFSCn	Port Group	<PORT0_base> + 0034 _H + n × 40 _H	16	PWE	PWE
	Port safe state data register	PSFSDn	Port Group	<PORT0_base> + 0038 _H + n × 40 _H	16	PWE	PWE
Port safe state output enable register	PSFSOEn	Port Group	<PORT0_base> + 003C _H + n × 40 _H	16	PWE	PWE	
PORT1	Port input buffer control register	PIBCn	Port Group	<PORT1_base> + 4000 _H + n × 40 _H	16	—	PWE
	Port bidirectional control register	PBDCn	Port Group	<PORT1_base> + 4004 _H + n × 40 _H	16	—	PWE
	Port IP control register	PIPCn	Port Group	<PORT1_base> + 4008 _H + n × 40 _H	16	—	PWE
	Pull-up option register	PU _n	Port Group	<PORT1_base> + 400C _H + n × 40 _H	16	—	PWE
	Pull-down option register	PD _n	Port Group	<PORT1_base> + 4010 _H + n × 40 _H	16	—	PWE
	Port open drain control register	PODCn	Port Group	<PORT1_base> + 4014 _H + n × 40 _H	16	PWE	PWE
	Port open drain control expansion register	PODCEn	Port Group	<PORT1_base> + 4038 _H + n × 40 _H	16	PWE	PWE
	Port drive strength control register	PDSCn	Port Group	<PORT1_base> + 4018 _H + n × 40 _H	16	PWE	PWE
	Port Input buffer selection register	PISn	Port Group	<PORT1_base> + 401C _H + n × 40 _H	16	—	PWE
	Port input buffer selection advanced register	PISAn	Port Group	<PORT1_base> + 4024 _H + n × 40 _H	16	—	PWE
	Port universal control register	PUCcN	Port Group	<PORT1_base> + 4028 _H + n × 40 _H	16	PWE	PWE
PORT0	Port control register	PCRn_m	Pin	<PORT0_base> + 2000 _H + n × 40 _H + m × 4 _H	32	PWE	PWE

Table 2.11 List of Registers (2/2)

Module Name	Register Name	Symbol	Register Access Unit	Address	Access	Access Protection*1		
						Excluding P21, 27	P21, 27 only*2,*3	
JPORT0	Port register	JPn	Port Group	<JPORT0_base> + 0000 _H + n × 40 _H	8	—	—	
	Port set/reset register	JPSRn	Port Group	<JPORT0_base> + 0004 _H + n × 40 _H	32	—	—	
	Port NOT register	JPNOTn	Port Group	<JPORT0_base> + 0008 _H + n × 40 _H	8	—	—	
	Port pin read register	JPPRn	Port Group	<JPORT0_base> + 000C _H + n × 40 _H	8	—	—	
	Port mode register	JPMn	Port Group	<JPORT0_base> + 0010 _H + n × 40 _H	8	—	—	
	Port mode control register	JPMCn	Port Group	<JPORT0_base> + 0014 _H + n × 40 _H	8	—	—	
	Port mode set/reset register	JPMsRn	Port Group	<JPORT0_base> + 0020 _H + n × 40 _H	32	—	—	
	Port mode control set/reset register	JPMCSRn	Port Group	<JPORT0_base> + 0024 _H + n × 40 _H	32	—	—	
	Port output level inversion register	JPINVn	Port Group	<JPORT0_base> + 0030 _H + n × 40 _H	8	—	—	
JPORT1	Port input buffer control register	JPIBCn	Port Group	<JPORT1_base> + 4000 _H + n × 40 _H	8	—	—	
	Port bidirectional control register	JPBDCn	Port Group	<JPORT1_base> + 4004 _H + n × 40 _H	8	—	—	
	Port IP control register	JPIPCn	Port Group	<JPORT1_base> + 4008 _H + n × 40 _H	8	—	—	
	Pull-up option register	JPUn	Port Group	<JPORT1_base> + 400C _H + n × 40 _H	8	—	—	
	Pull-down option register	JPDn	Port Group	<JPORT1_base> + 4010 _H + n × 40 _H	8	—	—	
	Port open drain control register	JPODCn	Port Group	<JPORT1_base> + 4014 _H + n × 40 _H	8	—	—	
	Port open drain control expansion register	JPODCEn	Port Group	<JPORT1_base> + 4038 _H + n × 40 _H	8	—	—	
	Port drive strength control register	JPDSCn	Port Group	<JPORT1_base> + 4018 _H + n × 40 _H	8	—	—	
	Port Input buffer selection register	JPISn	Port Group	<JPORT1_base> + 401C _H + n × 40 _H	8	—	—	
	Port input buffer selection advanced register	JPISAn	Port Group	<JPORT1_base> + 4024 _H + n × 40 _H	8	—	—	
	Port universal control register	JPUCn	Port Group	<JPORT1_base> + 4028 _H + n × 40 _H	8	—	—	
	JPORT0	Port control register	JPCRn_m	Pin	<JPORT0_base> + 2000 _H + n × 40 _H + m × 4 _H	32	—	—
	PORT0	Port Keycode Protection register	PKCPROT	—	<PORT0_base> + 2F40 _H	32	—	—
Port Write enable register		PWE	—	<PORT0_base> + 2F44 _H	32	PKCPROT	PKCPROT	
LVDS control A register		LVDSCTRLA	—	<PORT0_base> + 2F50 _H	32	—	—	
LVDS control B register		LVDSCTRLB	—	<PORT0_base> + 2F54 _H	32	—	—	
LVDS control C register		LVDSCTRLC	—	<PORT0_base> + 2F58 _H	32	—	—	

Note: n: Port number, m: Port bit number

Note 1. Write protect only.

Note 2. The P21 port group register is protected from all write accesses to prevent accidental register access. This is because in addition to their port function, the P21_* pins are multiplexed with the RHSIF function; however, these functions do not use the same power domain. This means that I/O of the P21 port group must be disabled when RHSIF is used.

Note 3. P27_0 is assigned exclusively to the RES_OUT function. The P27 port group register is therefore protected from all write accesses.

2.4.2 Port Group Unit Register Explanation

2.4.2.1 Configuration of PORT

(1) PMC/JPMC — Port Mode Control Register Group

(a) PMCn/JPMC0 — Port Mode Control Register

This register specifies whether the individual pins of port group n are in port mode or in alternative mode.

CAUTION

PMC21 and PMC27 are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMCn_15	PMCn_14	PMCn_13	PMCn_12	PMCn_11	PMCn_10	PMCn_9	PMCn_8	PMCn_7	PMCn_6	PMCn_5	PMCn_4	PMCn_3	PMCn_2	PMCn_1	PMCn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.12 PMCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PMCn_[15:0]	These pins specify the operation mode of the corresponding pin: 0: Port mode 1: Alternative mode

NOTE

The control bits of the JTAG Port Mode Control Register (JPMC0) are JPMC0_[7:0].

(b) PMCSRn/JPMCSR0 — Port Mode Control Set/Reset Register

This register provides an alternative method for writing data to a bit in the PMCn register.

The 16 higher bits of PMCSRn specify whether the data in the 16 lower bits of PMCSRn are written to PMCn.PMCn_m.

Even when pins being used by multiple programs belong to the same port group, the PMCSRn register allows masking of the unused bits when overwriting so that independent setting of the corresponding bits by each program is possible.

CAUTION

PMCSR21 and PMCSR27 are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMCSRn ₃₁	PMCSRn ₃₀	PMCSRn ₂₉	PMCSRn ₂₈	PMCSRn ₂₇	PMCSRn ₂₆	PMCSRn ₂₅	PMCSRn ₂₄	PMCSRn ₂₃	PMCSRn ₂₂	PMCSRn ₂₁	PMCSRn ₂₀	PMCSRn ₁₉	PMCSRn ₁₈	PMCSRn ₁₇	PMCSRn ₁₆
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMCSRn ₁₅	PMCSRn ₁₄	PMCSRn ₁₃	PMCSRn ₁₂	PMCSRn ₁₁	PMCSRn ₁₀	PMCSRn ₉	PMCSRn ₈	PMCSRn ₇	PMCSRn ₆	PMCSRn ₅	PMCSRn ₄	PMCSRn ₃	PMCSRn ₂	PMCSRn ₁	PMCSRn ₀
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.13 PMCSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PMCSRn _[31:16]	These bits specify whether the value of the corresponding lower bit of PMCSRn_m is written to PMCn_m: 0: PMCn_m does not depend on PMCSRn_m. 1: The value of PMCn_m is the same as that of PMCSRn_m. Example: When PMCSRn.PMCSRn ₃₁ = 1, the value of bit PMCSRn.PMCSRn ₁₅ is written to bit PMCn.PMCn ₁₅ . When read, 0000 _H is always returned.
15 to 0	PMCSRn _[15:0]	These bits specify the PMCn_m value when the corresponding upper bit PMCSRn _(m+16) is 1: 0: PMCn_m = 0 1: PMCn_m = 1 When read, the PMCn register value is returned.

NOTE

The control bits of the JTAG Port Mode Control Set/Reset Register (JPMCSR0) are JPMCSR0_[31:0].

(2) PIPCN/JIPIC0 — Port IP Control Register

This register specifies whether the I/O direction of the Pn_m pin is controlled by port mode register PMn.PMn_m or by an alternative function.

When the Pn_m pin is operated in alternative mode (PMn.PMn_m = 1) and the alternative function directly controls the I/O direction of Pn_m. In this case, PIPCN.PIPCN_m must be set to 1 as well. This hands over I/O control to the alternative function and overrules the PMn.PMn_m setting.

CAUTION

PIPC21 and PIPC27 are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIPCn_15	PIPCn_14	PIPCn_13	PIPCn_12	PIPCn_11	PIPCn_10	PIPCn_9	PIPCn_8	PIPCn_7	PIPCn_6	PIPCn_5	PIPCn_4	PIPCn_3	PIPCn_2	PIPCn_1	PIPCn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.14 PIPCN Register Contents

Bit Position	Bit Name	Function
15 to 0	PIPCn_[15:0]	These bits specify the I/O control mode: 0: I/O mode is selected by PMn.PMn_m (S/W I/O control alternative mode). 1: I/O mode is selected by peripheral function (direct I/O control alternative mode).

NOTE

The control bits of the JTAG Port IP Control Register (JIPIC0) are JIPIC0_[7:0].

(3) PM/JPM — Port Mode Register Group

(a) PMn/JPM0 — Port Mode Register

The PMn register specifies whether the individual pins of port group n are in input mode or in output mode.

CAUTION

PM21 and PM27 are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMn_15	PMn_14	PMn_13	PMn_12	PMn_11	PMn_10	PMn_9	PMn_8	PMn_7	PMn_6	PMn_5	PMn_4	PMn_3	PMn_2	PMn_1	PMn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.15 PMn Register Contents

Bit Position	Bit Name	Function
15 to 0	PMn_[15:0]	These bits specify input/output mode of the corresponding pin: 0: Output mode (output enabled) 1: Input mode (output disabled)

NOTE

- To use a pin in input port mode (PMn.PMn_m = 0 and PMn.PMn_m = 1), the input buffer must be enabled (PIBCn.PIBCn_m = 1).
- PMn_m specifies the I/O direction in port mode (PMn.PMn_m = 0) and alternative mode (PMn.PMn_m = 1) because PIPCN.PIPCN_m = 0 after reset.
- The control bits of the JTAG Port Mode Register (JPM0) are JPM0_[7:0].

(b) PMSRn/JPMSR0 — Port Mode Set/Reset Register

This register provides an alternative method for setting a bit in the PMn register.

The 16 higher bits of PMSRn specify whether the data in the 16 lower bits of PMSRn are written to PMn.PMn_m.

Even when pins being used by multiple programs belong to the same port group, the PMSRn register allows masking of the unused bits when overwriting so that independent setting of the corresponding bits by each program is possible.

CAUTION

PMSR21 and PMSR27 are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMSRn_31	PMSRn_30	PMSRn_29	PMSRn_28	PMSRn_27	PMSRn_26	PMSRn_25	PMSRn_24	PMSRn_23	PMSRn_22	PMSRn_21	PMSRn_20	PMSRn_19	PMSRn_18	PMSRn_17	PMSRn_16
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMSRn_15	PMSRn_14	PMSRn_13	PMSRn_12	PMSRn_11	PMSRn_10	PMSRn_9	PMSRn_8	PMSRn_7	PMSRn_6	PMSRn_5	PMSRn_4	PMSRn_3	PMSRn_2	PMSRn_1	PMSRn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.16 PMSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PMSRn_[31:16]	These bits specify whether the value of the corresponding lower bit of PMSRn_m is written to PMn_m: 0: PMn_m does not depend on PMSRn_m. 1: The value of PMn_m is the same as that of PMSRn_m. Example: When PMSRn.PMSRn_31 = 1, the value of bit PMSRn.PMSRn_15 is written to bit PMn.PMn_15. When read, 0000 _H is always returned.
15 to 0	PMSRn_[15:0]	These bits specify the PMn_m value when the corresponding upper bit PMSRn_(m+16) is 1: 0: PMn_m = 0 1: PMn_m = 1 When read, the PMn register value is returned.

NOTE

The control bits of the JTAG Port Mode Set/Reset Register (JPMSR0) are JPMSR0_[31:0].

(4) PIBCn/JPIBC0 — Port Input Buffer Control Register

When a pin is being used in input port mode ($PMCn.PMCn_m = 0$ and $PMn.PMn_m = 1$), this register enables or disables the input buffer. However, when the pin is used as an input pin in S/W I/O control alternative mode ($PMCn.PMCn_m = 1$ and $PIPCn.PIPCn_m = 0$) or direct I/O control alternative mode ($PMCn.PMCn_m = 1$ and $PIPCn.PIPCn_m = 1$), set $PIBCn.PIBCn_m = 0$.

When pins are in bidirectional mode ($PBDCn.PBDCn_m = 1$), the shared output level loop-back function and pin output level-read function can be selected by setting $PIBCn.PIBCn_m$. Refer to **Section 2.4.2.2(1), PBDCn/JPBDC0 — Port Bidirectional Control Register**.

CAUTION

PIBC21 and PIBC27 are write-protected by PWE.
See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIBCn_15	PIBCn_14	PIBCn_13	PIBCn_12	PIBCn_11	PIBCn_10	PIBCn_9	PIBCn_8	PIBCn_7	PIBCn_6	PIBCn_5	PIBCn_4	PIBCn_3	PIBCn_2	PIBCn_1	PIBCn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.17 PIBCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PIBCn_[15:0]	These bits are used to enable and disable the input buffer in input port mode. 0: Input buffer is disabled. 1: Input buffer is enabled.

NOTES

- When the input buffer is disabled, a shoot-through current does not flow even if the pin level is in Hi-Z state. Thus the pin does not need to be fixed to a high or low level externally.
- When the input buffer is enabled in alternative mode ($PMCn.PMCn_m = 1$, $PMn.PMn_m = 1$), there is a possibility that a shoot-through current may flow even if the pin level is Hi-Z state. Then take appropriate measures, such as pull-up/pull-down.
- The control bits of the JTAG Port Input Buffer Control Register (JPIBC0) are JPIBC0_[7:0].

(5) PFC — Port Function Control Register Group

(a) PFCn — Port Function Control Register

This register, together with the PFCEn, PFCAEn and PFCEAEn registers, specifies an alternative function of the pins.

Some alternative functions directly control input/output of pin Pn_m. For such alternative functions PIPcN.PIPCn_m must be set to 1 as well.

For other alternative functions, input/output must be specified by PMn.PMn_m.

CAUTION

PFC21 and PFC27 are protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCn_15	PFCn_14	PFCn_13	PFCn_12	PFCn_11	PFCn_10	PFCn_9	PFCn_8	PFCn_7	PFCn_6	PFCn_5	PFCn_4	PFCn_3	PFCn_2	PFCn_1	PFCn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.18 PFCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCn_[15:0]	These bits specify an alternative function of a pin. For detailed information, refer to Table 2.9, Outline of Alternative Mode Selection (PMCn.PMCn_m = 1) .

(b) PFCEn — Port Function Control Expansion Register

This register, together with the PFCn, PFCAEn and PFCEAEn registers, specifies an alternative function of the pins.

Some alternative functions directly control input/output of pin Pn_m. For such alternative functions PIPCN.PIPCN_m must be set to 1 as well.

For other alternative functions, input/output must be specified by PMn.PMn_m.

CAUTION

PFCE21 and PFCE27 are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCEn _15	PFCEn _14	PFCEn _13	PFCEn _12	PFCEn _11	PFCEn _10	PFCEn _9	PFCEn _8	PFCEn _7	PFCEn _6	PFCEn _5	PFCEn _4	PFCEn _3	PFCEn _2	PFCEn _1	PFCEn _0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.19 PFCEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCEn_[15:0]	These bits specify an alternative function of a pin. For detailed information, refer to Table 2.9, Outline of Alternative Mode Selection (PMCN.PMCn_m = 1) .

(c) PFCAEn — Port Function Control Additional Expansion Register

This register, together with the PFCn, PFCEn and PFCEAEn registers, specifies an alternative function of the pins.

Some alternative functions directly control input/output of pin Pn_m. For such alternative functions PIPCN.PIPCN_m must be set to 1 as well.

For other alternative functions, input/output must be specified by PMn.PMn_m.

CAUTION

PFCAE21 and PFCAE27 are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCAE n_15	PFCAE n_14	PFCAE n_13	PFCAE n_12	PFCAE n_11	PFCAE n_10	PFCAE n_9	PFCAE n_8	PFCAE n_7	PFCAE n_6	PFCAE n_5	PFCAE n_4	PFCAE n_3	PFCAE n_2	PFCAE n_1	PFCAE n_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.20 PFCAEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCAEn_[15:0]	These bits specify an alternative function of a pin. For detailed information, refer to Table 2.9, Outline of Alternative Mode Selection (PMCN.PMCN_m = 1) .

(d) PFCEAEn — Port Function Control Extra Additional Expansion Register

This register, together with the PFCn, PFCEn, and PFCAEn registers, specifies an alternative function of the pins.

Some alternative functions directly control input/output of pin Pn_m. For such alternative functions PIPCN.PIPCN_m must be set to 1 as well.

For other alternative functions, input/output must be specified by PMn.PMn_m.

CAUTION

PFCEAE21 and PFCEAE27 are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCEA En_15	PFCEA En_14	PFCEA En_13	PFCEA En_12	PFCEA En_11	PFCEA En_10	PFCEA En_9	PFCEA En_8	PFCEA En_7	PFCEA En_6	PFCEA En_5	PFCEA En_4	PFCEA En_3	PFCEA En_2	PFCEA En_1	PFCEA En_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.21 PFCEAEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PSCEAEn_[15:0]	These bits specify an alternative function of a pin. For detailed information, refer to Table 2.9, Outline of Alternative Mode Selection (PMCN.PMCN_m = 1) .

2.4.2.2 Configuration of Pin Data Input/Output

(1) PBDCn/JPBDC0 — Port Bidirectional Control Register

This register enables the input buffer when a pin is used in output mode, and permits bidirectional mode. The Pn_m pin level is read via PPRn.PPRn_m in bidirectional mode.

- Alternative output level loopback function

When the Pn_m pin is used as the alternative output function, the actual pin output level based on the alternative output function can be looped back to the alternative input side by setting PBDCn.PBDCn_m = 1 and

PIBCn.PIBCn_m = 0. For example, the pin output level based on the first alternative function can be looped back to the same alternative input side. Also the pin output level can be read via PPRn.PPRn_m.

- Pin output level read function

When the Pn_m pin is used as the general output port function or the alternative output function, the actual pin output level can be read via PPRn.PPRn_m by setting PBDCn.PBDCn_m = 1 and PIBCn.PIBCn_m = 1. Under this setting, the pin output level will never be looped back to the alternative input side even in alternative output mode.

CAUTION

PBDC21 and PBDC27 are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PBDCn _15	PBDCn _14	PBDCn _13	PBDCn _12	PBDCn _11	PBDCn _10	PBDCn _9	PBDCn _8	PBDCn _7	PBDCn _6	PBDCn _5	PBDCn _4	PBDCn _3	PBDCn _2	PBDCn _1	PBDCn _0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.22 PBDCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PBDCn_[15:0]	These bits enable/disable bidirectional mode of the corresponding pin: 0: Bidirectional mode is disabled. 1: Bidirectional mode is enabled.

NOTE

The control bits of the JTAG Port Bidirectional Control Register (JPBC0) are JPBC0_[7:0].

(2) PPRn/JPPR0 — Port Pin Read Register

This register reflects an actual Pn_m pin level, a Pn.Pn_m bit value, or an output level of the alternative function. The value to be read depends on various control settings as described in **Table 2.10, Mode Description and PPRn_m Read Values for Each Mode**.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PPRn_15	PPRn_14	PPRn_13	PPRn_12	PPRn_11	PPRn_10	PPRn_9	PPRn_8	PPRn_7	PPRn_6	PPRn_5	PPRn_4	PPRn_3	PPRn_2	PPRn_1	PPRn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.23 PPRn Register Contents

Bit Position	Bit Name	Function
15 to 0	PPRn_[15:0]	These bits indicate the Pn_m pin level, Pn.Pn_m value, or alternative function output level.

NOTE

The control bits of the JTAG Port Pin Read Register (JPPR0) are JPPR0_[7:0].

(3) P/JP — Port Register Group

(a) Pn/JP0 — Port Register

This register sets and holds the Pn.Pn_m data to be output via the related Pn_m port in output port mode (PMcn.PMCn_m = 0 and PMn.PMn_m = 0).

CAUTION

P21 and P27 are write-protected by PWE.
See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Pn_15	Pn_14	Pn_13	Pn_12	Pn_11	Pn_10	Pn_9	Pn_8	Pn_7	Pn_6	Pn_5	Pn_4	Pn_3	Pn_2	Pn_1	Pn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.24 Pn Register Contents

Bit Position	Bit Name	Function
15 to 0	Pn_[15:0]	These bits set the output level of pin m (m = 0 to 15): 0: Low level output 1: High level output

NOTES

- Bits of this register can be manipulated by various means. Refer to the subsection, “Writing to the Pn Register” in **Section 2.2.1.3, Pin Data Input/Output** for details.
- The control bits of the JTAG Port Register (JP0) are JP0_[7:0].

(b) PSRn/JPSR0 — Port Set/Reset Register

This register provides an alternative method for setting a bit in the Pn register.

The 16 higher bits of PSRn specify whether the data in the 16 lower bits of PSRn are written to Pn.Pn_m.

Even when pins being used by multiple programs belong to the same port group, the PSRn register allows masking of the unused bits when overwriting so that independent setting of the corresponding bits by each program is possible.

CAUTION

PSR21 and PSR27 are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PSRn_31	PSRn_30	PSRn_29	PSRn_28	PSRn_27	PSRn_26	PSRn_25	PSRn_24	PSRn_23	PSRn_22	PSRn_21	PSRn_20	PSRn_19	PSRn_18	PSRn_17	PSRn_16
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSRn_15	PSRn_14	PSRn_13	PSRn_12	PSRn_11	PSRn_10	PSRn_9	PSRn_8	PSRn_7	PSRn_6	PSRn_5	PSRn_4	PSRn_3	PSRn_2	PSRn_1	PSRn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.25 PSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PSRn_[31:16]	These bits specify whether the value of the corresponding lower bit of PSRn_m is written to Pn_m: 0: Pn_m does not depend on PSRn_m. 1: The value of Pn_m is the same as that of PSRn_m. Example: When PSRn.PSRn_31 = 1, the value of bit PSRn.PSRn_15 is written to bit Pn.Pn_15 and output. When read, 0000 _H is always returned.
15 to 0	PSRn_[15:0]	These bits specify the Pn_m value when the corresponding upper bit PSRn_(m+16) is 1: 0: Pn_m = 0 1: Pn_m = 1 When read, the Pn register value is returned.

NOTE

The control bits of the JTAG Port Set/Reset Register (JPSR0) are JPSR0_[31:0].

(4) PNOTn/JPNOT0 — Port NOT Register

This register enables inverting the Pn_m bit of the port register without directly writing to Pn.

CAUTION

PNOT21 and PNOT27 are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PNOTn _15	PNOTn _14	PNOTn _13	PNOTn _12	PNOTn _11	PNOTn _10	PNOTn _9	PNOTn _8	PNOTn _7	PNOTn _6	PNOTn _5	PNOTn _4	PNOTn _3	PNOTn _2	PNOTn _1	PNOTn _0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.26 PNOTn Register Contents

Bit Position	Bit Name	Function
15 to 0	PNOTn_[15:0]	These bits specify if Pn.Pn_m is inverted or not: 0: Pn.Pn_m is not inverted (Pn_m → Pn_m). 1: Pn.Pn_m is inverted ($\overline{\text{Pn_m}}$ → Pn_m)

NOTE

The control bits of the JTAG Port NOT Register (JPNOT0) are JPNOT0_[7:0].

(5) PINVn/JPINV0 — Port Output Level Inversion Register

This register enables inverting the output level from a pin. This is valid when the pin is in output mode regardless of port output mode or alternative output mode. It is not valid when Port safe state or the ICUM function are active.

CAUTION

These registers are write-protected by PWE.
See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PINVn_15	PINVn_14	PINVn_13	PINVn_12	PINVn_11	PINVn_10	PINVn_9	PINVn_8	PINVn_7	PINVn_6	PINVn_5	PINVn_4	PINVn_3	PINVn_2	PINVn_1	PINVn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.27 PINVn Register Contents

Bit Position	Bit Name	Function
15 to 0	PINVn_[15:0]	These bits specify whether the output level from a pin is inverted or not. 0: Pin output level is not inverted. 1: Pin output level is inverted.

NOTE

The control bits of the JTAG Port Output Level Inversion Register (JPINV0) are JPINV0_[7:0].

(6) PSFS — Port Safe State Register Group

(a) PSFSCn — Port Safe State Control Register

All Port Groups have Port Safe State Function implemented. Each of these Port Groups has its own Port safe state enable bit that is located in the PSFSCn_m register.

CAUTION

These registers are write-protected by PWE.
See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSFSC n_15	PSFSC n_14	PSFSC n_13	PSFSC n_12	PSFSC n_11	PSFSC n_10	PSFSC n_9	PSFSC n_8	PSFSC n_7	PSFSC n_6	PSFSC n_5	PSFSC n_4	PSFSC n_3	PSFSC n_2	PSFSC n_1	PSFSC n_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.28 PSFSC Register Contents

Bit Position	Bit Name	Function
15 to 0	PSFSCn_[15:0]	These bits enable/disable Port Safe State Function of pin m (m = 0 to 15).
	PSFSCn_m	Description
	0	Port Safe State Function is disabled Output control still remains. (Port mode or alternative control mode.)
	1	Port Safe State Function is enabled. Output control is changed to Port safe state mode.

(b) PSFSDn — Port Safe State Data Register

When the Port safe state trigger signal becomes active, the port register (Pn_m) is disconnected and switched to the Port Safe State Data register (PSFSDn_m), and the Port Safe State Output Enable register (PSFSOEn_m) takes over output enable control.

CAUTION

These registers are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSFSD n_15	PSFSD n_14	PSFSD n_13	PSFSD n_12	PSFSD n_11	PSFSD n_10	PSFSD n_9	PSFSD n_8	PSFSD n_7	PSFSD n_6	PSFSD n_5	PSFSD n_4	PSFSD n_3	PSFSD n_2	PSFSD n_1	PSFSD n_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.29 PSFSD Register Contents

Bit Position	Bit Name	Function
15 to 0	PSFSDn_[15:0]	These bits set the output level of pin m (m = 0 to 15) in Port safe state mode.
	PSFSDn_m	Data Bus Width
	0	Low level output in Port safe state mode
	1	High level output in Port safe state mode

(c) PSFSOEn — Port Safe State Output Enable Register

When the Port safe state trigger signal becomes active, the port register (Pn_m) is disconnected and switched to the Port Safe State Data register (PSFSDn_m), and the Port Safe State Output Enable register (PSFSOEn_m) takes over output enable control.

CAUTION

These registers are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSFSO En_15	PSFSO En_14	PSFSO En_13	PSFSO En_12	PSFSO En_11	PSFSO En_10	PSFSO En_9	PSFSO En_8	PSFSO En_7	PSFSO En_6	PSFSO En_5	PSFSO En_4	PSFSO En_3	PSFSO En_2	PSFSO En_1	PSFSO En_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.30 PSFSOE Register Contents

Bit Position	Bit Name	Function
15 to 0	PSFSOEn_[15:0]	These bits set the output enable level of pin m (m = 0 to 15) in Port safe state mode.
	PSFSOEn_m	Data Bus Width
	0	Output disabled in Port safe state mode
	1	Output enabled in Port safe state mode

2.4.2.3 Configuration of Electrical Characteristic

(1) PUD/JPU0 — Pull-Up/Down Register Group

(a) PUn/JPU0 — Pull-Up Option Register

This register specifies whether an on-chip pull-up resistor is connected to an input pin.

CAUTION

PU21 and PU27 are write-protected by PWE.
See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PUn_15	PUn_14	PUn_13	PUn_12	PUn_11	PUn_10	PUn_9	PUn_8	PUn_7	PUn_6	PUn_5	PUn_4	PUn_3	PUn_2	PUn_1	PUn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.31 PUn Register Contents

Bit Position	Bit Name	Function	
15 to 0	PUn_[15:0]	These bits specify whether an on-chip pull-up resistor is connected to the corresponding pin in combination with the PDn register. For the electrical characteristics for each setting, see <i>the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware</i> .	
	PUn_m	PDn_m	Description
	0	0	No on-chip input pull-up/pull-down resistor is connected.
	0	1	On-chip pull-down resistor is connected.
	1	0	On-chip pull-up resistor is connected.
	1	1	Setting is prohibited

NOTES

- Do not set PUn.PUn_m = 1 and PDn.PDn_m = 1 to a single pin.
- The on-chip pull-up resistor has no effect when the pin is operated in output mode.
- The control bits of the JTAG Pull Up Option Register (JPU0) are JPU0_[7:0]

(b) PDn/JPD0 — Pull-Down Option Register

This register specifies whether an on-chip pull-down resistor is connected to an input pin.

CAUTION

PD21 and PD27 are write-protected by PWE.
See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDn_15	PDn_14	PDn_13	PDn_12	PDn_11	PDn_10	PDn_9	PDn_8	PDn_7	PDn_6	PDn_5	PDn_4	PDn_3	PDn_2	PDn_1	PDn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.32 PDn Register Contents

Bit Position	Bit Name	Function
15 to 0	PDn_[15:0]	These bits set pull-down resistor of input port pins in combination with the PUn register. See Table 2.31, PUn Register Contents .

NOTES

- Do not set PUn.PUn_m = 1 and PDn.PDn_m = 1 to a single pin.
- The on-chip pull-down resistor has no effect when the pin is operated in output mode.
- The control bits of the JTAG Pull Down Option Register (JPD0) are JPD0_[7:0]

(2) PODC/JPODC — Port Open Drain Control Register Group**(a) PODCn/JPODC0 — Port Open Drain Control Register**

This register selects port pin's output buffer function as push-pull type or emulated open drain type.

CAUTION

These registers are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PODCn _15	PODCn _14	PODCn _13	PODCn _12	PODCn _11	PODCn _10	PODCn _9	PODCn _8	PODCn _7	PODCn _6	PODCn _5	PODCn _4	PODCn _3	PODCn _2	PODCn _1	PODCn _0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.33 PODCn Register Contents

Bit Position	Bit Name	Function															
15 to 0	PODCn_[15:0]	These bits enable/disable N-ch or P-ch open drain function of pin m (m = 0 to 15) in combination with the PODCEn register.															
		<table border="1"> <thead> <tr> <th>PODCEn_m</th> <th>PODCn_m</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Output buffer operates as push-pull driver</td> </tr> <tr> <td>0</td> <td>1</td> <td>Output buffer operates as N-ch open drain driver</td> </tr> <tr> <td>1</td> <td>0</td> <td>Output buffer operates as push-pull driver</td> </tr> <tr> <td>1</td> <td>1</td> <td>Output buffer operates as P-ch open drain driver</td> </tr> </tbody> </table>	PODCEn_m	PODCn_m	Description	0	0	Output buffer operates as push-pull driver	0	1	Output buffer operates as N-ch open drain driver	1	0	Output buffer operates as push-pull driver	1	1	Output buffer operates as P-ch open drain driver
PODCEn_m	PODCn_m	Description															
0	0	Output buffer operates as push-pull driver															
0	1	Output buffer operates as N-ch open drain driver															
1	0	Output buffer operates as push-pull driver															
1	1	Output buffer operates as P-ch open drain driver															

NOTE

The control bits of the JTAG Port Open Drain Control Register (JPODC0) are JPODC0_[7:0].

(b) PODCEn/JPODCE0 — Port Open Drain Control Expansion Register

This register selects port pin's output buffer function as push-pull type or emulated open drain type.

CAUTION

These registers are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PODCE n_15	PODCE n_14	PODCE n_13	PODCE n_12	PODCE n_11	PODCE n_10	PODCE n_9	PODCE n_8	PODCE n_7	PODCE n_6	PODCE n_5	PODCE n_4	PODCE n_3	PODCE n_2	PODCE n_1	PODCE n_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.34 PODCEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PODCEn_[15:0]	These bits select port pin's output buffer function as push-pull type or emulated open drain type in combination with the PODCn register. See Table 2.33, PODCn Register Contents for register setting and port pin's output buffer function.

NOTE

The control bits of the JTAG Port Open Drain Control Expansion Register (JPODCE0) are JPODCE0_[7:0].

(3) PDSC/JPDSC — Port Drive Strength Control Register Group

(a) PDSCn/JPDSC0 — Port Drive Strength Control Register

This register is used to configure the driving ability (driving strength) of output port pins and multiplexed output pin functions in combination with the PUCcN register. Specific driving ability settings may be required, depending on the pin function to be used. For details, see *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

CAUTION

These registers are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDSCn _15	PDSCn _14	PDSCn _13	PDSCn _12	PDSCn _11	PDSCn _10	PDSCn _9	PDSCn _8	PDSCn _7	PDSCn _6	PDSCn _5	PDSCn _4	PDSCn _3	PDSCn _2	PDSCn _1	PDSCn _0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.35 PDSCn Register Contents

Bit Position	Bit Name	Function															
15 to 0	PDSCn_[15:0]	These bits set driving ability of output port pins and multiplexed output pin functions in combination with the PUCcN register. For the electrical characteristics for each setting, see <i>the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware</i> .															
		<table border="1"> <thead> <tr> <th>PUCcN_m</th> <th>PDSCn_m</th> <th>Output Driving Ability</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Drive strength 1 (Very Low)</td> </tr> <tr> <td>0</td> <td>1</td> <td>Drive strength 2 (Low)</td> </tr> <tr> <td>1</td> <td>0</td> <td>Drive strength 3 (Mid)</td> </tr> <tr> <td>1</td> <td>1</td> <td>Drive strength 4 (High)</td> </tr> </tbody> </table>	PUCcN_m	PDSCn_m	Output Driving Ability	0	0	Drive strength 1 (Very Low)	0	1	Drive strength 2 (Low)	1	0	Drive strength 3 (Mid)	1	1	Drive strength 4 (High)
PUCcN_m	PDSCn_m	Output Driving Ability															
0	0	Drive strength 1 (Very Low)															
0	1	Drive strength 2 (Low)															
1	0	Drive strength 3 (Mid)															
1	1	Drive strength 4 (High)															

NOTE

The control bits of the JTAG Port Drive Strength Control Register (JPDSC0) are JPDSC0_[7:0].

(b) PUCcN/JPUCc0 — Port Universal Control Register

This register is used to configure the driving ability (driving strength) of output port pins and multiplexed output pin functions in combination with the PDSCn register. Specific driving ability settings may be required, depending on the pin function to be used. For details, see *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

CAUTION

These registers are write-protected by PWE.

See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PUCcN _15	PUCcN _14	PUCcN _13	PUCcN _12	PUCcN _11	PUCcN _10	PUCcN _9	PUCcN _8	PUCcN _7	PUCcN _6	PUCcN _5	PUCcN _4	PUCcN _3	PUCcN _2	PUCcN _1	PUCcN _0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.36 PUCcN Register Contents

Bit Position	Bit Name	Function
15 to 0	PUCcN_[15:0]	These bits set driving ability of output port pins and multiplexed output pin functions in combination with the PDSCn register. See Table 2.35, PDSCn Register Contents for register setting and output driving ability.

NOTE

The control bits of the JTAG Port Universal Control Register (JPUCc0) are JPUCc0_[7:0].

(4) PIS/JPIS — Port Input Buffer Selection Register Group

(a) PISn/JPIS0 — Port Input Buffer Selection Register

This register and the PISAn register select the port pin's input buffer characteristics. Specific input characteristic settings may be required, depending on the pin function to be used. For details, see *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

CAUTION

PIS21 and PIS27 are write-protected by PWE.
See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PISn_15	PISn_14	PISn_13	PISn_12	PISn_11	PISn_10	PISn_9	PISn_8	PISn_7	PISn_6	PISn_5	PISn_4	PISn_3	PISn_2	PISn_1	PISn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.37 PISn Register Contents

Bit Position	Bit Name	Function	
15 to 0	PISn_[15:0]	These bits select input buffer type of pin m (m = 0 to 15) in combination with the PISAn register.	
	PISAn_m	PISn_m	Description
	0	0	Input buffer type is Schmitt 1
	0	1	Input buffer type is Schmitt MSC (Setting prohibited except for SchmittMSC pins.)
	1	x	Input buffer type is TTL

NOTES

- The control bits of the JTAG Port Input Buffer Selection Register (JPIS0) are JPIS0_[7:0].
- Schmitt MSC input is supported by P10_2, P10_4, P10_5, P12_0, P12_1, P12_2, P12_3, P12_4, P12_7, P14_4, P15_4, P15_5, P15_6 only. These pins are multiplexed with the function of RHSBnSlx, RHSBnEMRG [n = 0, 1, x = 0, 1].
- TTL input is not supported with P40 port group.
- For details of selectable input characteristics, refer to *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.
- Input buffer type of TDI, TCK and TMS is selected to TTL type when these pins are used as Nexus, LPD-4 pin and Boundary SCAN interface.

(b) PISAn/JPISA0 — Port Input Buffer Selection Advanced Register

This register and the PISn register select the port pin's input buffer characteristics. Specific input characteristic settings may be required, depending on the pin function to be used. For details, see *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

CAUTION

PISA21 and PISA27 are write-protected by PWE.
See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PISAn_15	PISAn_14	PISAn_13	PISAn_12	PISAn_11	PISAn_10	PISAn_9	PISAn_8	PISAn_7	PISAn_6	PISAn_5	PISAn_4	PISAn_3	PISAn_2	PISAn_1	PISAn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.38 PISAn Register Contents

Bit Position	Bit Name	Function
15 to 0	PISAn_[15:0]	These bits set input buffer selection in combination with the PISn register. See Table 2.37, PISn Register Contents for the register settings and selected input type.

NOTES

- The control bits of the JTAG Port Input Buffer Selection Advanced Register (JPISA0) are JPISA0_[7:0].
- Schmitt MSC input is supported by P10_2, P10_4, P10_5, P12_0, P12_1, P12_2, P12_3, P12_4, P12_7, P14_4, P15_4, P15_5, P15_6 only. These pins are multiplexed with RHSBnSlx, RHSBnEMRG [n = 0, 1, x = 0, 1].
- TTL input is not supported with P40 port group.
- For details of selectable input characteristics, refer to *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.
- Input buffer type of TDI, TCK and TMS is selected to TTL type when these pins are used as Nexus, LPD-4 pin and Boundary SCAN interface.

2.4.2.4 Other Registers

(1) PKCPROT — Port Keycode Protection Register

This register is used for protection against writing registers that may have a material effect on the system so that the application system is not incorrectly stopped due to program malfunction and the like.

For how to set protection, see **Section 2.4.4, Port Setting Procedures**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PKCPR OT_31	PKCPR OT_30	PKCPR OT_29	PKCPR OT_28	PKCPR OT_27	PKCPR OT_26	PKCPR OT_25	PKCPR OT_24	PKCPR OT_23	PKCPR OT_22	PKCPR OT_21	PKCPR OT_20	PKCPR OT_19	PKCPR OT_18	PKCPR OT_17	PKCPR OT_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PKCPR OT_15	PKCPR OT_14	PKCPR OT_13	PKCPR OT_12	PKCPR OT_11	PKCPR OT_10	PKCPR OT_9	PKCPR OT_8	PKCPR OT_7	PKCPR OT_6	PKCPR OT_5	PKCPR OT_4	PKCPR OT_3	PKCPR OT_2	PKCPR OT_1	PWEE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 2.39 PKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	PKCPROT_[31:1]	These bits enable or disable modification of the PWEE bit. The value written is not retained. These bits are always read as 0.*1
0	PWEE	PWE Enable bit 0: Disable write access of PWE register. 1: Enable write access of PWE register.

Note 1. Set A5A5 A500_H with {PKCPROT[31:1], PWEE} to disable writing the PWE register.
Set A5A5 A501_H with {PKCPROT[31:1], PWEE} to enable writing PWE register.

(2) PWE — Port Write Enable Register

This register is used to control which port's protected registers can be written. When PKCPROT.PWEE = 1, this register can be written. When PKCPROT.PWEE = 0, this register cannot be written.

For how to set protection, see **Section 2.4.4, Port Setting Procedures**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	PWE P34	PWE P33	PWE P32	—	PWE P30	PWE P27
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PWE P25	PWE P24	PWE P23	PWE P22	PWE P21	PWE P20	PWE P15	PWE P14	PWE P13	PWE P12	PWE P11	PWE P10	PWE P02	PWE P01	PWE P00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.40 PWE Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 22	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
21	PWEP34	Port Write enable for P34 0: Disable write access 1: Enable write access
20	PWEP33	Port Write enable for P33 0: Disable write access 1: Enable write access
19	PWEP32	Port Write enable for P32 0: Disable write access 1: Enable write access
18	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
17	PWEP30	Port Write enable for P30 0: Disable write access 1: Enable write access
16	PWEP27	Port Write enable for P27 0: Disable write access 1: Enable write access
15	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
14	PWEP25	Port Write enable for P25 0: Disable write access 1: Enable write access
13	PWEP24	Port Write enable for P24 0: Disable write access 1: Enable write access

Table 2.40 PWE Register Contents (2/2)

Bit Position	Bit Name	Function
12	PWEP23	Port Write enable for P23 0: Disable write access 1: Enable write access
11	PWEP22	Port Write enable for P22 0: Disable write access 1: Enable write access
10	PWEP21	Port Write enable for P21 0: Disable write access 1: Enable write access
9	PWEP20	Port Write enable for P20 0: Disable write access 1: Enable write access
8	PWEP15	Port Write enable for P15 0: Disable write access 1: Enable write access
7	PWEP14	Port Write enable for P14 0: Disable write access 1: Enable write access
6	PWEP13	Port Write enable for P13 0: Disable write access 1: Enable write access
5	PWEP12	Port Write enable for P12 0: Disable write access 1: Enable write access
4	PWEP11	Port Write enable for P11 0: Disable write access 1: Enable write access
3	PWEP10	Port Write enable for P10 0: Disable write access 1: Enable write access
2	PWEP02	Port Write enable for P02 0: Disable write access 1: Enable write access
1	PWEP01	Port Write enable for P01 0: Disable write access 1: Enable write access
0	PWEP00	Port Write enable for P00 0: Disable write access 1: Enable write access

Note. Register protection is not supported for virtual port groups.

(3) LVDSCTRLA — LVDS Control A Register

This register specifies whether to use the LVDS function.

For details, refer to **Section 2.4.5.4, Register Setting to Use the LVDS Buffer.**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LVDSCTRLA_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	LVDSCTRLA_3	—	—	LVDSCTRLA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W

Table 2.41 LVDSCTRLA Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	LVDSCTRLA_16	This bit specify the voltage for the LVDS input function: 0: LVDS input function of P21_2 and P21_3 pin is used in 5V mode. 1: LVDS input function of P21_2 and P21_3 pin is used in 3V mode.
15 to 4	—	When read, the value after reset is returned. When writing, write the value after reset.
3	LVDSCTRLA_3	These bits specify whether to use the LVDS function: 0: Disable LVDS output function of P21_4 and P21_5 pin. 1: Enable LVDS output function of P21_4 and P21_5 pin.
2, 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	LVDSCTRLA_0	This bit specify whether to use the LVDS function: 0: Disable LVDS input function of P21_2 and P21_3 pin. 1: Enable LVDS input function of P21_2 and P21_3 pin.

(4) LVDSCTRLB — LVDS Control B Register

This register specifies whether to use the LVDS function.

For details, refer to **Section 2.4.5.4, Register Setting to Use the LVDS Buffer.**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	LVDSCTRLB_5	LVDSCTRLB_4	LVDSCTRLB_3	—	—	LVDSCTRLB_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R/W

Table 2.42 LVDSCTRLB Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	LVDSCTRLB_5	This bit specify whether to use the LVDS function: 0: Disable LVDS output function of P13_0 and P13_1 pin. 1: Enable LVDS output function of P13_0 and P13_1 pin.
4	LVDSCTRLB_4	This bit specify whether to use the LVDS function: 0: Disable LVDS input function of P13_0 and P13_1 pin. 1: Enable LVDS input function of P13_0 and P13_1 pin.
3	LVDSCTRLB_3	This bit specify whether to use the LVDS function: 0: Disable LVDS output function of P13_2 and P13_3 pin. 1: Enable LVDS output function of P13_2 and P13_3 pin.
2, 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	LVDSCTRLB_0	This bit specify whether to use the LVDS function: 0: Disable LVDS input function of P14_4 and P14_5 pin. 1: Enable LVDS input function of P14_4 and P14_5 pin.

(5) LVDSCTRLC — LVDS Control C Register

This register specifies whether to use the LVDS function.

For details, refer to **Section 2.4.5.4, Register Setting to Use the LVDS Buffer.**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	LVDSCTRLC_3	—	LVDSCTRLC_1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R

Table 2.43 LVDSCTRLC Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	LVDSCTRLC_3	This bit specify whether to use the LVDS function: 0: Disable LVDS output function of P10_2 and P10_3 pin. 1: Enable LVDS output function of P10_2 and P10_3 pin.
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	LVDSCTRLC_1	This bit specify whether to use the LVDS function: 0: Disable LVDS output function of P10_0 and P10_1 pin. 1: Enable LVDS output function of P10_0 and P10_1 pin.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

2.4.3 Pin-Unit Register Explanation

2.4.3.1 PCRn_m/JPCRn_m — Port Control Register

Each register of a port group can be accessed via this register and the PCRn_m register can set all the functions of a single pin. For example, setting bit 6 of the PCRn_m register to 1 sets bit m of the PMCN register to 1 also.

Multiple bits in the PCRn_m register can be batch-set within the ranges described in **Section 2.4.4, Port Setting Procedures**, Individual Setting.

CAUTION

These registers are write-protected by PWE.
See **Section 2.4.2.4, Other Registers** for details.

Value after reset: Refer to **Section 2.5.1, List of Port Functions**.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	PINV	—	PODC	PODCE	—	PUCC	PDSC	—	PISA	—	PIS	PU	PD	PBDC	PIBC
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R/W	R	R/W	R/W	R	R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSFSOE	PSFSD	PSFSC	P	—	—	—	PPR	—	PMC	PIPC	PM	PFCEAE	PFCAE	PFCE	PFC
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to **Section 2.5.1, List of Port Functions**.

Table 2.44 PCRn_m Register Contents (1/2)

Bit Position	Bit Name	Function
31	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
30	PINV	Same function as bit m of the PINVn register
29	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
28	PODC	Same function as bit m of the PODCn register
27	PODCE	Same function as bit m of the PODCEn register
26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25	PUCC	Same function as bit m of the PUCCn register
24	PDSC	Same function as bit m of the PDSCn register
23	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
22	PISA	Same function as bit m of the PISAn register
21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	PIS	Same function as bit m of the PISn register
19	PU	Same function as bit m of the PUn register
18	PD	Same function as bit m of the PDn register

Table 2.44 PCRn_m Register Contents (2/2)

Bit Position	Bit Name	Function
17	PBDC	Same function as bit m of the PBDCn register
16	PIBC	Same function as bit m of the PIBCn register
15	PSFSOE	Same function as bit m of the PSFSOE _n register
14	PSFSD	Same function as bit m of the PSFSD _n register
13	PSFSC	Same function as bit m of the PSFSC _n register
12	P	Same function as bit m of the P _n register
11 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	PPR	Same function as bit m of the PPR _n register
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	PMC	Same function as bit m of the PMC _n register
5	PIPC	Same function as bit m of the PIPC _n register
4	PM	Same function as bit m of the PM _n register
3	PFCEAE	Same function as bit m of the PFCEAE _n register
2	PFCAE	Same function as bit m of the PFCAE _n register
1	PFCE	Same function as bit m of the PFCE _n register
0	PFC	Same function as bit m of the PFC _n register

NOTE

The control registers of the JTAG Port Control Register (JPCR0) are JPCR0_m (m = 0 to 3, 5)

2.4.4 Port Setting Procedures

Examples of port setting procedures are shown below.

For the procedures for setting the digital noise filter of a port, see **Section 2.6.1, Digital Noise Filter**.

For the procedures for setting the Port safe state mode, see **Section 2.6.2.3, Configuration Flow**.

CAUTION

Even if a port is used in S/W I/O control alternative mode in a batch setting, the PMn_m register should be set at the end of the process in order to prevent output glitches.

In a PCR configuration, do not change the value of the invert value register (PINVn_m) and output value register (Pn_m) at the same time after output enable is active. If these registers are changed, set them again from the start of the register setting flow.

When returning to the “Port initializing” state after port setting has been configured, please set port register again in the inverse order with Port Group Unit register. At that case, it can not be used with Pin Unit register.

NOTE

When writing a reserved bit in virtual port registers, any value can be set—not just the value after reset. For details of reserved bits, refer to **Section 2.5.1, List of Port Functions**.

2.4.4.1 Batch Setting

The following shows an example of collectively setting a port group.

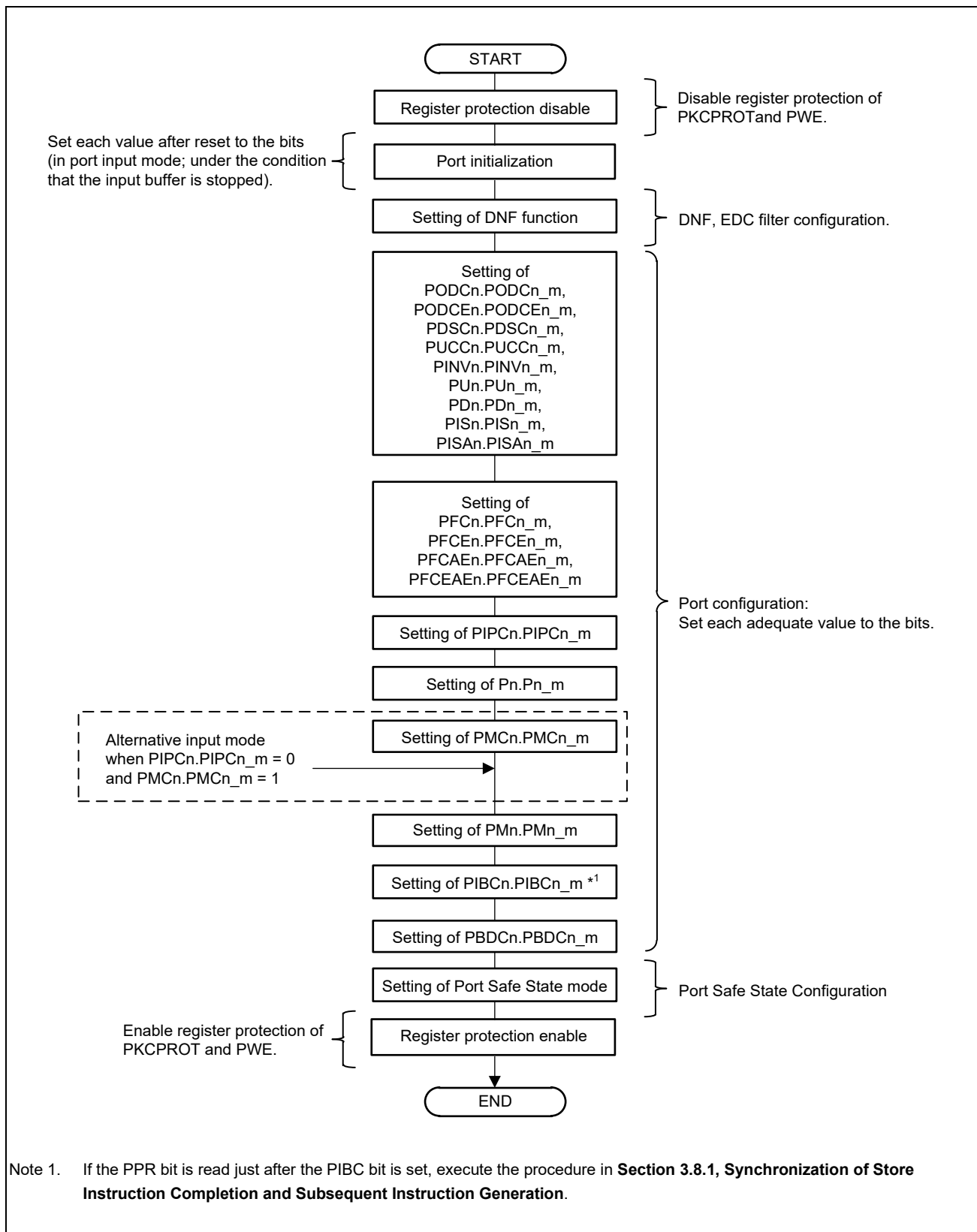


Figure 2.3 Port Setting Flow (Batch Setting)

2.4.4.2 Individual Setting

The following shows an example of individually setting a port group.

Bits can be set simultaneously within the settable range described below by using the PCRn_m register.

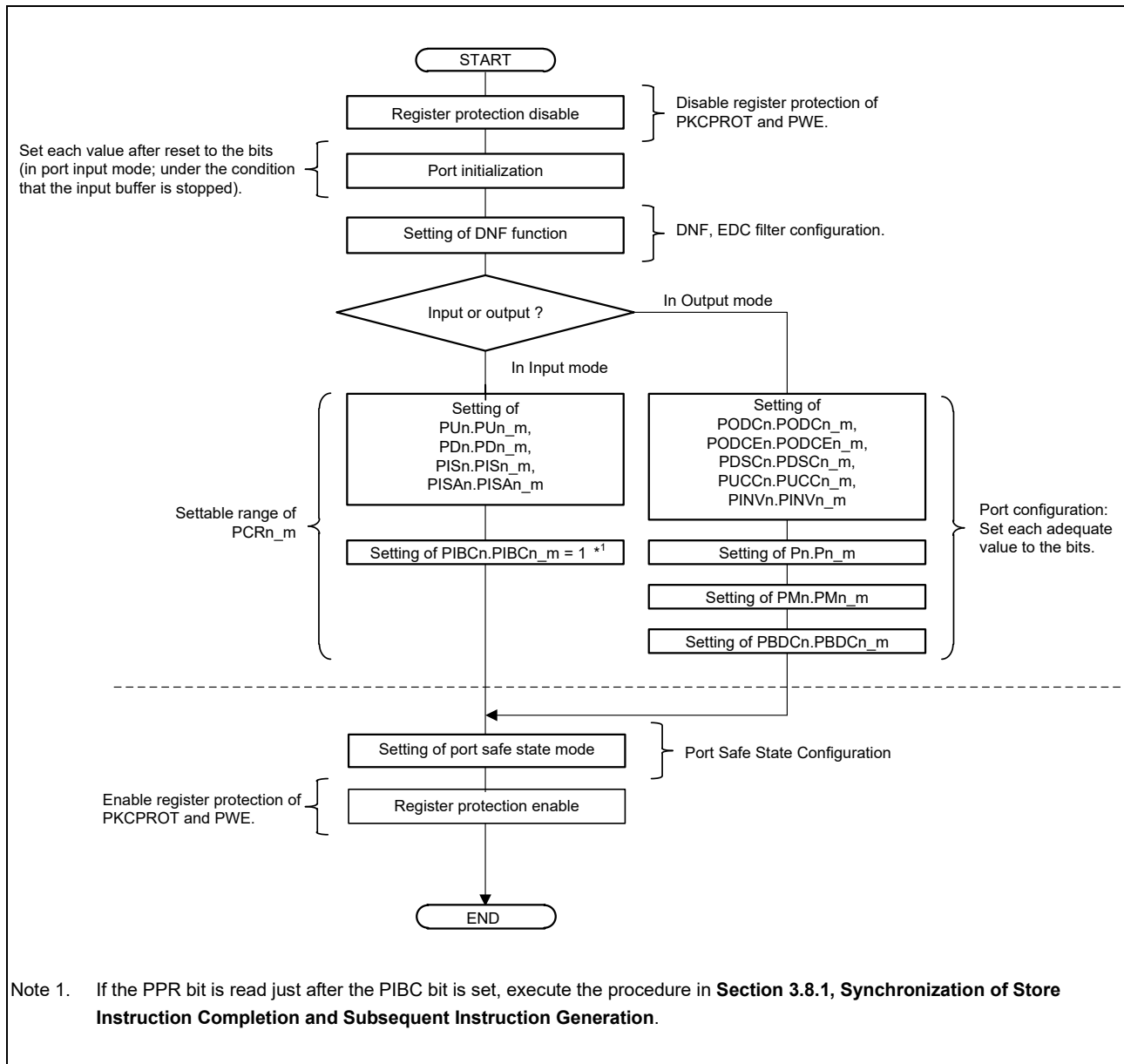


Figure 2.4 Port Setting Flow (in Port Mode)

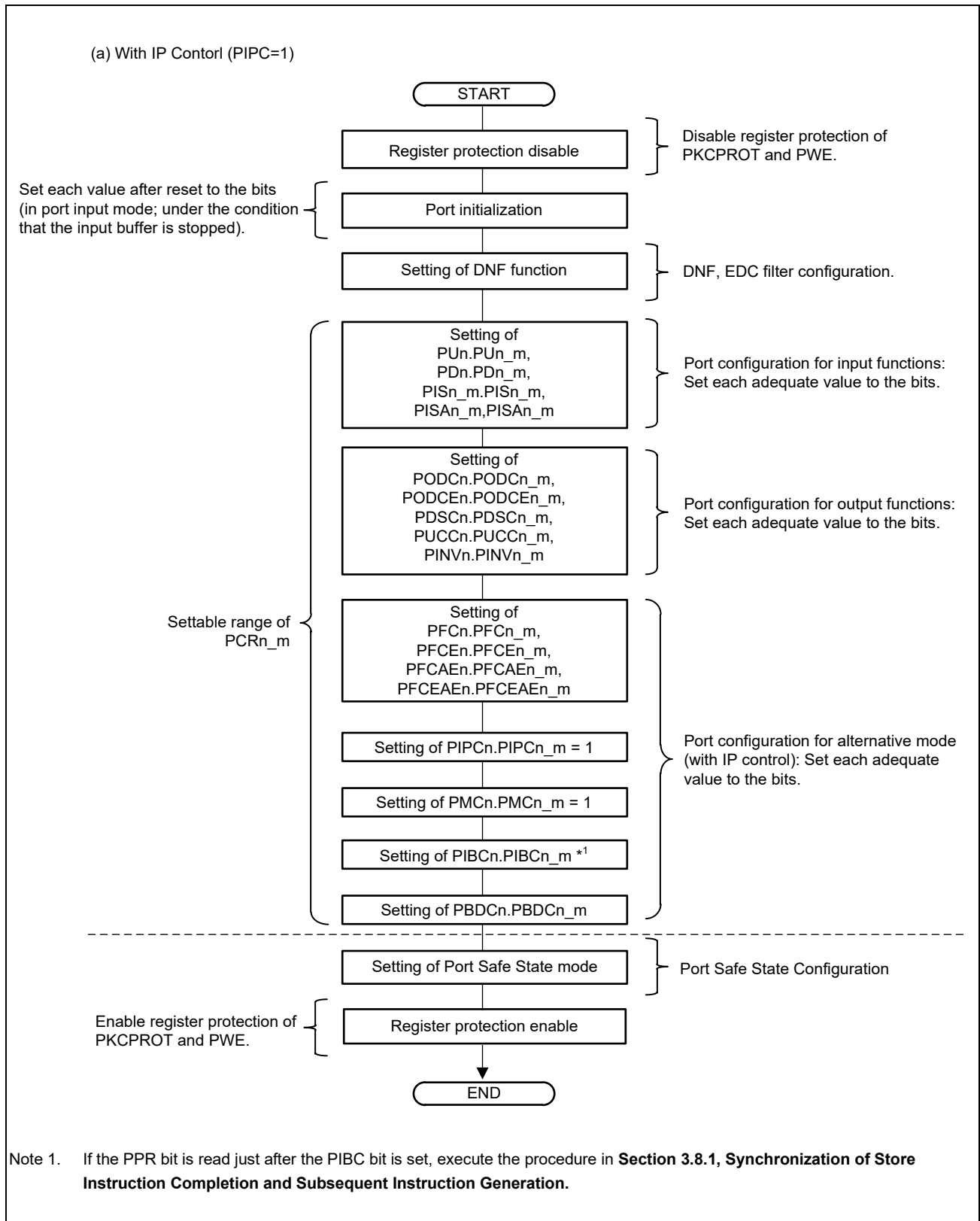


Figure 2.5 Port Setting Flow (in Alternative Mode) (1/3)

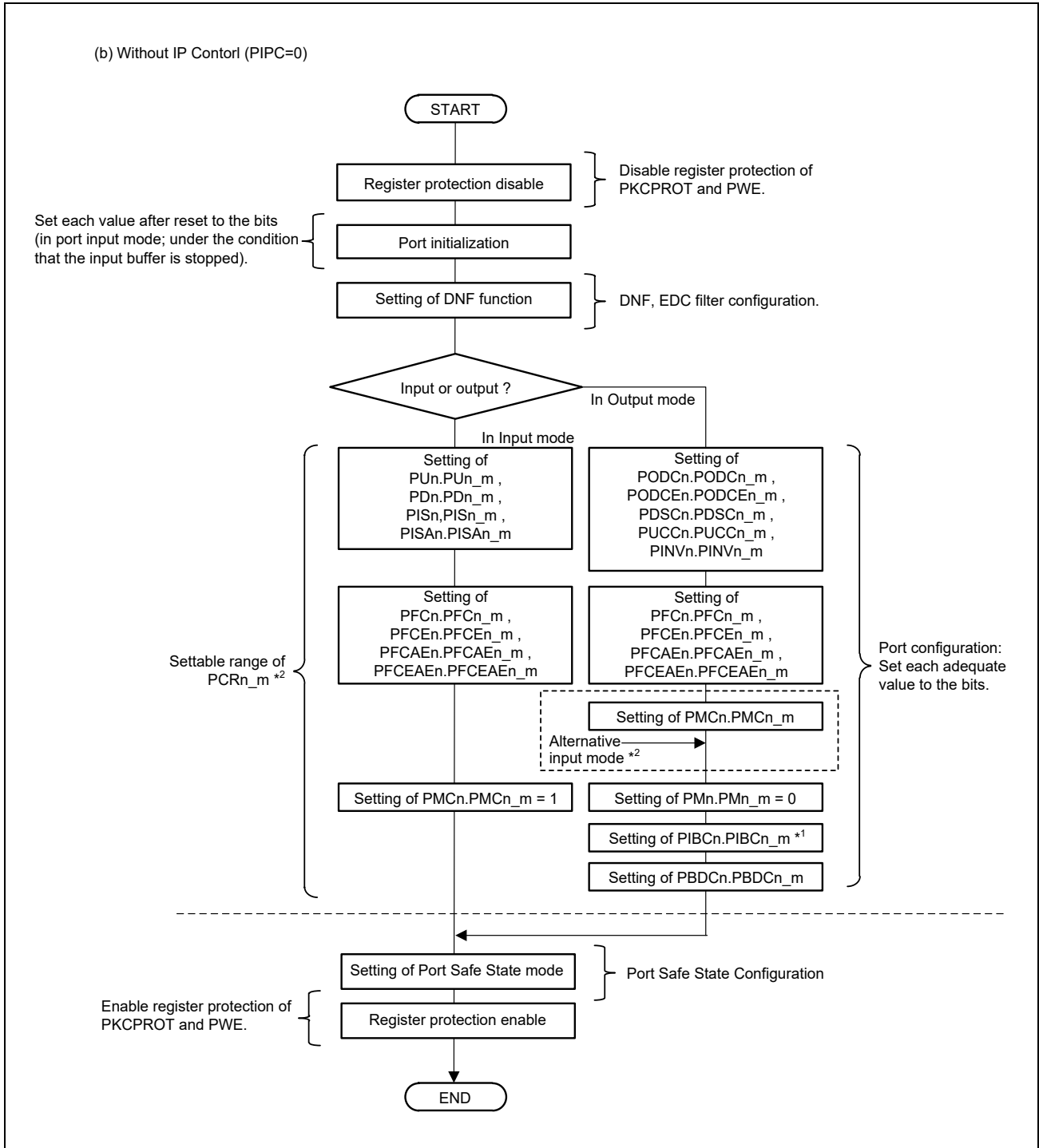


Figure 2.5 Port Setting Flow (in Alternative Mode) (2/3)

- Note 1. If the PPR bit is read just after the PIBC bit is set, execute the procedure in **Section 3.8.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation**.
- Note 2. In case of port configuration of the following alternative functions, there is a possibility that ECM Error Source Status for $\overline{\text{ERRORIN}}$ is set in ECMmESSTRn, and $\overline{\text{ERROROUT_M}}$ and $\overline{\text{ERROROUT_C}}$ output Low Level. ECM Error Source Status can be cleared by ECMESSTCn. $\overline{\text{ERROROUT_M}}$ and $\overline{\text{ERROROUT_C}}$ can be masked by ECMEMKn. Please set ECMEMK7 bit27 before port configuration of the following alternative function. Please clear ECMEMK7 bit27 after port configuration of that function if $\overline{\text{ERRORIN}}$ function is used.
- About ECMmESSTRn, ECMEMKn and ECMESSTCn, refer to **Section 39, Error Control Module (ECM)**.
- In case of Port configuration by Port Control Register(PCRN_m)
- P02_10: ATOM1_6 and ATOM5_2N
 - P15_6: TIM3_6, ATOM6_6 and ATOM7_6
 - P21_2: ATOM4_0 and ATOM2_4N
 - P32_0: LRX2 and ATOM1_5 (It is applicable to only E2xFCC1. Not applicable to E2M)
 - P32_5: ATOM3_5N

Figure 2.5 Port Setting Flow (in Alternative Mode) (3/3)

2.4.4.3 Protection Setting

The following shows an example of setting register protection.

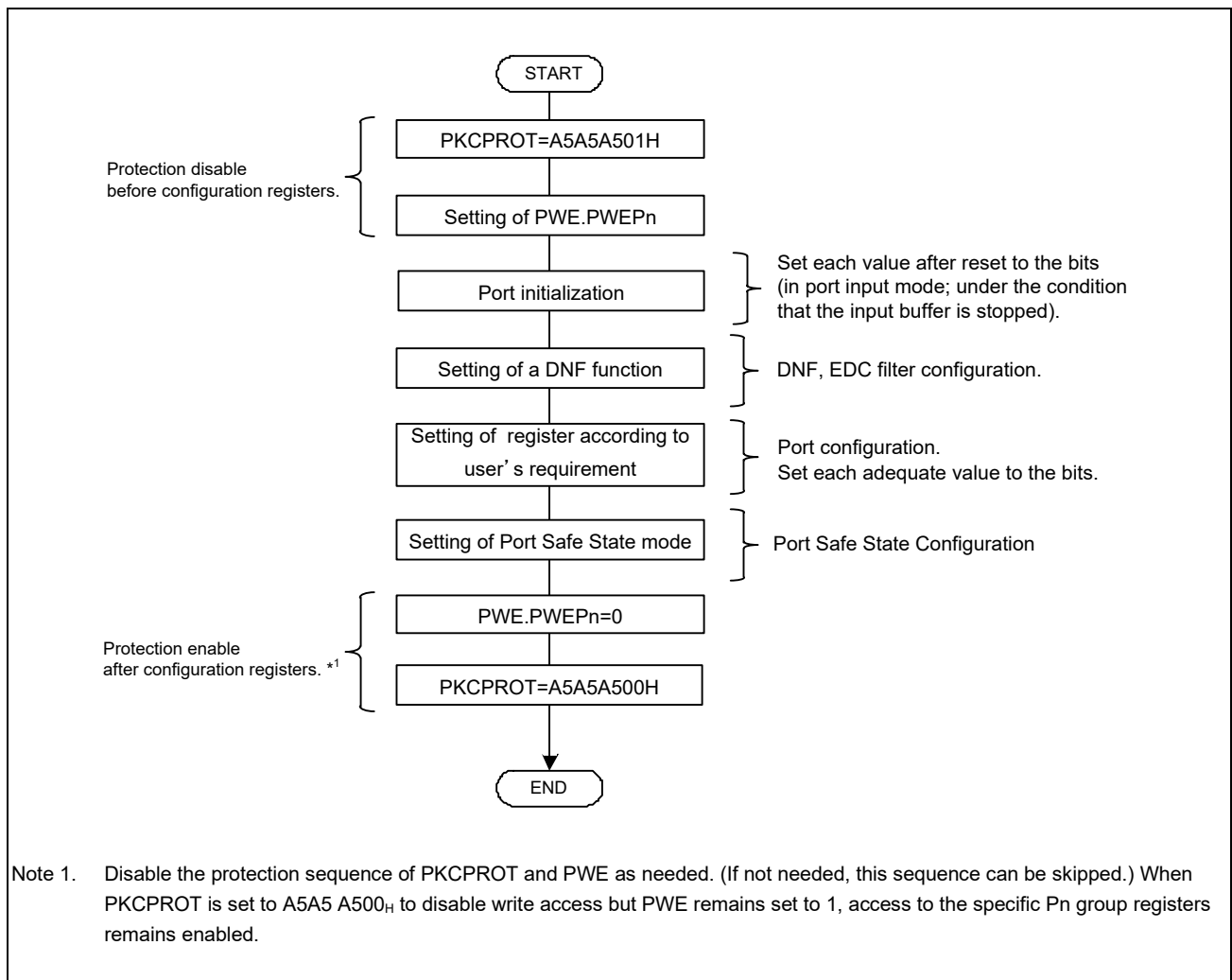


Figure 2.6 Protection Setting Flow

2.4.5 Functional Selection

2.4.5.1 Register Configuration When Using Alternative Functions

When the pin alternative function is used, set $PMn_m = 1$ and select the relevant alternative function by using the $PFCn_m$, $PFCEn_m$, $PFCAEn_m$ and $PFCEAEn_m$ bits.

For several peripheral functions, a single alternative I/O function is allocated to multiple pins. However, input functions should not be enabled on multiple pins at the same time.

2.4.5.2 Alternative Functions That Can Be Used in Direct I/O Control Alternative Mode

The following table shows alternative functions that can be used in direct I/O control alternative mode. To use these functions, set $PIPCn_m = 1$.

When $PIPCn_m = 1$, the set PMn_m value is ignored because the peripheral function controls the input and output of the buffer.

In addition, set $PIPCn_m = 0$ if you are using an alternative function not listed in following table.

When the LVDS buffer is to be used, $PIPCn_m$ register must not be set 1.

Table 2.45 List of Alternative Functions That Can Be Used in Direct I/O Control Alternative Mode (Must Set PIPCn_m = 1)

Category	Pin Name	I/O	Pin Function	Input/Output Control by Peripheral Functions
SCI3n	SCInRxD	I	Receive data input	Input through the pin is enabled when the SCI3nSCR.RE bit is set to enable reception by SCI3.
	SCInTxD	O	Transmit data output	Output through the pin is enabled when the SCI3nSCR.TE bit is set to enable transmission by SCI3.
	SCInSCK	I/O	Serial clock input/output	The SCI3nSCR.CKE[1:0] bits determine whether the pin is an input or output. <ul style="list-style-type: none"> Asynchronous transfer CKE[1:0] = 00: Both input and output through the pin are disabled. CKE[1:0] = 01: The pin is an output. CKE[1:0] = 1x: Prohibited settings Synchronous transfer CKE[1:0] = 0x: The pin is an output.
CSIHn	CSIHnTSSO	O	Transmit data output (data integrity verification)	<ul style="list-style-type: none"> Output control See “CSIHTSSO Operation for Slave Mode”, in Section 17, Clocked Serial Interface H (CSIH). Input control When the CSIHnCTL1.DCS bit is set to enable checking the consistency of data through CSIH, enabling the input buffer for the pin leads to the actual output level of the pin being looped back within the device.
	CSIHnTSCK	I/O	Serial clock input/output	When the setting of CSIHnCTL2 is for CSIH operation in slave mode, this pin is an input. When the setting is for master mode, it is an output.
	CSIHnTRY	I/O	Handshake signal input/output	When the setting of CSIHnCTL2 is for CSIH operation in slave mode, this pin is an output. When the setting is for master mode, it is an input.
ATU-V timer C	TIOCn0 to TIOCn3	I/O	Input capture trigger input, output compare output	This pin is an output when ATU-V timer C is set to a compare-match operation by the TIOCx register. This pin is an input when the setting is for an input capture operation.
HSSPIn	HSSPIn_TXD	O	Transmit data output	For details of output control and input control, see “Control of the HS-SPI Pins”, in Section 26, High Speed Serial Peripheral Interface (HS-SPI) .
	HSSPIn_CLK	I/O	Serial clock input/output	
	HSSPIn_SSLI	I	Slave selection input signal	
	HSSPIn_SSL	O	Slave selection output signal	
Ethernet	MDIO	I/O	PHY chip set I/F data input/output.	Input through the pin is enabled when the ETNCnPIR.MMD and ETNCnPIR.MDI bits are enabled. Output through the pin is enabled when the ETNCnPIR.MMD and ETNCnPIR.MDO bits are enabled.

Note: n indicates the number of a given peripheral module.

2.4.5.3 Register Setting When Using an Analog Input Pin

It is not necessary to set a port register to select a pin function since ANxxx (SAR-A/D), DSANxxxP, DSANxxxN (DS-A/D), CANxxxP and CANxxxN (Cyclic-A/D) for an analog input are always connected to the A/D converter.

However, some analog input ports are multiplexed with general input ports, so to use the ports as analog input ports, set $PMCn_m = 0^{*1}$ and $PIBCn_m = 0$ (the value after reset of the port register). This prevents shoot-through current from the input buffer on the digital side during an analog voltage input. The relevant pins of this product are port groups P40_m.

Note 1. Some pins may not be equipped with a PMC40_m bit. For further details, see **Section 2.5, Organization of Port Groups**.

2.4.5.4 Register Setting to Use the LVDS Buffer

When the LVDS buffer is to be used, set the LVDSCTRL[A,B,C] register to enable input/output. Also, set whether the supply voltage of LVDS buffer is 3V or 5V.

CAUTION

Input/output to/from the LVDS buffer actually being enabled/disabled takes at least 10 μ s after setting LVDSCTRL[A,B,C]. The following pins have LVDS buffer and also port functions, but these cannot be used at the same time. Be sure to set value after reset to each registers*1 as initialization in case LVDS buffer is used. In LVDS input mode, PFCn.PFCn_m, PFCEn.PFCEn_m and PMCn.PMCn_m can be set after IE bit of LVDSCTRL[A, B] is set. In LVDS output mode, PFCn.PFCn_m, PFCEn.PFCEn_m and PMCn.PMCn_m can be set before OE bit of LVDSCTRL[A, B, C] is set. And PMn.PMn_m and PSFSCn.PSFSCn_m can be set after OE bit of LVDSCTRL[A, B, C] is set. Refer to **Section 2.4.5.5, Setting Flow of LVDS function** for more detail.

Note 1. Set PIBCn.PIBCn_m = 0, PBDCn.PBDCn_m = 0, PMn.PMn_m = 1, PMCn.PMCn_m = 0, PIPCn, PIPCn_m = 0, PFCn.PFCn_m = 0, PFCEn.PFCEn_m = 0, PFCAEn.PFCAEn_m = 0, PFCEAEn.PFCEAEn_m = 0, PDn.PDn_m = 0, PUn.PUn_m = 0, PSFSCn.PSFSCn_m = 0, PSFSDn.PSFSDn_m = 0, PSFSOEn.PSFSOEn_m = 0.

The LVDS setting registers are shown in the following table.

Table 2.46 LVDS Enable Register

Port Register	Register Bit	Power Domain	I/F	Pin Name (A pair of Pch, Nch)	I/O	Control*1	Related Function (A pair of Pch, Nch)	E2x-FCC1		E2M	
								BGA 373	BGA 292 (E2M)	BGA 373	BGA 292
LVDS CTRLA	[3]	LVDVCC	IEEE	P21_5, P21_4	OUT	OE	HSIF0_TXDP, HSIF0_TXDN	✓	✓	✓	✓
	[0]				IN	IE	HSIF0_RXDP, HSIF0_RXDN	✓	✓	✓	✓
	[16]			IN	VSEL	HSIF0_RXDN	✓	✓	✓	✓	
LVDS CTRLB	[5]	E0VCC	ANSI	P13_1, P13_0	OUT	OE	RHSB0FCLP, RHSB0FCLN	✓	✓	✓	✓
					OUT	OE	HSSPI0_CLKP, HSSPI0_CLKN	✓	✓	✓	✓
	[4]			IN	IE	HSSPI0_CLKP, HSSPI0_CLKN	✓	✓	✓	✓	
				[3]	P13_3, P13_2	OUT	OE	RHSB0SOP, RHSB0SON	✓	✓	✓
	OUT					OE	HSSPI0_TXDP, HSSPI0_TXDN	✓	✓	✓	✓
	[0]			P14_4, P14_5	IN	IE	HSSPI0_RXDP, HSSPI0_RXDN	✓	✓	✓	✓
LVDS CTRLC	[3]	E0VCC	ANSI	P10_2, P10_3	OUT	OE	RHSB1SOP, RHSB1SON	✓	✓	✓	✓
	[1]				OUT	OE	RHSB1FCLP, RHSB1FCLN	✓	✓	✓	✓

Note 1. OE: Output Enable, IE: Input Enable. VSEL: Supply Voltage Select.

2.4.5.5 Setting Flow of LVDS function

The following shows the flow to use LVDS function.

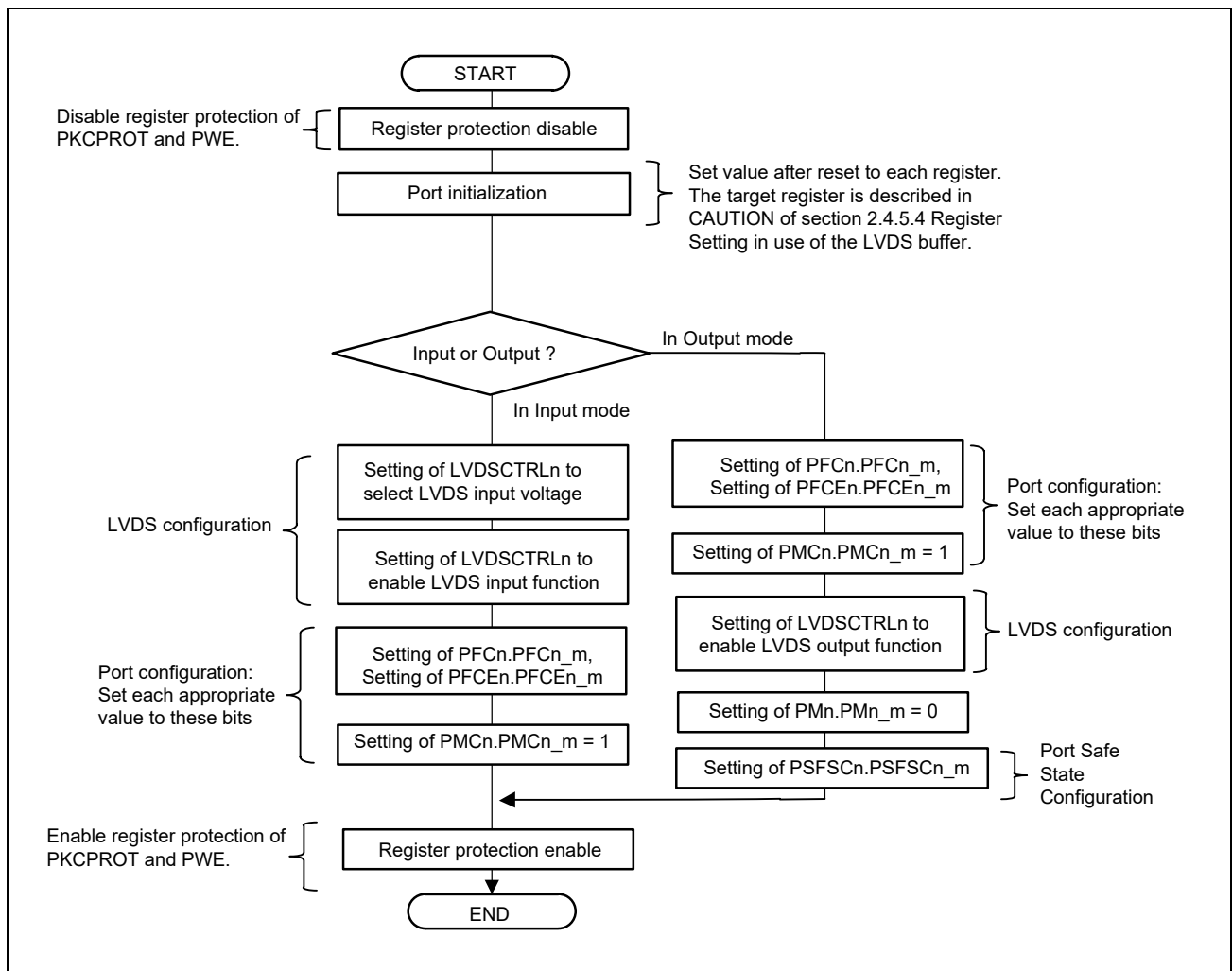


Figure 2.7 LVDS Configuration Flow

2.4.5.6 Selecton of Function for JTAG Port

To enable connection of multiple tools to the JTAG port, the operation mode can be defined by a combination of mode pin and option byte settings. For details, see **Section 45, Debugging and Calibration**.

2.4.5.7 Setting of the ERROROUT_C loopback

When the pin level of ERROROUT_C is confirmed by ECM Checker Error Source Status register, set PBDCn_m = 1 to enable the alternative output level loopback function.

See **Section 2.4.2.2(1), PBDCn/JPBDC0 — Port Bidirectional Control Register** for details.

2.5 Organization of Port Groups

2.5.1 List of Port Functions

For detailed information on the control registers, refer to Appendix “*E02-02_List of Port Functions.xlsx*”.

In the bitmap field, “x” means an effective bit and “—” means a reserved bit. Reserved areas are always read as the values after reset. The write value also should be the value after reset. The registers of unimplemented pins should not be modified after reset.

2.5.2 List of Alternative Functions

For detailed information on the alternative functions of each pin, refer to Appendix “*E02-01_List of Alternative Functions.xlsx*”.

In the file, “—” means a reserved function, and this function should not be selected. The functions of unimplemented ports cannot be selected and the registers of these unimplemented pins should not be modified after reset.

2.6 Functions

2.6.1 Digital Noise Filter

The Digital Noise Filter (DNF) eliminates digital noise from external input signals. This product includes two types of DNF: a peripheral function DNF and an edge detection DNF.

2.6.1.1 Example of Noise Elimination

Figure 2.8 Timing Chart of Digital Noise Elimination shows an example of noise elimination using a peripheral function DNF and an edge detection DNF. In this example, the sampling clock is set to 1/2 of the DNF input clock, the sampling count is set to two (twice), and the current output level is set to low level. “○” in the figure means that high level is detected.

In input examples 1, 2, and 3, the output level changes from low to high because the same level is detected twice in a row through sampling. In input examples 4, 5, and 6, on the other hand, the same level is not detected twice consecutively. Therefore, these inputs are regarded as noise and the input signal state is eliminated.

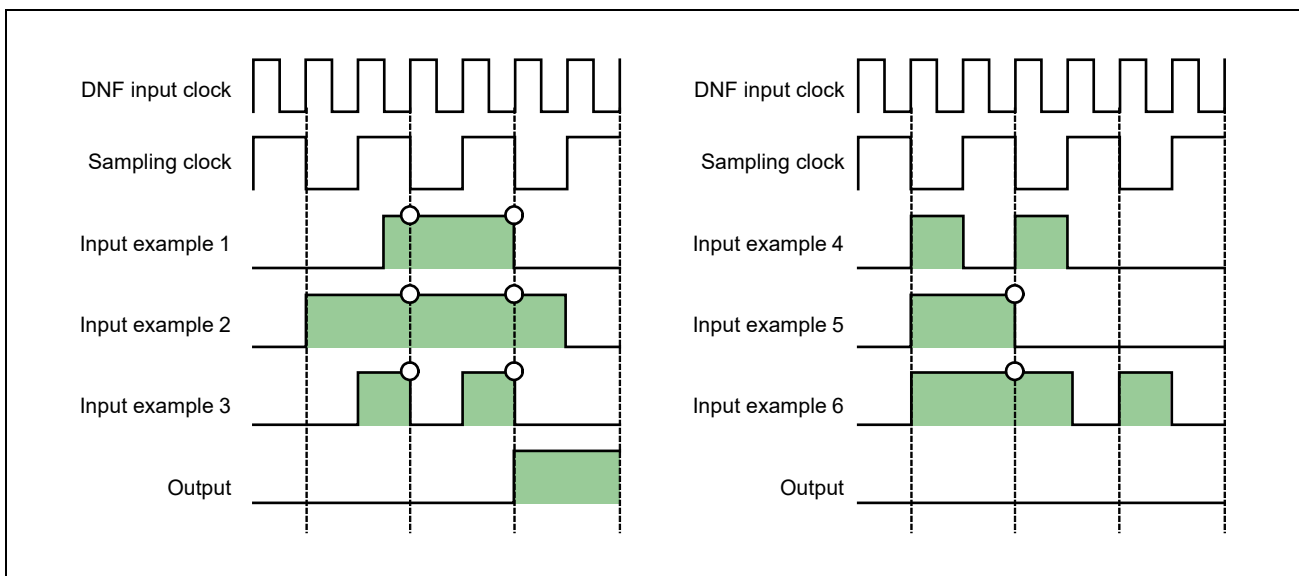


Figure 2.8 Timing Chart of Digital Noise Elimination

2.6.1.2 Peripheral Function DNF

(1) Function Overview

This DNF eliminates noise from the input pins for peripheral functions.

The peripheral function DNF has the following functions:

- Eliminates digital noise from input signals and outputs noiseless signals.
- Selects whether to output signals from which digital noise is eliminated or signals containing digital noise.
- Selects the digital noise elimination width from 2, 3, 4, and 5 counts of the sampling clock.
- Selects from the five types of sampling frequency shown below:
 - 1/1, 1/2, 1/4, 1/8, and 1/16 of the DNF input clock.
- The conditions for noise elimination can be set by each channel via registers.
- The DNF input clock of DNF group number 0, 2 and 3 is a low-speed peripheral clock.
- The DNF input clock of DNF group number 1 is a high-speed peripheral clock.

(2) DNF Register Base Addresses

The following table shows base addresses of DNF.

Table 2.47 Base Addresses of Peripheral Function DNF

Base Address Name	Base Address	Bus Group
<DNF0_Base>	FFC3 0000 _H	Peripheral Group 6
<DNF1_Base>	FFC3 0040 _H	Peripheral Group 6
<DNF2_Base>	FFC3 0080 _H	Peripheral Group 6
<DNF3_Base>	FFC3 00C0 _H	Peripheral Group 6

(3) DNF Clock Supplies

The peripheral function DNF clock supplies are shown in the following table.

Table 2.48 Clock Supplies of Peripheral Function DNF

Unit Name	Unit Clock Name	Supply Clock Name
DNF0	PCLK	CLK_LSB
DNF1	PCLK	CLK_HSB
DNF2	PCLK	CLK_LSB
DNF3	PCLK	CLK_LSB

(4) DNF Reset Sources

The peripheral function DNF reset sources are shown below.

The peripheral function DNF is initialized by the following reset sources.

Table 2.49 Reset Sources of Peripheral Function DNF

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
DNF0	All registers	✓	✓	✓	✓	—	—
DNF1	All registers	✓	✓	✓	✓	—	—
DNF2	All registers	✓	✓	✓	✓	—	—
DNF3	All registers	✓	✓	✓	✓	—	—

(5) List of Peripheral Function DNF Registers

Table 2.50 List of Registers

Module Name	Register Name	Symbol	Address	Value after reset	Access	Access Protection
DNFn	DNF Control Register	DNFP01nCTLm	<DNFn_base> + 04 _H × m	00 _H	8	—

Note: n: DNF group number, m: channel number

(6) DNFP01nCTLm — DNF Control Register

This register sets conditions for noise elimination of channel number m in DNF group number n.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NFEN	SLST[1:0]		—	—	PRS[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Table 2.51 DNFP01nCTLm Register Contents

Bit Position	Bit Name	Function
7	NFEN	This bit enables/disables digital noise elimination. 0: Digital noise is not eliminated. 1: Digital noise is eliminated.
6, 5	SLST[1:0]	These bits specify the sampling count for digital noise elimination. 00: Sampling count = 2 01: Sampling count = 3 10: Sampling count = 4 11: Sampling count = 5
4, 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	PRS[2:0]	These bits specify the sampling clock division ratio for digital noise elimination. 000: DNF input clock/1 001: DNF input clock/2 010: DNF input clock/4 011: DNF input clock/8 100: DNF input clock/16 Other than the above: Setting is prohibited.

(7) Setting Flow of Peripheral Function DNF

The following shows the procedures to set the peripheral function DNF.

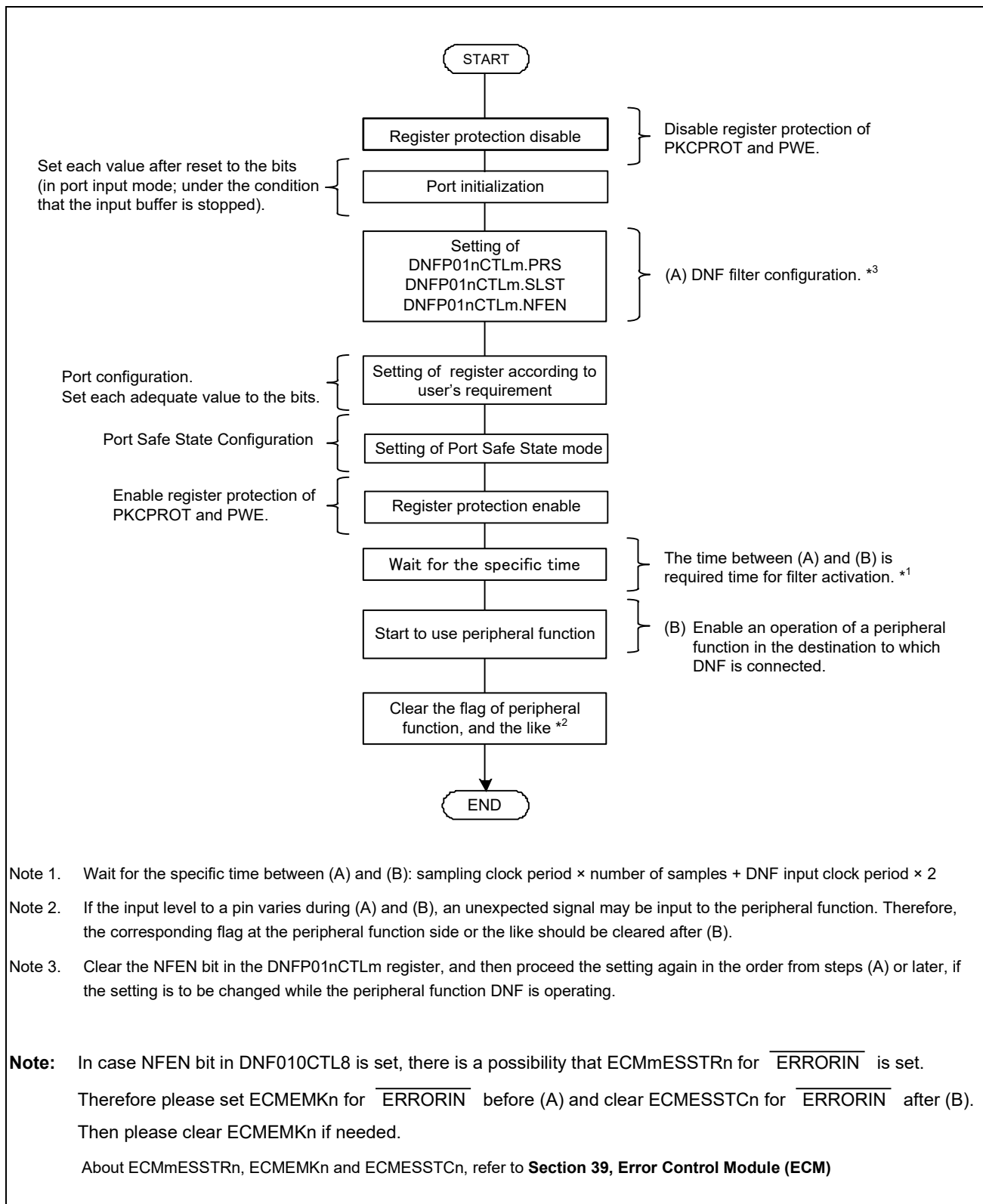


Figure 2.9 Peripheral Function DNF Setting Flow

(8) List of Pins on Which a DNF Is Inserted

The following table shows the pins on which a DNF is inserted.

Table 2.52 Pins on Which DNF is Inserted

DNF Group	DNF Channel Number (m)	DNF Insertion Target Pin			E2x-FCC1		E2M	
		Pin Name	Pin Function	Peripheral Function	BGA 373	BGA 292 (E2M)	BGA 373	BGA 292
DNF0	0	IRQ0	External interrupt input	INTC	✓	✓	✓	✓
	1	IRQ1	External interrupt input	INTC	✓	✓	✓	✓
	2	IRQ2	External interrupt input	INTC	✓	✓	✓	✓
	3	IRQ3	External interrupt input	INTC	✓	✓	✓	✓
	4	IRQ4	External interrupt input	INTC	✓	✓	✓	✓
	5	IRQ5	External interrupt input	INTC	✓	✓	✓	✓
	6	IRQ6	External interrupt input	INTC	✓	✓	✓	✓
	7	IRQ7	External interrupt input	INTC	✓	✓	✓	✓
	8	ERRORIN	Error input	ECM	✓	✓	✓	✓
DNF1	0	ADTRG0	SAR-AD conversion startup trigger input	SAR-AD0	✓	✓	✓	✓
	1	ADTRG1	SAR-AD conversion startup trigger input	SAR-AD1	✓	—	✓	—
	2	ADTRG2	SAR-AD conversion startup trigger input	SAR-AD2	✓	✓	✓	✓
	3	ADTRG3	SAR-AD conversion startup trigger input	SAR-AD3	✓	—	✓	—
	4	RHSB0EMRG	Emergency signal input	RHSB0	✓	✓	✓	✓
	5	RHSB1EMRG	Emergency signal input	RHSB1	✓	✓	✓	✓
	8	ENCA0E0	ENCA ch0 E0 input	ENCA	✓	✓	✓	✓
	9	ENCA0E1	ENCA ch0 E1 input	ENCA	✓	✓	✓	✓
	10	ENCA0EC	ENCA ch0 EC input	ENCA	✓	✓	✓	✓
	11	ENCA1E0	ENCA ch1 E0 input	ENCA	✓	✓	✓	✓
	12	ENCA1E1	ENCA ch1 E1 input	ENCA	✓	✓	✓	✓
	13	ENCA1EC	ENCA ch1 EC input	ENCA	✓	✓	✓	✓
	DNF2	0	RSENT0RX	RSENT0 receive data input	RSENT0	✓	✓	✓
1		RSENT1RX	RSENT1 receive data input	RSENT1	✓	✓	✓	✓
2		RSENT2RX	RSENT2 receive data input	RSENT2	✓	✓	✓	✓
3		RSENT3RX	RSENT3 receive data input	RSENT3	✓	✓	✓	✓
4		RSENT4RX	RSENT4 receive data input	RSENT4	✓	✓	✓	✓
5		RSENT5RX	RSENT5 receive data input	RSENT5	✓	✓	✓	✓
6		RSENT6RX	RSENT6 receive data input	RSENT6	✓	✓	✓	✓
7		RSENT7RX	RSENT7 receive data input	RSENT7	✓	✓	✓	✓
DNF3	0	RSENT8RX	RSENT8 receive data input	RSENT8	✓	✓	✓	✓
	1	RSENT9RX	RSENT9 receive data input	RSENT9	✓	✓	✓	✓
	2	RSENT10RX	RSENT10 receive data input	RSENT10	✓	✓	✓	✓
	3	RSENT11RX	RSENT11 receive data input	RSENT11	✓	✓	✓	✓
	4	RSENT12RX	RSENT12 receive data input	RSENT12	✓	✓	✓	✓
	5	RSENT13RX	RSENT13 receive data input	RSENT13	✓	✓	✓	✓
	6	RSENT14RX	RSENT14 receive data input	RSENT14	✓	✓	✓	✓
	7	RSENT15RX	RSENT15 receive data input	RSENT15	✓	✓	✓	✓
	8	RSENT16RX	RSENT16 receive data input	RSENT16	✓	✓	✓	✓

2.6.1.3 Edge Detection DNF

(1) Function Overview

This DNF detects the valid edge of a pin input signal. The signals to be used for edge detection can be selected either before or after noise elimination.

The edge detection DNF has the following functions:

- Eliminates digital noise from input signals and outputs noiseless signals.
- Selects whether to output signals from which digital noise is eliminated or signals containing digital noise.
- Selects the digital noise elimination width from 2, 3, 4, and 5 counts of the sampling clock.
- Selects from the five types of sampling frequency shown below: 1/1, 1/2, 1/4, 1/8, and 1/16 of the DNF input clock.
- The conditions for noise elimination can be set for all channels collectively by a register.
- The DNF input clock is a low-speed peripheral clock.

(2) Edge Detection DNF Register Base Addresses

The following table shows the base addresses of the edge detection DNF. “n” indicates the number of the DNF group.

Table 2.53 Base Addresses of Edge Detection DNF

Base Address Name	Base Address	Bus Group	E2x-FCC1		E2M	
			BGA373	BGA292 (E2M)	BGA373	BGA292
<EDC00_Base>	FFC3 1000 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC01_Base>	FFC3 1100 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC02_Base>	FFC3 1200 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC10_Base>	FFC3 1A00 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC11_Base>	FFC3 1B00 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC12_Base>	FFC3 1C00 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC13_Base>	FFC3 1D00 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC14_Base>	FFC3 1E00 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC15_Base>	FFC3 1F00 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC20_Base>	FFC3 2400 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC21_Base>	FFC3 2500 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC22_Base>	FFC3 2600 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC23_Base>	FFC3 2700 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC24_Base>	FFC3 2800 _H	Peripheral Group 6	✓	—	✓	—
<EDC25_Base>	FFC3 2900 _H	Peripheral Group 6	✓	—	✓	—
<EDC27_Base>	FFC3 2B00 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC30_Base>	FFC3 2E00 _H	Peripheral Group 6	✓	—	✓	—
<EDC32_Base>	FFC3 3000 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC33_Base>	FFC3 3100 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC34_Base>	FFC3 3200 _H	Peripheral Group 6	✓	✓	✓	✓
<EDC40_Base>	FFC3 3800 _H	Peripheral Group 6	✓	✓	✓	✓

(3) DNF Clock Supplies

The edge detection DNF clock supplies are shown in the following table.

Table 2.54 Clock Supplies of Edge Detection DNF

Unit Name	Unit Clock Name	Supply Clock Name	E2x-FCC1		E2M	
			BGA373	BGA292 (E2M)	BGA373	BGA292
EDC00	PCLK	CLK_LSB	✓	✓	✓	✓
EDC01	PCLK	CLK_LSB	✓	✓	✓	✓
EDC02	PCLK	CLK_LSB	✓	✓	✓	✓
EDC10	PCLK	CLK_LSB	✓	✓	✓	✓
EDC11	PCLK	CLK_LSB	✓	✓	✓	✓
EDC12	PCLK	CLK_LSB	✓	✓	✓	✓
EDC13	PCLK	CLK_LSB	✓	✓	✓	✓
EDC14	PCLK	CLK_LSB	✓	✓	✓	✓
EDC15	PCLK	CLK_LSB	✓	✓	✓	✓
EDC20	PCLK	CLK_LSB	✓	✓	✓	✓
EDC21	PCLK	CLK_LSB	✓	✓	✓	✓
EDC22	PCLK	CLK_LSB	✓	✓	✓	✓
EDC23	PCLK	CLK_LSB	✓	✓	✓	✓
EDC24	PCLK	CLK_LSB	✓	—	✓	—
EDC25	PCLK	CLK_LSB	✓	—	✓	—
EDC27	PCLK	CLK_LSB	✓	✓	✓	✓
EDC30	PCLK	CLK_LSB	✓	—	✓	—
EDC32	PCLK	CLK_LSB	✓	✓	✓	✓
EDC33	PCLK	CLK_LSB	✓	✓	✓	✓
EDC34	PCLK	CLK_LSB	✓	✓	✓	✓
EDC40	PCLK	CLK_LSB	✓	✓	✓	✓

(4) DNF Reset Sources

The edge detection DNF reset sources are shown below.

The edge detection DNF is initialized by the following reset sources.

Table 2.55 Reset Sources of Edge Detection DNF

Unit Name	Register Name	Reset Condition					E2x-FCC1			E2M	
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module reset	JTAG reset	BGA373	BGA292 (E2M)	BGA373	BGA292
EDC00	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC01	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC02	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC10	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC11	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC12	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC13	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC14	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC15	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC20	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC21	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC22	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC23	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC24	All registers	✓	✓	✓	✓	—	—	✓	—	✓	—
EDC25	All registers	✓	✓	✓	✓	—	—	✓	—	✓	—
EDC27	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC30	All registers	✓	✓	✓	✓	—	—	✓	—	✓	—
EDC32	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC33	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC34	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓
EDC40	All registers	✓	✓	✓	✓	—	—	✓	✓	✓	✓

(5) List of Edge Detection DNF Registers

Table 2.56 List of Registers

Module Name	Register Name	Symbol	Address	Value after reset	Access	Access Protection
EDCn	DNF Control Register	DNFP02nCTL	<EDCn_base> + 00 _H	00 _H	8	—
	Edge detection control Register	DNFP02nEDCm	<EDCn_base> + 04 _H + 08 _H × m	00 _H	8	—
	Edge detection Flag Register	DNFP02nEDFm	<EDCn_base> + 08 _H + 08 _H × m	00 _H	8	—

Note: n: DNF group number, m: channel number

(6) DNFP02nCTL — DNF Control Register

This register sets noise elimination conditions which are commonly used for all channels in DNF group number n.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	SLST[1:0]		—	—	PRS[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R	R	R/W	R/W	R/W

Table 2.57 DNFP02nCTL Register Contents

Bit Position	Bit Name	Function
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6, 5	SLST[1:0]	These bits specify the sampling count for digital noise elimination. 00: Sampling count = 2 01: Sampling count = 3 10: Sampling count = 4 11: Sampling count = 5
4, 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	PRS[2:0]	These bits specify the sampling clock division ratio for digital noise elimination. 000: DNF input clock/1 001: DNF input clock/2 010: DNF input clock/4 011: DNF input clock/8 100: DNF input clock/16 Other than the above: Setting is prohibited.

(7) DNFP02nEDCm — Edge Detection Control Register

This register controls edge detection of channel number m in DNF group number n.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NFEN	—	—	DMD[1:0]		—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R/W	R/W	R	R	R

Table 2.58 DNFP02nEDCm Register Contents

Bit Position	Bit Name	Function
7	NFEN	This bit enables/disables digital noise elimination. 0: Edge is detected for signals containing digital noise. 1: Edge is detected for signals from which digital noise is eliminated
6, 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4, 3	DMD[1:0]	This bit enables/disables edge detection and sets edge to be detected 00: Edge detection is disabled. 01: Rising edge is detected. 10: Falling edge is detected. 11: Both edges are detected.
2 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

(8) DNFP02nEDFm — Edge Detection Flag Register

This register has an edge detection flag and a flag clear bit of channel number m in DNF group number n.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	EDF	—	—	—	—	—	—	CLED
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 2.59 DNFP02nEDFm Register Contents

Bit Position	Bit Name	Function
7	EDF	This bit holds a result of edge detection based on the edge detection setting for channel number m. This bit is cleared by writing 1 to the CLED bit and continues to hold 1 unless it is cleared. Its state does not change while 1 is held, even if an edge is detected. 0: Valid edge has not been detected. 1: Valid edge has been detected.
6 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CLED	Writing 1 to this bit clears the status flag of EDF, bit 7. In case of contention between the writing of 1 and a setting source of bit 7, EDF clearing takes priority. Even writing 0 cannot change the state. The read value is always 0.

(9) Setting Flow of Edge Detection DNF

The following shows the procedures to set edge detection DNF.

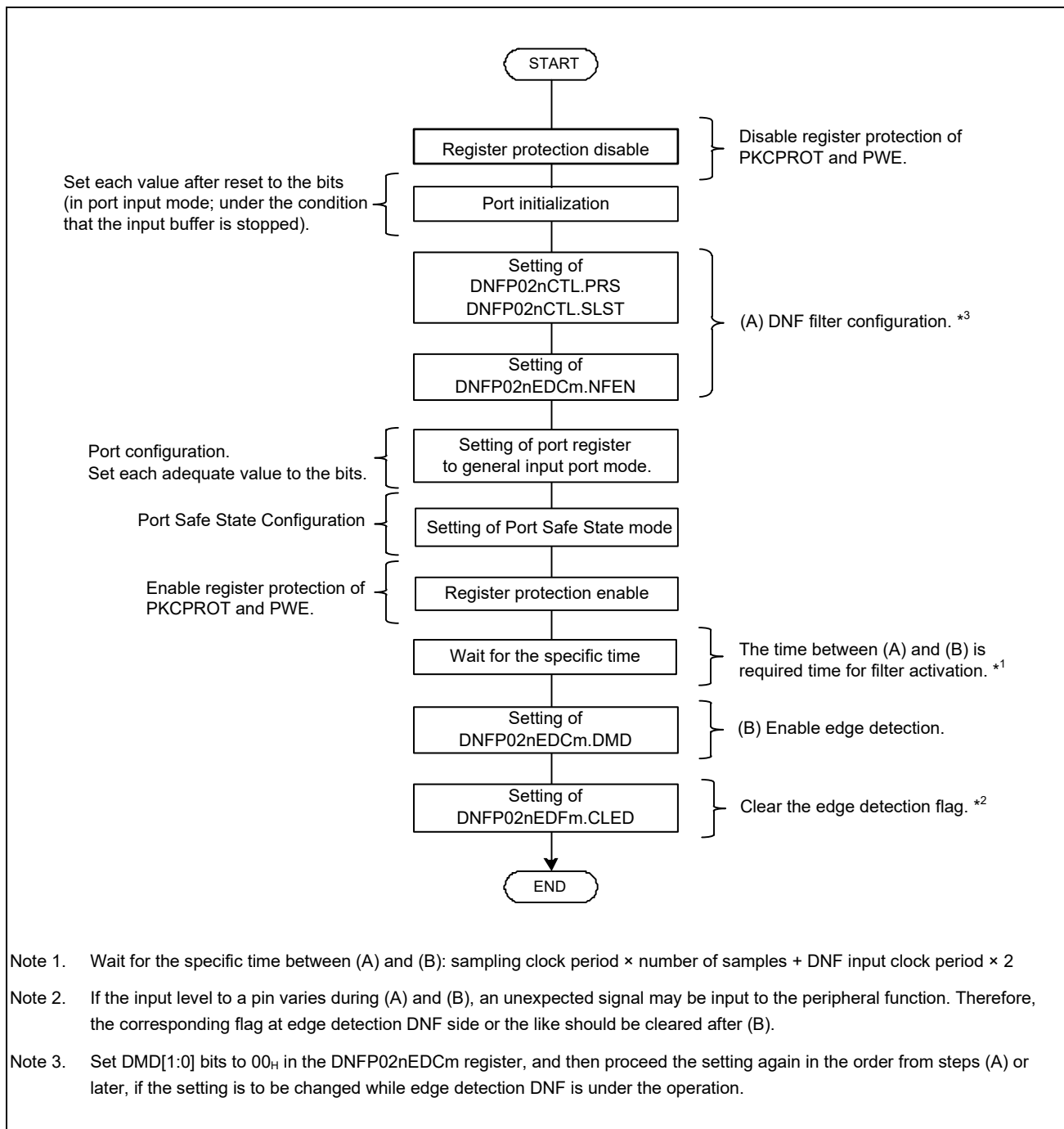


Figure 2.10 Edge Detection DNF Setting Flow

(10) List of Pins on Which an Edge Detection DNF Is Inserted

All port groups are the targets of edge detection DNF insertion. For details, see the following tables.

Table 2.60 DNF Insertion Target Pins (1/5)

Edge Detection DNF Group	DNF Channel Number m	Pin Name	E2x-FCC1		E2M	
			BGA373	BGA292 (E2M)	BGA373	BGA292
EDC00	0	P00_0	✓	✓	✓	✓
	1	P00_1	✓	✓	✓	✓
	2	P00_2	✓	✓	✓	✓
	3	P00_3	✓	✓	✓	✓
	4	P00_4	✓	✓	✓	✓
	5	P00_5	✓	✓	✓	✓
	6	P00_6	✓	✓	✓	✓
	7	P00_7	✓	✓	✓	✓
	8	P00_8	✓	—	✓	—
	9	P00_9	✓	—	✓	—
	10	P00_10	✓	—	✓	—
	11	P00_11	✓	—	✓	—
EDC01	3	P01_3	✓	✓	✓	✓
	4	P01_4	✓	✓	✓	✓
	5	P01_5	✓	✓	✓	✓
	6	P01_6	✓	✓	✓	✓
	7	P01_7	✓	✓	✓	✓
EDC02	0	P02_0	✓	✓	✓	✓
	1	P02_1	✓	✓	✓	✓
	2	P02_2	✓	✓	✓	✓
	3	P02_3	✓	✓	✓	✓
	4	P02_4	✓	✓	✓	✓
	5	P02_5	✓	✓	✓	✓
	6	P02_6	✓	✓	✓	✓
	7	P02_7	✓	✓	✓	✓
	8	P02_8	✓	✓	✓	✓
	9	P02_9	✓	✓	✓	✓
	10	P02_10	✓	✓	✓	✓
	11	P02_11	✓	✓	✓	✓
EDC10	0	P10_0	✓	✓	✓	✓
	1	P10_1	✓	✓	✓	✓
	2	P10_2	✓	✓	✓	✓
	3	P10_3	✓	✓	✓	✓
	4	P10_4	✓	✓	✓	✓
	5	P10_5	✓	✓	✓	✓
	6	P10_6	✓	✓	✓	✓
	7	P10_7	✓	✓	✓	✓
	8	P10_8	✓	✓	✓	✓
	9	P10_9	✓	—	✓	—
	10	P10_10	✓	—	✓	—

Table 2.60 DNF Insertion Target Pins (2/5)

Edge Detection DNF Group	DNF Channel Number m	Pin Name	E2x-FCC1		E2M	
			BGA373	BGA292 (E2M)	BGA373	BGA292
EDC10	11	P10_11	✓	—	✓	—
	12	P10_12	✓	—	✓	—
	13	P10_13	✓	—	✓	—
	14	P10_14	✓	—	✓	—
EDC11	0	P11_0	✓	✓	✓	✓
	1	P11_1	✓	✓	✓	✓
	2	P11_2	✓	✓	✓	✓
	3	P11_3	✓	✓	✓	✓
	4	P11_4	✓	✓	✓	✓
	5	P11_5	✓	✓	✓	✓
	6	P11_6	✓	✓	✓	✓
	7	P11_7	✓	✓	✓	✓
	8	P11_8	✓	✓	✓	✓
	9	P11_9	✓	✓	✓	✓
EDC12	0	P12_0	✓	✓	✓	✓
	1	P12_1	✓	✓	✓	✓
	2	P12_2	✓	✓	✓	✓
	3	P12_3	✓	✓	✓	✓
	4	P12_4	✓	✓	✓	✓
	5	P12_5	✓	✓	✓	✓
	6	P12_6	✓	✓	✓	✓
	7	P12_7	✓	✓	✓	✓
	8	P12_8	✓	✓	✓	✓
	9	P12_9	✓	✓	✓	✓
EDC13	0	P13_0	✓	✓	✓	✓
	1	P13_1	✓	✓	✓	✓
	2	P13_2	✓	✓	✓	✓
	3	P13_3	✓	✓	✓	✓
	4	P13_4	✓	—	✓	—
	5	P13_5	✓	—	✓	—
	6	P13_6	✓	—	✓	—
	7	P13_7	✓	—	✓	—
	8	P13_8	✓	—	✓	—
	9	P13_9	✓	—	✓	—
	10	P13_10	✓	—	✓	—
	11	P13_11	✓	—	✓	—
	12	P13_12	✓	—	✓	—
	13	P13_13	✓	—	✓	—
14	P13_14	✓	—	✓	—	
EDC14	0	P14_0	✓	✓	✓	✓
	1	P14_1	✓	✓	✓	✓
	2	P14_2	✓	✓	✓	✓

Table 2.60 DNF Insertion Target Pins (3/5)

Edge Detection DNF Group	DNF Channel Number m	Pin Name	E2x-FCC1		E2M		
			BGA373	BGA292 (E2M)	BGA373	BGA292	
EDC14	3	P14_3	✓	✓	✓	✓	
	4	P14_4	✓	✓	✓	✓	
	5	P14_5	✓	✓	✓	✓	
	6	P14_6	✓	✓	✓	✓	
	7	P14_7	✓	✓	✓	✓	
	8	P14_8	✓	—	✓	—	
	9	P14_9	✓	—	✓	—	
	10	P14_10	✓	—	✓	—	
	11	P14_11	✓	—	✓	—	
	12	P14_12	✓	—	✓	—	
	EDC15	0	P15_0	✓	✓	✓	✓
		1	P15_1	✓	✓	✓	✓
2		P15_2	✓	✓	✓	✓	
3		P15_3	✓	✓	✓	✓	
4		P15_4	✓	✓	✓	✓	
5		P15_5	✓	✓	✓	✓	
6		P15_6	✓	✓	✓	✓	
7		P15_7	✓	✓	✓	✓	
8		P15_8	✓	✓	✓	✓	
EDC20	0	P20_0	✓	✓	✓	✓	
	1	P20_1	✓	✓	✓	✓	
	2	P20_2	✓	✓	✓	✓	
	3	P20_3	✓	✓	✓	✓	
	4	P20_4	✓	✓	✓	✓	
	5	P20_5	✓	✓	✓	✓	
	6	P20_6	✓	✓	✓	✓	
	7	P20_7	✓	✓	✓	✓	
EDC21	2	P21_2	✓	✓	✓	✓	
	3	P21_3	✓	✓	✓	✓	
	4	P21_4	✓	✓	✓	✓	
	5	P21_5	✓	✓	✓	✓	
EDC22	0	P22_0	✓	✓	✓	✓	
	1	P22_1	✓	✓	✓	✓	
	2	P22_2	✓	✓	✓	✓	
	3	P22_3	✓	✓	✓	✓	
	4	P22_4	✓	✓	✓	✓	
	5	P22_5	✓	✓	✓	✓	
	6	P22_6	✓	✓	✓	✓	
	7	P22_7	✓	✓	✓	✓	
	8	P22_8	✓	✓	✓	✓	
	9	P22_9	✓	✓	✓	✓	
	10	P22_10	✓	✓	✓	✓	
	11	P22_11	✓	✓	✓	✓	

Table 2.60 DNF Insertion Target Pins (4/5)

Edge Detection DNF Group	DNF Channel Number m	Pin Name	E2x-FCC1		E2M	
			BGA373	BGA292 (E2M)	BGA373	BGA292
EDC22	12	P22_12	✓	✓	✓	✓
	13	P22_13	✓	✓	✓	✓
EDC23	0	P23_0	✓	✓	✓	✓
	1	P23_1	✓	✓	✓	✓
	2	P23_2	✓	✓	✓	✓
	3	P23_3	✓	✓	✓	✓
	4	P23_4	✓	✓	✓	✓
	5	P23_5	✓	✓	✓	✓
	6	P23_6	✓	✓	✓	✓
	7	P23_7	✓	✓	✓	✓
EDC24	0	P24_0	✓	—	✓	—
	1	P24_1	✓	—	✓	—
	2	P24_2	✓	—	✓	—
	3	P24_3	✓	—	✓	—
	4	P24_4	✓	—	✓	—
	5	P24_5	✓	—	✓	—
	6	P24_6	✓	—	✓	—
	7	P24_7	✓	—	✓	—
	8	P24_8	✓	—	✓	—
	9	P24_9	✓	—	✓	—
	10	P24_10	✓	—	✓	—
	11	P24_11	✓	—	✓	—
	12	P24_12	✓	—	✓	—
	13	P24_13	✓	—	✓	—
	14	P24_14	✓	—	✓	—
15	P24_15	✓	—	✓	—	
EDC25	0	P25_0	✓	—	✓	—
	1	P25_1	✓	—	✓	—
	2	P25_2	✓	—	✓	—
	3	P25_3	✓	—	✓	—
	4	P25_4	✓	—	✓	—
	5	P25_5	✓	—	✓	—
	6	P25_6	✓	—	✓	—
EDC27	0	P27_0	✓	✓	✓	✓
EDC30	0	P30_0	✓	—	✓	—
	1	P30_1	✓	—	✓	—
	2	P30_2	✓	—	✓	—
	3	P30_3	✓	—	✓	—
EDC32	0	P32_0	✓	✓	✓	✓
	1	P32_1	✓	✓	✓	✓
	2	P32_2	✓	✓	✓	✓
	3	P32_3	✓	✓	✓	✓
	4	P32_4	✓	✓	✓	✓

Table 2.60 DNF Insertion Target Pins (5/5)

Edge Detection DNF Group	DNF Channel Number m	Pin Name	E2x-FCC1		E2M	
			BGA373	BGA292 (E2M)	BGA373	BGA292
EDC32	5	P32_5	✓	✓	✓	✓
	6	P32_6	✓	✓	✓	✓
EDC33	0	P33_0	✓	✓	✓	✓
	1	P33_1	✓	✓	✓	✓
	2	P33_2	✓	✓	✓	✓
	3	P33_3	✓	✓	✓	✓
	4	P33_4	✓	✓	✓	✓
	5	P33_5	✓	✓	✓	✓
	6	P33_6	✓	✓	✓	✓
	7	P33_7	✓	✓	✓	✓
	8	P33_8	✓	✓	✓	✓
	9	P33_9	✓	✓	✓	✓
	10	P33_10	✓	✓	✓	✓
	11	P33_11	✓	✓	✓	✓
	12	P33_12	✓	✓	✓	✓
13	P33_13	✓	✓	✓	✓	
EDC34	0	P34_0	✓	✓	✓	✓
	1	P34_1	✓	✓	✓	✓
	2	P34_2	✓	✓	✓	✓
	3	P34_3	✓	✓	✓	✓
EDC40	4	P34_4	✓	✓	✓	✓
	0	P40_0	✓	✓	✓	✓
	1	P40_1	✓	✓	✓	✓
	2	P40_2	✓	✓	✓	✓
	3	P40_3	✓	✓	✓	✓
	4	P40_4	✓	✓	✓	✓
	5	P40_5	✓	✓	✓	✓
	6	P40_6	✓	✓	✓	✓
	7	P40_7	✓	✓	✓	✓
	8	P40_8	✓	✓	✓	✓
	9	P40_9	✓	✓	✓	✓
	10	P40_10	✓	✓	✓	✓
	11	P40_11	✓	✓	✓	✓
	12	P40_12	✓	✓	✓	✓
	13	P40_13	✓	✓	✓	✓
14	P40_14	✓	✓	✓	✓	
15	P40_15	✓	✓	✓	✓	

2.6.2 Port Safe State Mode

All Port Groups have a Port Safe State Function implemented. Each of these Port Groups has its own Port safe state enable bit that is located in the PSFSCn_m register. When Port safe state trigger signal named ERROROUTZ controlled by the Error Control Module becomes active, the port behavior is changed to the following states.

- Case of PSFSCn_m = 0

The output control still remains. (Port mode or alternative control mode.)

- Case of PSFSCn_m = 1

The output control is changed to Port Safe State mode. The Port register (Pn_m) is disconnected and switched to the Port Safe State Data register (PSFSDn_m), and the Port mode register (PMn_m) is switched to the Port Safe State Output Enable register (PSFSOEn_m) which takes over output enable control.

NOTES

1. Do not set PSFSOEn_m = 1 if the target port is used in input mode.
2. Do not set PSFSOEn_m = 1 and PSFSDn_m = 1 if the target port is used in LVDS output mode.
3. The source of Port safe state trigger signal is an error output signal named ERROROUTZ in the Error Control Module (ECM). It can be controlled by the ECMmESET and ECMmECLR registers. Port safe state mode must be set after the error signal is cleared by the ECMmECLR register. For details, refer to **Section 39.3.3, ECMmECLR — ECM Master/Checker Error Clear Trigger Register (m = M/C)**.
4. When the Port safe state function is enabled for dynamically toggled pins such as the serial communication, these pins may output unexpected signals as normal communication because the state of target pins is fixed as soon as the error has been detected. But in that case, this function can block abnormal output signals immediately by fixing these ports to Port safe state.

In case output level is set to High or Low by using the Port safe state function, the number of pins which can be applied to this function is restricted to 5 pins or less in each group.

(PSFSCn.PSFSCn_m = 1 and PSFSOEn.PSFSoEn_m = 1)

On the other hand, the number of these pins is not restricted if output level is set to Hi-z.

(PSFSCn.PSFSCn_m = 1 and PSFSOEn.PSFSoEn_m = 0)

For each pin group, refer to *Note 3 of Section 1.2.6 Output Voltage Characteristics* in *RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

2.6.2.1 Block Diagram

The following figure shows an overview of the Port Safe State Function.

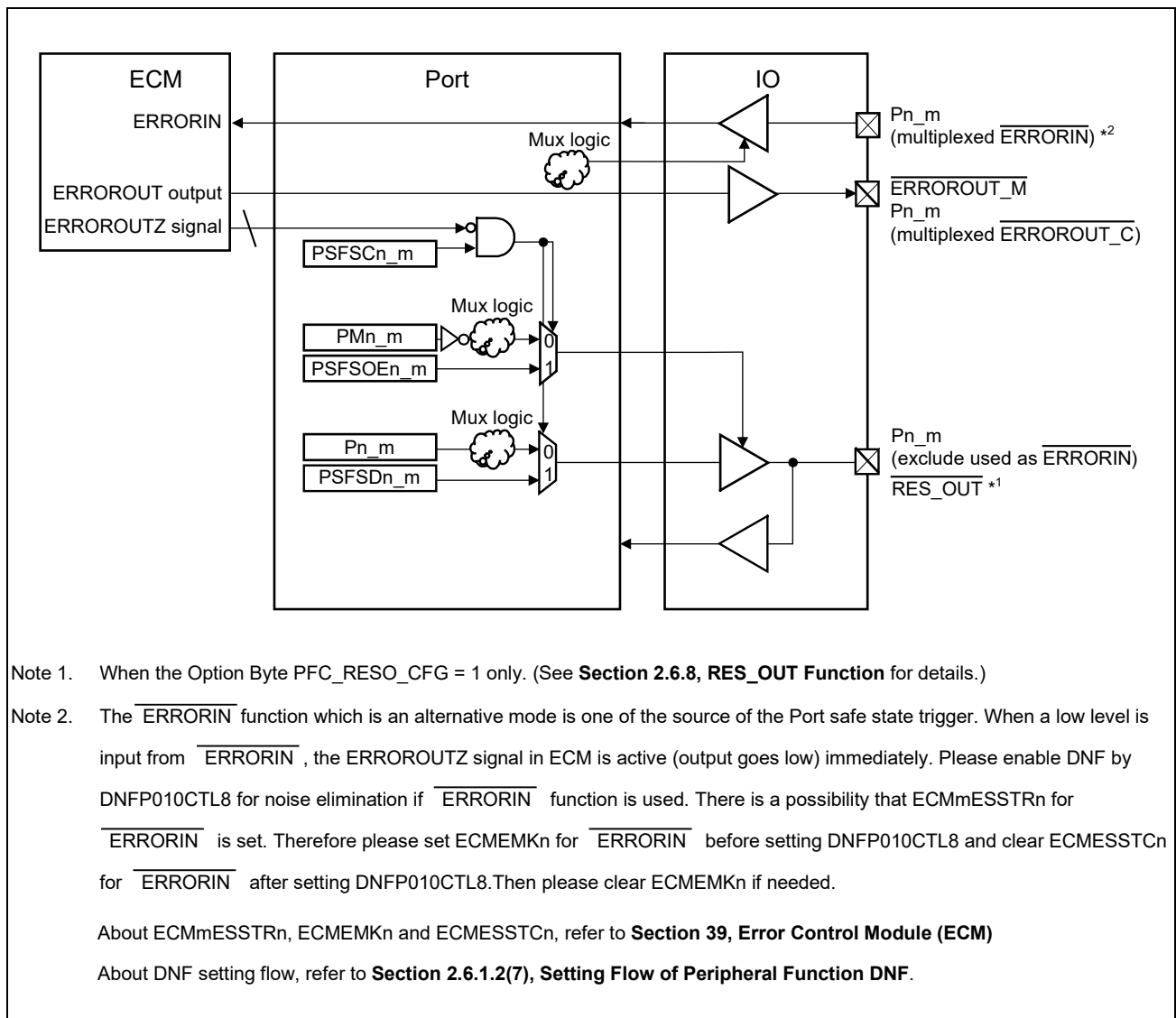


Figure 2.11 Block Diagram of Port Safe State

2.6.2.2 Supported Pins

All port groups are the target of the Port Safe State function. (See **Section 2.1.1, Port Group** for the supported port groups). The following table shows a list of pins not subject to the Port Safe State function.

Table 2.61 Pins Not Subject to Port Safe State Function

Category	Pin Name
Reference voltage pins of A/D converters	ADSVREFH, ADSVREFL, A0VREFH, A1VREFH* ¹ , A2VREFH, A3VREFH* ¹
Analog input pins* ²	ANxxx, CANxxxN, CANxxxP, DSANxxxN, DSANxxxP
Pins connected to capacitors	ADSVCL, RAMSVCL
External reset input pins	$\overline{\text{RES_IN}}$
Non-maskable input pins	NMI
Debug control pins	TCK, TDI, TDO, TMS, $\overline{\text{TRST}}$, $\overline{\text{DRDY}}$, $\overline{\text{AURÖRES}}$, CICLEFP, CICLEFN, TODP0, TODN0
Mode setting pins	MD0, MD1
Crystal pins	XTAL, EXTAL
Error output pins	$\overline{\text{ERROROUT_M}}$ * ³

Note 1. These power supply pins are not implemented in BGA292.

Note 2. Some analog input pins are not implemented in BGA292. Refer to the **Section 2.6.6, Dedicated Analog Pin Table** for details.

Note 3. When Port safe state trigger signal becomes active, the status of $\overline{\text{ERROROUT_M}}$ will be changed to Low output as the Port safe state depends on the ECM setting. Refer to **Section 39, Error Control Module (ECM)** for details.

2.6.2.3 Configuration Flow

The following shows the flow to use Port safe state mode.

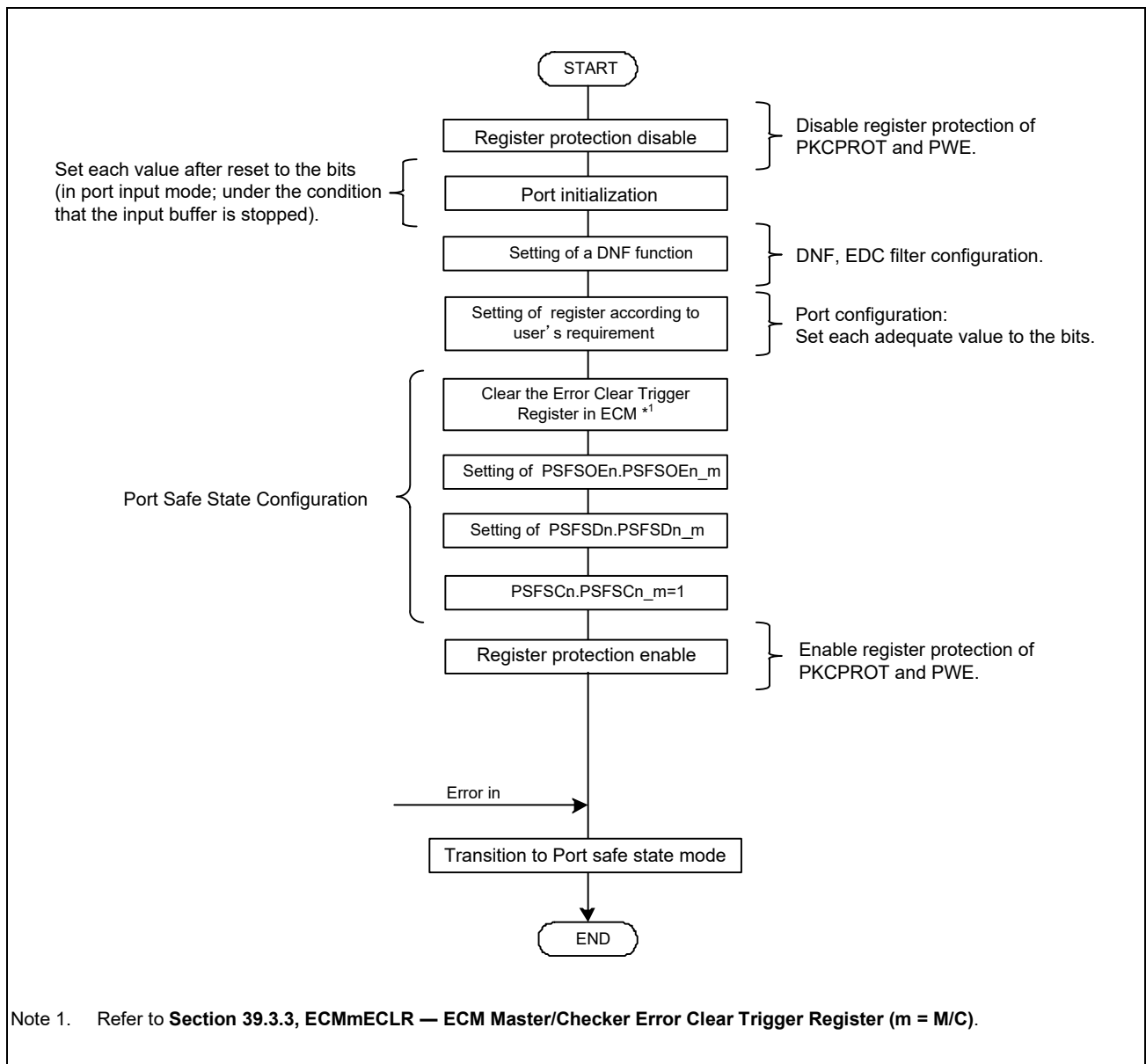


Figure 2.12 Setting Flow of Port Safe State

2.6.3 Virtual Port Function via RHSB (XBAR)

Virtual ports are port groups that are not connected to external pins but are instead connected to internal functions. Port groups P50, P51, P52 and P53 are virtual ports for the XBAR function of RHSB. Configurable output signals are 64 bits in total (each port group has 16 bits x 4 groups), and it is possible to transfer these data serially to other devices via XBAR in the same way as with a port register setting (XBAR function, see **Section 22.6, Cross Bar (XBAR)**).

- The output data (64 bits) of P50, P51, P52 and P53 is transferring to input of XBAR function.
- Output pins of P50, P51, P52 and P53 are connected to XBAR function instead of IO output buffer.
- The data from XBAR function is transferred to RHSB.

2.6.3.1 Block Diagram

The following figure shows the use of coordination with XBAR Function.

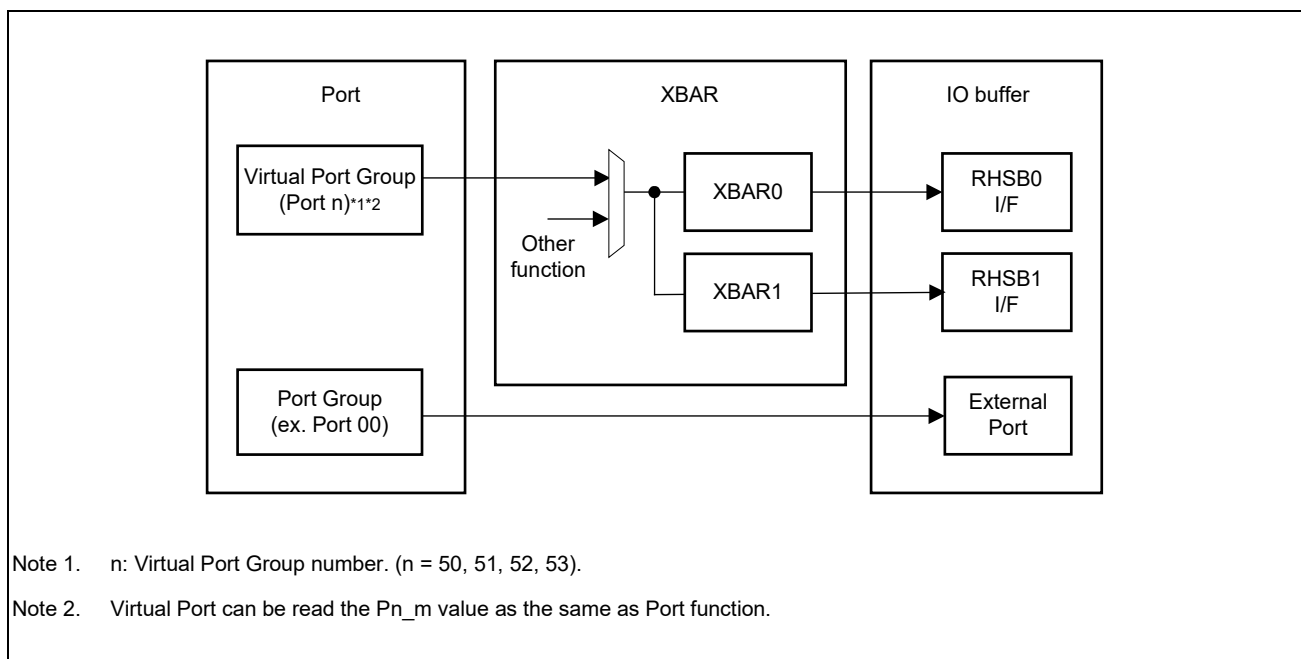


Figure 2.13 Block Diagram of Virtual Port Function via RHSB

2.6.4 Virtual Port Function via HSSPI

Virtual ports are port groups that are not connected to external pins but are instead connected to internal functions. Port groups P54, P55, P56 and P57 are virtual port groups for HSSPI I/F. Their function is to transfer the received serial data from HSSPI I/F to GTM/ATU, as the inverted function of delivering the serial data to XBAR described in **Section 2.6.3, Virtual Port Function via RHSB (XBAR)**.

About the HSSPI function, see **Section 26, High Speed Serial Peripheral Interface (HS-SPI)**.

- The data (max 64 bits) from HSSPI is transferred to the input pins of P54, P55, P56 and P57.
- The input pins of P54, P55, P56 and P57 are connected to HSSPI instead of the IO input buffer.
- The data from HSSPI is transferred to GTM/ATU via the port function (pin multiplexed table, see **Section 2.5.2, List of Alternative Functions**).
- The 64 bits of data from HSSPI can be divided into 8 bits per frame.
- For example, the received data is $8*n$ bits ($n = 1, 2, 3, \dots, 8$)

2.6.4.1 Block Diagram

The following figure shows the use of P54, P55, P56 and P57. It may look like the data for GTM/ATU is from P54, P55, P56 and P57, but actually, the data is from HSSPI.

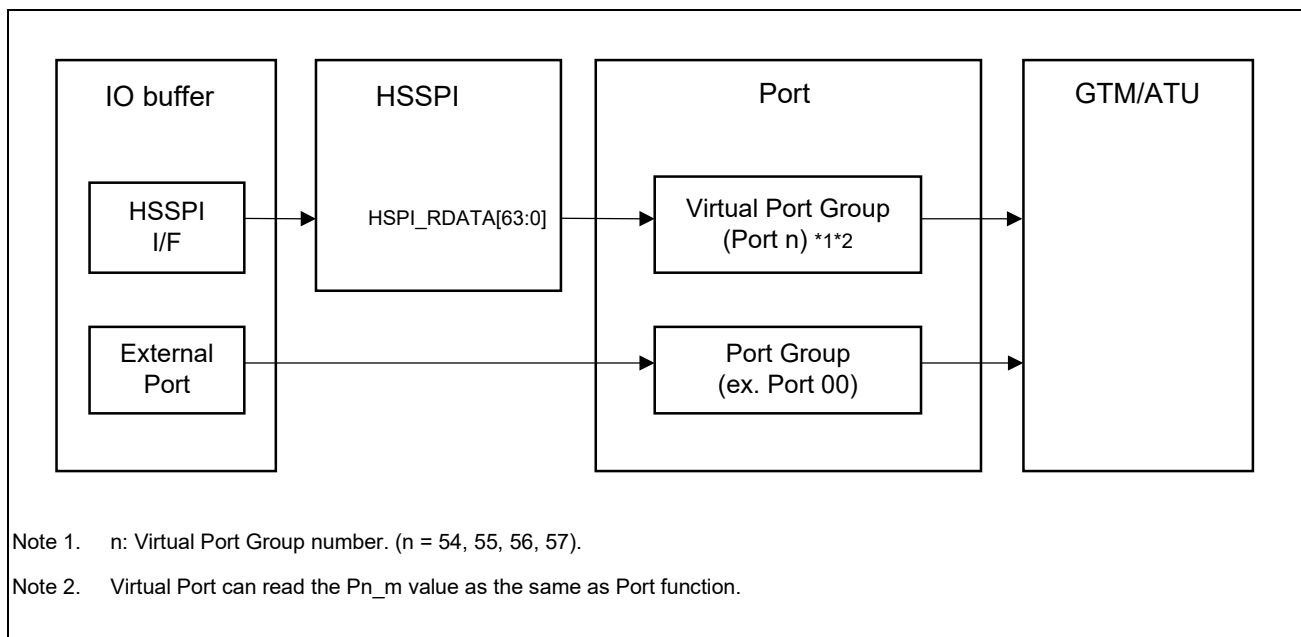


Figure 2.14 Block Diagram of Virtual Port Function via HSSPI

2.6.5 Virtual Port Function via Analog

Virtual ports are port groups that are not connected to external pins but are instead connected to internal functions. The target port groups are P41, P42 and P43 for E2x-FCC1 and E2M. This function is to transfer the received serial data from Analog Boundary Flag generator to GTM/ATU, as the inverted function of delivering the serial data to XBAR described in **Section 2.6.3, Virtual Port Function via RHSB (XBAR)**.

About Analog Boundary Flag generator, see **Section 34, Analog to Digital Converter (ADCH)**, **Section 35, Delta-Sigma Analog to Digital Converter (DSADC)** and **Section 36, Cyclic Analog to Digital Converter (CADC)**.

- The data (max 40 bits for E2x-FCC1 and E2M.) from Analog Boundary Flag generator is transferring to input pins of P41, P42 and P43.
- Input pins of virtual ports are connected to Analog Boundary Flag generator instead of IO input buffer.
- The data from Analog Boundary Flag generator is transferred to GTM/ATU via port function (pin multiplexed table, see **Section 2.5.2, List of Alternative Functions**).

2.6.5.1 Block Diagram

The following figure shows the use of virtual ports. It may look like the data for GTM/ATU is from a virtual port, but actually, the data is from the analog module.

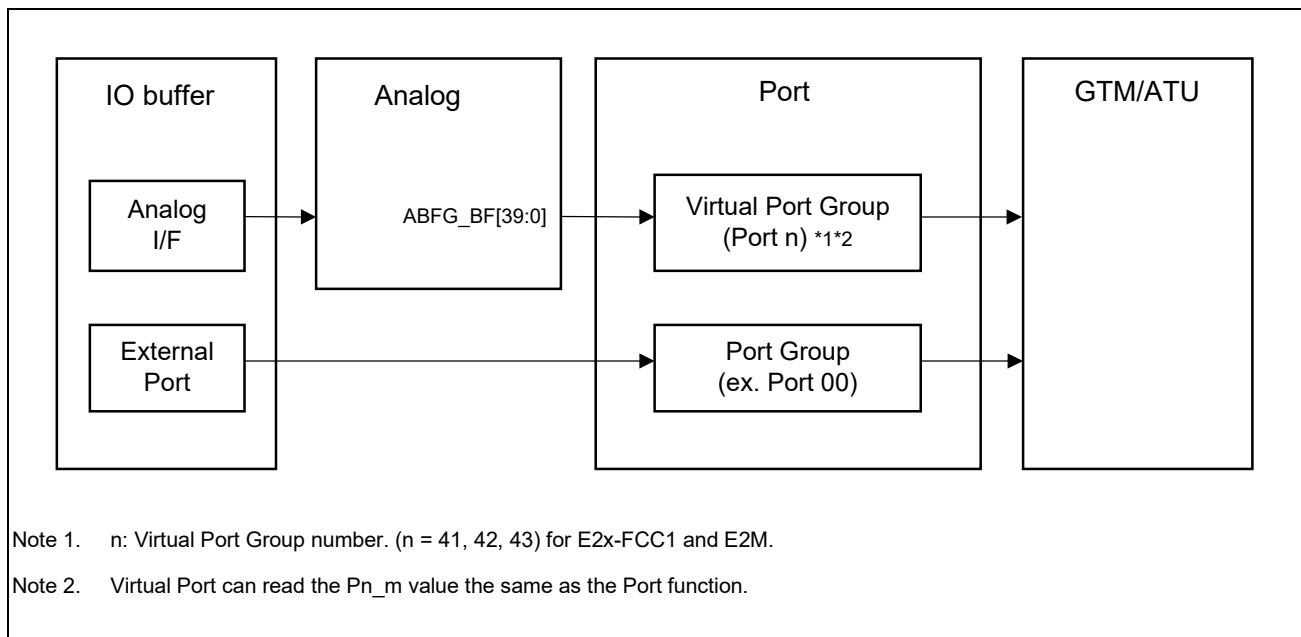


Figure 2.15 Block Diagram of Virtual Port Function via Analog

2.6.6 Dedicated Analog Pin Table

The following tables list the dedicated analog pin in each product.

Table 2.62 Dedicated Analog Pin (1/4)

Pin	Special Func (Analog)	E2x-FCC1		E2M	
		BGA373	BGA292 (E2M)	BGA373	BGA292
AN000	AN000	✓	✓	✓	✓
	DSAN001N	✓	✓	✓	✓
AN001	AN001	✓	✓	✓	✓
	DSAN000N	✓	✓	✓	✓
AN002	AN002	✓	✓	✓	✓
	DSAN001P	✓	✓	✓	✓
AN003	AN003	✓	✓	✓	✓
	DSAN000P	✓	✓	✓	✓
AN010	AN010	✓	✓	✓	✓
	DSAN002P	✓	✓	✓	✓
AN011	AN011	✓	✓	✓	✓
	DSAN002N	✓	✓	✓	✓
AN012	AN012	✓	✓	✓	✓
	DSAN003P	✓	✓	✓	✓
AN013	AN013	✓	✓	✓	✓
	DSAN003N	✓	✓	✓	✓
AN020	AN020	✓	✓	✓	✓
	DSAN110N	✓	✓	✓	✓
AN021	AN021	✓	✓	✓	✓
	DSAN110P	✓	✓	✓	✓
AN022	AN022	✓	✓	✓	✓
	DSAN111P	✓	✓	✓	✓
AN023	AN023	✓	✓	✓	✓
	DSAN130N	✓	✓	✓	✓
AN030	AN030	✓	✓	✓	✓
	DSAN130P	✓	✓	✓	✓
AN031	AN031	✓	✓	✓	✓
	DSAN111N	✓	✓	✓	✓
AN032	AN032	✓	✓	✓	✓
	DSAN131P	✓	✓	✓	✓
AN033	AN033	✓	✓	✓	✓
	DSAN131N	✓	✓	✓	✓
AN040	AN040	✓	✓	✓	✓
AN041	AN041	✓	✓	✓	✓
AN042	AN042	✓	✓	✓	✓
AN043	AN043	✓	✓	✓	✓
AN050	AN050	✓	✓	✓	✓
	AN143	✓	—	✓	—
AN051	AN051	✓	✓	✓	✓
	AN142	✓	—	✓	—

Table 2.62 Dedicated Analog Pin (2/4)

Pin	Special Func (Analog)	E2x-FCC1		E2M	
		BGA373	BGA292 (E2M)	BGA373	BGA292
AN052	AN052	✓	✓	✓	✓
	AN141	✓	—	✓	—
AN053	AN053	✓	✓	✓	✓
	AN140	✓	—	✓	—
AN060	AN060	✓	✓	✓	✓
	AN133	✓	—	✓	—
AN061	AN061	✓	✓	✓	✓
	AN132	✓	—	✓	—
AN062	AN062	✓	✓	✓	✓
	AN131	✓	—	✓	—
AN063	AN063	✓	✓	✓	✓
	AN130	✓	—	✓	—
AN100	AN100	✓	—	✓	—
AN101	AN101	✓	—	✓	—
AN102	AN102	✓	—	✓	—
AN103	AN103	✓	—	✓	—
AN110	AN110	✓	—	✓	—
AN111	AN111	✓	—	✓	—
AN112	AN112	✓	—	✓	—
AN113	AN113	✓	—	✓	—
AN120	AN120	✓	—	✓	—
AN121	AN121	✓	—	✓	—
AN122	AN122	✓	—	✓	—
AN123	AN123	✓	—	✓	—
AN200	AN200	✓	✓	✓	✓
	CAN000P	✓	✓	✓	✓
AN201	AN201	✓	✓	✓	✓
	CAN000N	✓	✓	✓	✓
AN202	AN202	✓	✓	✓	✓
	CAN001P	✓	✓	✓	✓
AN203	AN203	✓	✓	✓	✓
	CAN001N	✓	✓	✓	✓
AN210	AN210	✓	✓	✓	✓
	CAN003P	✓	✓	✓	✓
AN211	AN211	✓	✓	✓	✓
	CAN002N	✓	✓	✓	✓
AN212	AN212	✓	✓	✓	✓
	CAN002P	✓	✓	✓	✓
AN213	AN213	✓	✓	✓	✓
	DSAN200P	✓	✓	✓	✓
AN220	AN220	✓	✓	✓	✓
	CAN003N	✓	✓	✓	✓

Table 2.62 Dedicated Analog Pin (3/4)

Pin	Special Func (Analog)	E2x-FCC1		E2M	
		BGA373	BGA292 (E2M)	BGA373	BGA292
AN221	AN221	✓	✓	✓	✓
	DSAN101N	✓	✓	✓	✓
AN222	AN222	✓	✓	✓	✓
	DSAN200N	✓	✓	✓	✓
AN223	AN223	✓	✓	✓	✓
	DSAN100P	✓	✓	✓	✓
AN230	AN230	✓	✓	✓	✓
	DSAN121N	✓	✓	✓	✓
AN231	AN231	✓	✓	✓	✓
	DSAN101P	✓	✓	✓	✓
AN232	AN232	✓	✓	✓	✓
	DSAN100N	✓	✓	✓	✓
AN233	AN233	✓	✓	✓	✓
	DSAN120N	✓	✓	✓	✓
AN240	AN240	✓	✓	✓	✓
	DSAN121P	✓	✓	✓	✓
AN241	AN241	✓	✓	✓	✓
	DSAN120P	✓	✓	✓	✓
AN242	AN242	✓	✓	✓	✓
AN243	AN243	✓	✓	✓	✓
AN250	AN250	✓	✓	✓	✓
	AN343	✓	—	✓	—
AN251	AN251	✓	✓	✓	✓
	AN342	✓	—	✓	—
AN252	AN252	✓	✓	✓	✓
	AN341	✓	—	✓	—
AN253	AN253	✓	✓	✓	✓
	AN340	✓	—	✓	—
AN260	AN260	✓	—	✓	—
	AN333	✓	—	✓	—
AN261	AN261	✓	—	✓	—
	AN332	✓	—	✓	—
AN262	AN262	✓	—	✓	—
	AN331	✓	—	✓	—
AN263	AN263	✓	—	✓	—
	AN330	✓	—	✓	—
AN300	AN300	✓	—	✓	—
AN301	AN301	✓	—	✓	—
AN302	AN302	✓	—	✓	—
AN303	AN303	✓	—	✓	—
AN310	AN310	✓	—	✓	—
AN311	AN311	✓	—	✓	—
AN312	AN312	✓	—	✓	—

Table 2.62 Dedicated Analog Pin (4/4)

Pin	Special Func (Analog)	E2x-FCC1		E2M	
		BGA373	BGA292 (E2M)	BGA373	BGA292
AN313	AN313	✓	—	✓	—
AN320	AN320	✓	—	✓	—
AN321	AN321	✓	—	✓	—
AN322	AN322	✓	—	✓	—
AN323	AN323	✓	—	✓	—

2.6.7 ICUM Select Function

For the ICUM select function, ports are controlled by ICUM even if the port is in alternative mode or Port safe state mode. Then no matter what function is selected for the port, the ICUM selection function is always active. The control from ICUM is asynchronous to the clock bus. The priority of the ICUM select function is higher than GPIO control and IP control, but lower than reset.

ICUM select function target pin: P22_11.

For information on how to use the ICUM function, refer to the separate User Manual: *Intelligent Cryptographic Unit*.

2.6.7.1 Block Diagram

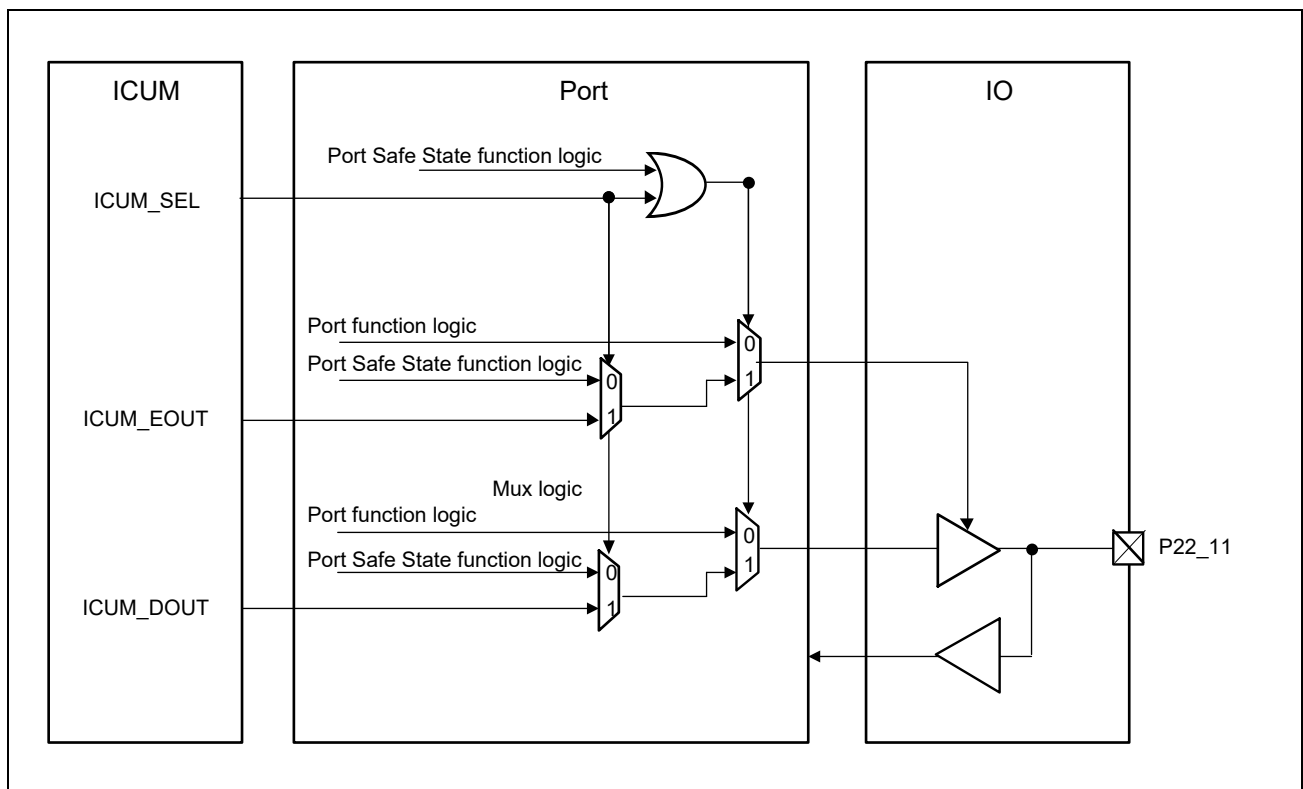


Figure 2.16 Block Diagram of ICUM Function

2.6.8 RES_OUT Function

2.6.8.1 Overview

This product has a $\overline{\text{RES_OUT}}$ pin to handshake with a debugger. The behavior of this signal's output can be selected by using an Option Byte as follows. For detailed information about Option Bytes, refer to the **Section 42, Basic Hardware Protection (BHP)**.

(1) Option Byte PFC_RESO_CFG = 1

- $\overline{\text{RES_OUT}}$ pin is assign P27 group (P27_0 only) supported with output in order to handshake to debugger only.
- This port setting is configurable as a high/low output with push-pull and open drain modes that are the same as other port. However, all of the P27 registers are protected by a key code register.
- The output signal is low when reset factor (see **Section 2.1.5, Reset Sources** for details) is asserted. It will hold the low output until port setting is changed by user SW command.
- This pin shows the internal reset state, so can be used as handshake for external devices like debugger.

The following figure shows the use of $\overline{\text{RES_OUT}}$ function.

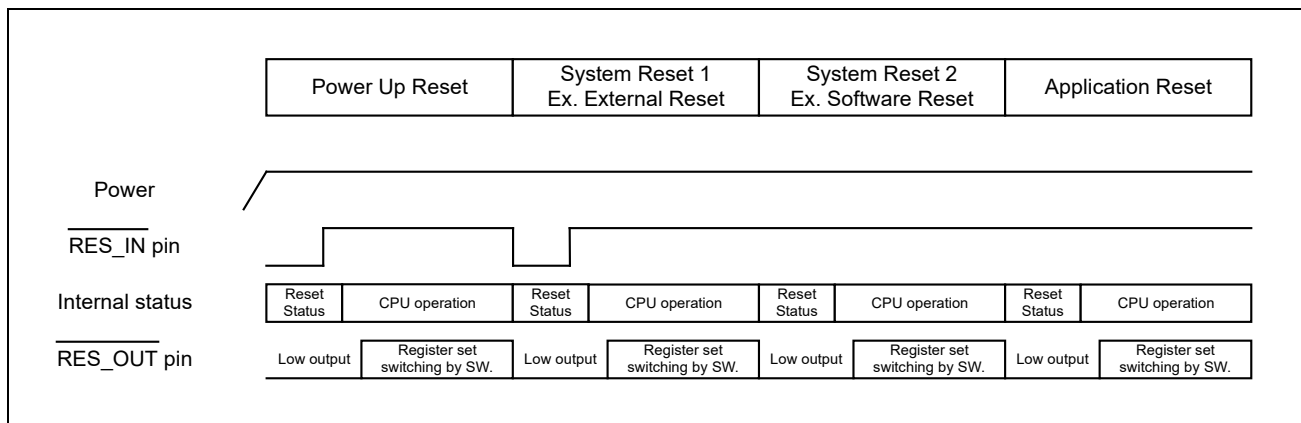


Figure 2.17 Timing Chart for $\overline{\text{RES_OUT}}$ Function for Each Reset Factor (PFC_RESO_CFG = 1).

(2) Option Byte PFC_RESO_CFG = 0

- $\overline{\text{RES_OUT}}$ pin is not assigned to a port group, so the port setting is not configurable.
- The output signal is low when reset factor (see **Section 2.1.5, Reset Sources** for details.) is asserted. And the terminal state is automatically changed from low to high after internal reset.

The following figure shows the use of $\overline{\text{RES_OUT}}$ function.

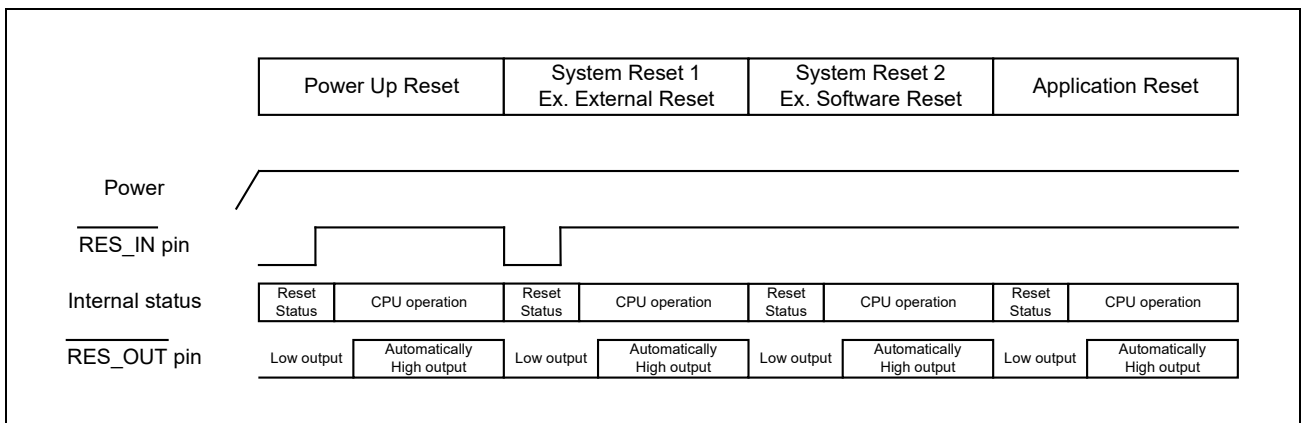


Figure 2.18 Timing Chart for $\overline{\text{RES_OUT}}$ Function for Each Reset Factor (PFC_RESO_CFG = 0).

2.7 List of Pin Function

2.7.1 Overview

Table 2.63 list the functions of each pin. “✓” indicates that the pin support the relevant function and “—” indicates that it does not. Do not use pins that do not support the relevant functions.

2.7.2 Details Function

Table 2.63 Pin Function (1/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
ATU-V (common)	TCLKA	I	External clock input	✓	✓	✓	✓
	TCLKB	I		✓	✓	✓	✓
ATU-V (Timer A)	TIA00	I	Input capture trigger to each channel of timer A	✓	✓	✓	✓
	TIA01	I		✓	✓	✓	✓
	TIA02	I		✓	✓	✓	✓
	TIA03	I		✓	✓	✓	✓
	TIA04	I		✓	✓	✓	✓
	TIA05	I		✓	✓	✓	✓
	TIA06	I		✓	✓	✓	✓
	TIA07	I		✓	✓	✓	✓
ATU-V (Timer C)	TIOC00	IO	Input capture trigger and output compare output to timer C	✓	✓	✓	✓
	TIOC01	IO		✓	✓	✓	✓
	TIOC02	IO		✓	✓	✓	✓
	TIOC03	IO		✓	✓	✓	✓
	TIOC10	IO		✓	✓	✓	✓
	TIOC11	IO		✓	✓	✓	✓
	TIOC12	IO		✓	✓	✓	✓
	TIOC13	IO		✓	✓	✓	✓
	TIOC20	IO		✓	✓	✓	✓
	TIOC21	IO		✓	✓	✓	✓
	TIOC22	IO		✓	✓	✓	✓
	TIOC23	IO		✓	✓	✓	✓
	TIOC30	IO		✓	✓	✓	✓
	TIOC31	IO		✓	✓	✓	✓
	TIOC32	IO		✓	✓	✓	✓
	TIOC33	IO		✓	✓	✓	✓
	TIOC40	IO		✓	✓	✓	✓
	TIOC41	IO		✓	✓	✓	✓
	TIOC42	IO		✓	✓	✓	✓
	TIOC43	IO		✓	✓	✓	✓
	TIOC50	IO		✓	✓	✓	✓
	TIOC51	IO		✓	✓	✓	✓
	TIOC52	IO		✓	✓	✓	✓
	TIOC53	IO		✓	✓	✓	✓
	TIOC60	IO		✓	✓	✓	✓
TIOC61	IO	✓	✓	✓	✓		
TIOC62	IO	✓	✓	✓	✓		
TIOC63	IO	✓	✓	✓	✓		
TIOC70	IO	✓	✓	✓	✓		
TIOC71	IO	✓	✓	✓	✓		
TIOC72	IO	✓	✓	✓	✓		
TIOC73	IO	✓	✓	✓	✓		

Table 2.63 Pin Function (2/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
ATU-V (Timer C)	TIOC80	IO	Input capture trigger and output compare output to timer C	✓	✓	✓	✓
	TIOC81	IO		✓	✓	✓	✓
	TIOC82	IO		✓	✓	✓	✓
	TIOC83	IO		✓	✓	✓	✓
	TIOC90	IO		✓	✓	✓	✓
	TIOC91	IO		✓	✓	✓	✓
	TIOC92	IO		✓	✓	✓	✓
	TIOC93	IO		✓	✓	✓	✓
	TIOCA0	IO		✓*1	✓*1	✓*1	✓*1
	TIOCA1	IO		✓*1	✓*1	✓*1	✓*1
	TIOCA2	IO		✓*1	✓*1	✓*1	✓*1
	TIOCA3	IO		✓*1	✓*1	✓*1	✓*1
ATU-V (Timer D)	TOD00A	IO	One-shot pulse output of timer D	✓	✓	✓	✓
	TOD01A	O		✓	✓	✓	✓
	TOD02A	O		✓	✓	✓	✓
	TOD03A	O		✓	✓	✓	✓
	TOD10A	O		✓	✓	✓	✓
	TOD11A	O		✓	✓	✓	✓
	TOD12A	O		✓	✓	✓	✓
	TOD13A	O		✓	✓	✓	✓
	TOD20A	O		✓	✓	✓	✓
	TOD21A	O		✓	✓	✓	✓
	TOD22A	O		✓	✓	✓	✓
	TOD23A	O		✓	✓	✓	✓
	TOD30A	O		✓	✓	✓	✓
	TOD31A	O		✓	✓	✓	✓
	TOD32A	O		✓	✓	✓	✓
	TOD33A	O		✓	✓	✓	✓
	TOD40A	O		✓	✓	✓	✓
	TOD41A	O		✓	✓	✓	✓
	TOD42A	O		✓	✓	✓	✓
	TOD43A	O		✓	✓	✓	✓
	TOD50A	O		✓	✓	✓	✓
	TOD51A	O		✓	✓	✓	✓
	TOD52A	O		✓	✓	✓	✓
	TOD53A	O		✓	✓	✓	✓
	TOD60A	O		✓	✓	✓	✓
	TOD61A	O		✓	✓	✓	✓
TOD62A	O	✓	✓	✓	✓		
TOD63A	O	✓	✓	✓	✓		
TOD70A	O	✓	✓	✓	✓		
TOD71A	O	✓	✓	✓	✓		
TOD72A	O	✓	✓	✓	✓		

Table 2.63 Pin Function (3/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
ATU-V (Timer D)	TOD73A	O	One-shot pulse output of timer D	✓	✓	✓	✓
	TOD80A	O		✓	✓	✓	✓
	TOD81A	O		✓	✓	✓	✓
	TOD82A	O		✓	✓	✓	✓
	TOD83A	O		✓	✓	✓	✓
	TOD00B	O		✓	✓	✓	✓
	TOD01B	O		✓	✓	✓	✓
	TOD02B	O		✓	✓	✓	✓
	TOD03B	O		✓	✓	✓	✓
	TOD10B	O		✓	✓	✓	✓
	TOD11B	O		✓	✓	✓	✓
	TOD12B	O		✓	✓	✓	✓
	TOD13B	O		✓	✓	✓	✓
	TOD20B	O		✓	✓	✓	✓
	TOD21B	O		✓	✓	✓	✓
	TOD22B	O		✓	✓	✓	✓
	TOD23B	O		✓	✓	✓	✓
	TOD30B	O		✓	✓	✓	✓
	TOD31B	O		✓	✓	✓	✓
	TOD32B	O		✓	✓	✓	✓
	TOD33B	O		✓	✓	✓	✓
	TOD40B	O		✓	✓	✓	✓
	TOD41B	O		✓	✓	✓	✓
	TOD42B	O		✓	✓	✓	✓
	TOD43B	O		✓	✓	✓	✓
	TOD50B	O		✓	✓	✓	✓
	TOD51B	O		✓	✓	✓	✓
	TOD52B	O		✓	✓	✓	✓
	TOD53B	O		✓	✓	✓	✓
	TOD60B	O		✓	✓	✓	✓
	TOD61B	O		✓	✓	✓	✓
	TOD62B	O		✓	✓	✓	✓
	TOD63B	O		✓	✓	✓	✓
TOD70B	O	✓	✓	✓	✓		
TOD71B	O	✓	✓	✓	✓		
TOD72B	O	✓	✓	✓	✓		
TOD73B	O	✓	✓	✓	✓		
TOD80B	O	✓	✓	✓	✓		
TOD81B	O	✓	✓	✓	✓		
TOD82B	O	✓	✓	✓	✓		
TOD83B	O	✓	✓	✓	✓		

Table 2.63 Pin Function (4/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
ATU-V (Timer E)	TOE00	O	PWM output of timer E	✓	✓	✓	✓
	TOE01	O		✓	✓	✓	✓
	TOE02	O		✓	✓	✓	✓
	TOE03	O		✓	✓	✓	✓
	TOE10	O		✓	✓	✓	✓
	TOE11	O		✓	✓	✓	✓
	TOE12	O		✓	✓	✓	✓
	TOE13	O		✓	✓	✓	✓
	TOE20	O		✓	✓	✓	✓
	TOE21	O		✓	✓	✓	✓
	TOE22	O		✓	✓	✓	✓
	TOE23	O		✓	✓	✓	✓
	TOE30	O		✓	✓	✓	✓
	TOE31	O		✓	✓	✓	✓
	TOE32	O		✓	✓	✓	✓
	TOE33	O		✓	✓	✓	✓
	TOE40	O		✓	✓	✓	✓
	TOE41	O		✓	✓	✓	✓
	TOE42	O		✓	✓	✓	✓
	TOE43	O		✓	✓	✓	✓
	TOE50	O		✓	✓	✓	✓
	TOE51	O		✓	✓	✓	✓
	TOE52	O		✓	✓	✓	✓
	TOE53	O		✓	✓	✓	✓
	TOE60	O		✓	✓	✓	✓
	TOE61	O		✓	✓	✓	✓
	TOE62	O		✓	✓	✓	✓
	TOE63	O		✓	✓	✓	✓
	TOE70	O		✓	✓	✓	✓
	TOE71	O		✓	✓	✓	✓
TOE72	O	✓	✓	✓	✓		
TOE73	O	✓	✓	✓	✓		
TOE80	O	✓	✓	✓	✓		
TOE81	O	✓	✓	✓	✓		
TOE82	O	✓	✓	✓	✓		
TOE83	O	✓	✓	✓	✓		

Table 2.63 Pin Function (5/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
ATU-V (Timer F)	TIF0A	I	Event input signal to timer F	✓	✓	✓	✓
	TIF1A	I		✓	✓	✓	✓
	TIF2A	I		✓	✓	✓	✓
	TIF0B	I		✓	✓	✓	✓
	TIF1B	I		✓	✓	✓	✓
	TIF2B	I		✓	✓	✓	✓
	TIF3	I		✓	✓	✓	✓
	TIF4	I		✓	✓	✓	✓
	TIF5	I		✓	✓	✓	✓
	TIF6	I		✓	✓	✓	✓
	TIF7	I		✓	✓	✓	✓
	TIF8	I		✓	✓	✓	✓
	TIF9	I		✓	✓	✓	✓
	TIF10	I		✓	✓	✓	✓
	TIF11	I		✓	✓	✓	✓
	TIF12	I		✓	✓	✓	✓
TIF13	I	✓	✓	✓	✓		
TIF14	I	✓	✓	✓	✓		
TIF15	I	✓	✓	✓	✓		
GTM	TIM0_0	I	Input signal to GTM (TIM)	✓	✓	✓	✓
	TIM0_1	I		✓	✓	✓	✓
	TIM0_2	I		✓	✓	✓	✓
	TIM0_3	I		✓	✓	✓	✓
	TIM0_4	I		✓	✓	✓	✓
	TIM0_5	I		✓	✓	✓	✓
	TIM0_6	I		✓	✓	✓	✓
	TIM0_7	I		✓	✓	✓	✓
	TIM1_0	I		✓	✓	✓	✓
	TIM1_1	I		✓	✓	✓	✓
	TIM1_2	I		✓	✓	✓	✓
	TIM1_3	I		✓	✓	✓	✓
	TIM1_4	I		✓	✓	✓	✓
	TIM1_5	I		✓	✓	✓	✓
	TIM1_6	I		✓	✓	✓	✓
	TIM1_7	I		✓	✓	✓	✓
	TIM2_0	I		✓	✓	✓	✓
	TIM2_1	I		✓	✓	✓	✓
	TIM2_2	I		✓	✓	✓	✓
	TIM2_3	I		✓	✓	✓	✓
TIM2_4	I	✓	✓	✓	✓		
TIM2_5	I	✓	✓	✓	✓		
TIM2_6	I	✓	✓	✓	✓		
TIM2_7	I	✓	✓	✓	✓		

Table 2.63 Pin Function (6/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
GTM	TIM3_0	I	Input signal to GTM (TIM)	✓	✓	✓	✓
	TIM3_1	I		✓	✓	✓	✓
	TIM3_2	I		✓	✓	✓	✓
	TIM3_3	I		✓	✓	✓	✓
	TIM3_4	I		✓	✓	✓	✓
	TIM3_5	I		✓	✓	✓	✓
	TIM3_6	I		✓	✓	✓	✓
	TIM3_7	I		✓	✓	✓	✓
	TIM4_0	I		✓	✓*1	✓	—
	TIM4_1	I		✓	✓*1	✓	—
	TIM4_2	I		✓	✓*1	✓	—
	TIM4_3	I		✓	✓*1	✓	—
	TIM4_4	I		✓	✓*1	✓	—
	TIM4_5	I		✓	✓*1	✓	—
	TIM4_6	I		✓	✓*1	✓	—
	TIM4_7	I		✓	✓*1	✓	—
	TIM5_0	I		✓	✓	✓	✓
	TIM5_1	I		✓	✓	✓	✓
	TIM5_2	I		✓	✓	✓	✓
	TIM5_3	I		✓	✓	✓	✓
	TIM5_4	I		✓	✓	✓	✓
	TIM5_5	I		✓	✓	✓	✓
	TIM5_6	I		✓	✓	✓	✓
	TIM5_7	I		✓	✓	✓	✓
	TIM6_0	I		✓	✓	✓	✓
	TIM6_1	I		✓	✓*1	✓	—
	TIM6_2	I		✓	✓*1	✓	—
	TIM6_3	I		✓	✓*1	✓	—
	TIM6_4	I		✓	✓*1	✓	—
	TIM6_5	I		✓	✓*1	✓	—
	TIM6_6	I		✓	✓*1	✓	—
	TIM6_7	I		✓	✓*1	✓	—
	TIM7_0	I		✓	✓*1	✓	—
	TIM7_1	I		✓	✓*1	✓	—
	TIM7_2	I		✓	✓*1	✓	—
TIM7_3	I	✓	✓*1	✓	—		
TIM7_4	I	✓	✓*1	✓	—		
TIM7_5	I	✓	✓*1	✓	—		
TIM7_6	I	✓	✓*1	✓	—		
TIM7_7	I	✓	✓*1	✓	—		

Table 2.63 Pin Function (7/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
GTM	TOM0_0	O	Output signal from GTM (TOM)	✓	✓	✓	✓
	TOM0_1	O		✓	✓	✓	✓
	TOM0_2	O		✓	✓	✓	✓
	TOM0_3	O		✓	✓	✓	✓
	TOM0_4	O		✓	✓	✓	✓
	TOM0_5	O		✓	✓	✓	✓
	TOM0_6	O		✓	✓	✓	✓
	TOM0_7	O		✓	✓	✓	✓
	TOM0_8	O		✓	✓	✓	✓
	TOM0_9	O		✓	✓	✓	✓
	TOM0_10	O		✓	✓	✓	✓
	TOM0_11	O		✓	✓	✓	✓
	TOM0_12	O		✓	✓	✓	✓
	TOM0_13	O		✓	✓	✓	✓
	TOM0_14	O		✓	✓	✓	✓
	TOM0_15	O		✓	✓	✓	✓
	TOM1_0	O		✓	✓	✓	✓
	TOM1_1	O		✓	✓	✓	✓
	TOM1_2	O		✓	✓	✓	✓
	TOM1_3	O		✓	✓	✓	✓
	TOM1_4	O		✓	✓	✓	✓
	TOM1_5	O		✓	✓	✓	✓
	TOM1_6	O		✓	✓	✓	✓
	TOM1_7	O		✓	✓	✓	✓
	TOM1_8	O		✓	✓	✓	✓
	TOM1_9	O		✓	✓	✓	✓
	TOM1_10	O		✓	✓	✓	✓
	TOM1_11	O		✓	✓	✓	✓
	TOM1_12	O		✓	✓	✓	✓
	TOM1_13	O		✓	✓	✓	✓
	TOM1_14	O		✓	✓	✓	✓
	TOM1_15	O		✓	✓	✓	✓
	TOM2_0	O		✓	✓	✓	✓
	TOM2_1	O		✓	✓	✓	✓
	TOM2_2	O		✓	✓	✓	✓
	TOM2_3	O		✓	✓	✓	✓
	TOM2_4	O		✓	✓	✓	✓
	TOM2_5	O		✓	✓	✓	✓
	TOM2_6	O		✓	✓	✓	✓
	TOM2_7	O		✓	✓	✓	✓
TOM2_8	O	✓	✓	✓	✓		
TOM2_9	O	✓	✓	✓	✓		
TOM2_10	O	✓	✓	✓	✓		

Table 2.63 Pin Function (8/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
GTM	TOM2_11	O	Output signal from GTM (TOM)	✓	✓	✓	✓
	TOM2_12	O		✓	✓	✓	✓
	TOM2_13	O		✓	✓	✓	✓
	TOM2_14	O		✓	✓	✓	✓
	TOM2_15	O		✓	✓	✓	✓
	TOM3_0	O		✓	—	✓	—
	TOM3_1	O		✓	—	✓	—
	TOM3_2	O		✓	—	✓	—
	TOM3_3	O		✓	—	✓	—
	TOM3_4	O		✓	—	✓	—
	TOM3_5	O		✓	—	✓	—
	TOM3_6	O		✓	—	✓	—
	TOM3_7	O		✓	—	✓	—
	TOM3_8	O		✓	—	✓	—
	TOM3_9	O		✓	—	✓	—
	TOM3_10	O		✓	—	✓	—
	TOM3_11	O		✓	—	✓	—
	TOM3_12	O		✓	✓	✓	✓
	TOM3_13	O		✓	—	✓	—
	TOM3_14	O		✓	—	✓	—
	TOM3_15	O		✓	—	✓	—
	TOM4_0	O		✓	—	—	—
	TOM4_1	O		✓	—	—	—
	TOM4_2	O		✓	—	—	—
	TOM4_3	O		✓	—	—	—
	TOM4_4	O		✓	—	—	—
	TOM4_5	O		✓	—	—	—
	TOM4_6	O		✓	—	—	—
	TOM4_7	O		✓	—	—	—
	TOM4_8	O		✓	—	—	—
	TOM4_9	O		✓	—	—	—
	TOM4_10	O		✓	—	—	—
TOM4_11	O	✓	—	—	—		
TOM4_12	O	✓	✓	—	—		
TOM4_13	O	✓	✓	—	—		
TOM4_14	O	✓	✓	—	—		
TOM4_15	O	✓	✓	—	—		

Table 2.63 Pin Function (9/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M		
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292	
GTM	TOM0_0N	O	Output signal from GTM (TOM_N for DTM)	✓	✓	✓	✓	
	TOM0_1N	O		✓	✓	✓	✓	
	TOM0_2N	O		✓	✓	✓	✓	
	TOM0_3N	O		✓	✓	✓	✓	
	TOM0_4N	O		✓	✓	✓	✓	
	TOM0_5N	O		✓	✓	✓	✓	
	TOM0_6N	O		✓	✓	✓	✓	
	TOM0_7N	O		✓	✓	✓	✓	
	TOM1_0N	O		✓	✓	✓	✓	
	TOM1_1N	O		✓	✓	✓	✓	
	TOM1_2N	O		✓	✓	✓	✓	
	TOM1_3N	O		✓	✓	✓	✓	
	TOM1_4N	O		✓	✓	✓	✓	
	TOM1_5N	O		✓	✓	✓	✓	
	TOM1_6N	O		✓	✓	✓	✓	
	TOM1_7N	O		✓	✓	✓	✓	
	TOM2_0N	O		✓	✓	✓	✓	
	TOM2_1N	O		✓	✓	✓	✓	
	TOM2_2N	O		✓	✓	✓	✓	
	TOM2_3N	O		✓	✓	✓	✓	
	TOM2_4N	O		✓	✓	✓	✓	
	TOM2_5N	O		✓	✓	✓	✓	
	TOM2_6N	O		✓	✓	✓	✓	
	TOM2_7N	O		✓	✓	✓	✓	
	TOM3_0N	O		✓	✓	—	—	
	TOM3_1N	O		✓	✓	—	—	
	TOM3_2N	O		✓	✓	—	—	
	TOM3_3N	O		✓	✓	—	—	
	TOM3_4N	O		✓	✓	—	—	
	TOM3_5N	O		✓	✓	—	—	
	TOM3_6N	O		✓	✓	—	—	
	TOM3_7N	O		✓	✓	—	—	
	ATOM0_0	O		Output signal from GTM (ATOM)	✓	✓	✓	✓
	ATOM0_1	O			✓	✓	✓	✓
	ATOM0_2	O			✓	✓	✓	✓
ATOM0_3	O	✓	✓		✓	✓		
ATOM0_4	O	✓	✓		✓	✓		
ATOM0_5	O	✓	✓		✓	✓		
ATOM0_6	O	✓	✓		✓	✓		
ATOM0_7	O	✓	✓		✓	✓		
ATOM1_0	O	✓	✓		✓	✓		
ATOM1_1	O	✓	✓		✓	✓		
ATOM1_2	O	✓	✓		✓	✓		

Table 2.63 Pin Function (10/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
GTM	ATOM1_3	O	Output signal from GTM (ATOM)	✓	✓	✓	✓
	ATOM1_4	O		✓	✓	✓	✓
	ATOM1_5	O		✓	✓	✓	✓
	ATOM1_6	O		✓	✓	✓	✓
	ATOM1_7	O		✓	✓	✓	✓
	ATOM2_0	O		✓	✓	✓	✓
	ATOM2_1	O		✓	✓	✓	✓
	ATOM2_2	O		✓	✓	✓	✓
	ATOM2_3	O		✓	✓	✓	✓
	ATOM2_4	O		✓	✓	✓	✓
	ATOM2_5	O		✓	✓	✓	✓
	ATOM2_6	O		✓	✓	✓	✓
	ATOM2_7	O		✓	✓	✓	✓
	ATOM3_0	O		✓	✓	✓	✓
	ATOM3_1	O		✓	✓	✓	✓
	ATOM3_2	O		✓	✓	✓	✓
	ATOM3_3	O		✓	✓	✓	✓
	ATOM3_4	O		✓	✓	✓	✓
	ATOM3_5	O		✓	✓	✓	✓
	ATOM3_6	O		✓	✓	✓	✓
	ATOM3_7	O		✓	✓	✓	✓
	ATOM4_0	O		✓	✓	✓	✓
	ATOM4_1	O		✓	✓	✓	✓
	ATOM4_2	O		✓	✓	✓	✓
	ATOM4_3	O		✓	✓	✓	✓
	ATOM4_4	O		✓	✓	✓	✓
	ATOM4_5	O		✓	✓	✓	✓
	ATOM4_6	O		✓	✓	✓	✓
	ATOM4_7	O		✓	✓	✓	✓
	ATOM5_0	O		✓	✓	✓	✓
	ATOM5_1	O		✓	✓	✓	✓
	ATOM5_2	O		✓	✓	✓	✓
ATOM5_3	O	✓	✓	✓	✓		
ATOM5_4	O	✓	✓	✓	✓		
ATOM5_5	O	✓	✓	✓	✓		
ATOM5_6	O	✓	✓	✓	✓		
ATOM5_7	O	✓	✓	✓	✓		
ATOM6_0	O	✓	✓	✓	✓		
ATOM6_1	O	✓	✓	✓	✓		
ATOM6_2	O	✓	✓	✓	✓		
ATOM6_3	O	✓	✓	✓	✓		
ATOM6_4	O	✓	✓	✓	✓		
ATOM6_5	O	✓	✓	✓	✓		

Table 2.63 Pin Function (11/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
GTM	ATOM6_6	O	Output signal from GTM (ATOM)	✓	✓	✓	✓
	ATOM6_7	O		✓	✓	✓	✓
	ATOM7_0	O		✓	✓	✓	✓
	ATOM7_1	O		✓	✓	✓	✓
	ATOM7_2	O		✓	✓	✓	✓
	ATOM7_3	O		✓	✓	✓	✓
	ATOM7_4	O		✓	✓	✓	✓
	ATOM7_5	O		✓	✓	✓	✓
	ATOM7_6	O		✓	✓	✓	✓
	ATOM7_7	O		✓	✓	✓	✓
	ATOM8_0	O		✓	—	—	—
	ATOM8_1	O		✓	—	—	—
	ATOM8_2	O		✓	—	—	—
	ATOM8_3	O		✓	—	—	—
	ATOM8_4	O		✓	✓	—	—
	ATOM8_5	O		✓	—	—	—
	ATOM8_6	O		✓	—	—	—
	ATOM8_7	O		✓	—	—	—
	ATOM0_0N	O	Output signal from GTM (ATOM_N for DTM)	✓	✓	✓	✓
	ATOM0_1N	O		✓	✓	✓	✓
	ATOM0_2N	O		✓	✓	✓	✓
	ATOM0_3N	O		✓	✓	✓	✓
	ATOM0_4N	O		✓	✓	✓	✓
	ATOM0_5N	O		✓	✓	✓	✓
	ATOM0_6N	O		✓	✓	✓	✓
	ATOM0_7N	O		✓	✓	✓	✓
	ATOM1_0N	O		✓	✓	✓	✓
	ATOM1_1N	O		✓	✓	✓	✓
	ATOM1_2N	O		✓	✓	✓	✓
	ATOM1_3N	O		✓	✓	✓	✓
	ATOM1_4N	O		✓	✓	✓	✓
	ATOM1_5N	O		✓	✓	✓	✓
	ATOM1_6N	O		✓	✓	✓	✓
	ATOM1_7N	O		✓	✓	✓	✓
	ATOM2_0N	O		✓	✓	✓	✓
	ATOM2_1N	O		✓	✓	✓	✓
	ATOM2_2N	O		✓	✓	✓	✓
	ATOM2_3N	O		✓	✓	✓	✓
	ATOM2_4N	O		✓	✓	✓	✓
	ATOM2_5N	O		✓	✓	✓	✓
	ATOM2_6N	O		✓	✓	✓	✓
	ATOM2_7N	O		✓	✓	✓	✓
ATOM3_0N	O	✓		✓	✓	✓	

Table 2.63 Pin Function (12/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
GTM	ATOM3_1N	O	Output signal from GTM (ATOM_N for DTM)	✓	✓	✓	✓
	ATOM3_2N	O		✓	✓	✓	✓
	ATOM3_3N	O		✓	✓	✓	✓
	ATOM3_4N	O		✓	✓	✓	✓
	ATOM3_5N	O		✓	✓	✓	✓
	ATOM3_6N	O		✓	✓	✓	✓
	ATOM3_7N	O		✓	✓	✓	✓
	ATOM4_0N	O		✓	✓	✓	✓
	ATOM4_1N	O		✓	✓	✓	✓
	ATOM4_2N	O		✓	✓	✓	✓
	ATOM4_3N	O		✓	✓	✓	✓
	ATOM4_4N	O		✓	✓	✓	✓
	ATOM4_5N	O		✓	✓	✓	✓
	ATOM4_6N	O		✓	✓	✓	✓
	ATOM4_7N	O		✓	✓	✓	✓
	ATOM5_0N	O		✓	✓	—	—
	ATOM5_1N	O		✓	✓	—	—
	ATOM5_2N	O		✓	✓	—	—
	ATOM5_3N	O		✓	✓	—	—
	ATOM6_0N	O		✓	—	—	—
	ATOM6_1N	O		✓	—	—	—
	ATOM6_2N	O		✓	—	—	—
	ATOM6_3N	O		✓	—	—	—
GTM	GTMECLK0	O	Output clock signal from GTM	✓	✓	✓	✓
	GTMECLK1	O		✓	✓	✓	✓
	GTMECLK2	O		✓	✓	✓	✓
ENCA	ENCA0E0	I	Input signal to ENCA ch0 E0	✓	✓	✓	✓
	ENCA0E1	I	Input signal to ENCA ch0 E1	✓	✓	✓	✓
	ENCA0EC	I	Input signal to ENCA ch0 EC	✓	✓	✓	✓
	ENCA1E0	I	Input signal to ENCA ch1 E0	✓	✓	✓	✓
	ENCA1E1	I	Input signal to ENCA ch1 E1	✓	✓	✓	✓
	ENCA1EC	I	Input signal to ENCA ch1 EC	✓	✓	✓	✓

Table 2.63 Pin Function (13/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
RLIN3	LRX0	I	LIN0 receive data input	✓	✓	✓	✓
	LTX0	O	LIN0 transmit data output	✓	✓	✓	✓
	LRX1	I	LIN1 receive data input	✓	✓	✓	✓
	LTX1	O	LIN1 transmit data output	✓	✓	✓	✓
	LRX2	I	LIN2 receive data input	✓	✓	✓	✓
	LTX2	O	LIN2 transmit data output	✓	✓	✓	✓
	LRX3	I	LIN3 receive data input	✓	✓	✓	✓
	LTX3	O	LIN3 transmit data output	✓	✓	✓	✓
	LRX4	I	LIN4 receive data input	✓	✓	✓	✓
	LTX4	O	LIN4 transmit data output	✓	✓	✓	✓
	LRX5	I	LIN5 receive data input	✓	✓	✓	✓
	LTX5	O	LIN5 transmit data output	✓	✓	✓	✓
SCI0	SCI0TxD	O	SCI0 transmit data output	✓	✓	✓	✓
	SCI0RxD	I	SCI0 receive data input	✓	✓	✓	✓
	SCI0SCK	IO	SCI0 serial clock input/output	✓	✓	✓	✓
SCI1	SCI1TxD	O	SCI1 transmit data output	✓	✓	✓	✓
	SCI1RxD	I	SCI1 receive data input	✓	✓	✓	✓
	SCI1SCK	IO	SCI1 serial clock input/output	✓	✓	✓	✓
SCI2	SCI2TxD	O	SCI2 transmit data output	✓	✓	✓	✓
	SCI2RxD	I	SCI2 receive data input	✓	✓	✓	✓
	SCI2SCK	IO	SCI2 serial clock input/output	✓	✓	✓	✓
SCI3	SCI3TxD	O	SCI3 transmit data output	✓	✓	✓	✓
	SCI3RxD	I	SCI3 receive data input	✓	✓	✓	✓
	SCI3SCK	IO	SCI3 serial clock input/output	✓	✓	✓	✓
CSIH0	CSIH0TSI	I	CSIH0 receive data input	✓	✓	✓	✓
	CSIH0TSO	O	CSIH0 transmit data output	✓	✓	✓	✓
	CSIH0TSCK	IO	CSIH0 serial clock input/output	✓	✓	✓	✓
	CSIH0TSSI	I	CSIH0 slave select signal input	✓	✓	✓	✓
	CSIH0TRY	IO	CSIH0 handshake signal input/output	✓	✓	✓	✓
	CSIH0TCSS0	O	CSIH0 slave select signal output	✓	✓	✓	✓
	CSIH0TCSS1	O		✓	✓	✓	✓
CSIH0	CSIH0TCSS2	O	CSIH0 slave select signal output	✓	✓	✓	✓
	CSIH0TCSS3	O		✓	✓	✓	✓
	CSIH0TCSS4	O		✓	✓	✓	✓
	CSIH0TCSS5	O		✓	✓	✓	✓

Table 2.63 Pin Function (14/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
CSIH1	CSIH1TSI	I	CSIH1 receive data input	✓	✓	✓	✓
	CSIH1TSO	O	CSIH1 transmit data output	✓	✓	✓	✓
	CSIH1TSCK	IO	CSIH1 serial clock input/output	✓	✓	✓	✓
	$\overline{\text{CSIH1TSSI}}$	I	CSIH1 slave select signal input	✓	✓	✓	✓
	CSIH1TRY	IO	CSIH1 handshake signal input/output	✓	✓	✓	✓
	CSIH1TCSS0	O	CSIH1 slave select signal output	✓	✓	✓	✓
	CSIH1TCSS1	O		✓	✓	✓	✓
	CSIH1TCSS2	O		✓	✓	✓	✓
	CSIH1TCSS3	O		✓	✓	✓	✓
CSIH2	CSIH2TSI	I	CSIH2 receive data input	✓	✓	✓	✓
	CSIH2TSO	O	CSIH2 transmit data output	✓	✓	✓	✓
	CSIH2TSCK	IO	CSIH2 serial clock input/output	✓	✓	✓	✓
	$\overline{\text{CSIH2TSSI}}$	I	CSIH2 slave select signal input	✓	✓	✓	✓
	CSIH2TRY	IO	CSIH2 handshake signal input/output	✓	✓	✓	✓
	CSIH2TCSS0	O	CSIH2 slave select signal output	✓	✓	✓	✓
	CSIH2TCSS1	O		✓	✓	✓	✓
	CSIH2TCSS2	O		✓	✓	✓	✓
	CSIH2TCSS3	O		✓	✓	✓	✓
CSIH3	CSIH3TSI	I	CSIH3 receive data input	✓	✓	✓	✓
	CSIH3TSO	O	CSIH3 transmit data output	✓	✓	✓	✓
	CSIH3TSCK	IO	CSIH3 serial clock input/output	✓	✓	✓	✓
	$\overline{\text{CSIH3TSSI}}$	I	CSIH3 slave select signal input	✓	✓	✓	✓
	CSIH3TRY	IO	CSIH3 handshake signal input/output	✓	✓	✓	✓
	CSIH3TCSS0	O	CSIH3 slave select signal output	✓	✓	✓	✓
	CSIH3TCSS1	O		✓	✓	✓	✓
	CSIH3TCSS2	O		✓	✓	✓	✓
	CSIH3TCSS3	O		✓	✓	✓	✓
CSIH4	CSIH4TSI	I	CSIH4 receive data input	✓	✓	✓	✓
	CSIH4TSO	O	CSIH4 transmit data output	✓	✓	✓	✓
	CSIH4TSCK	IO	CSIH4 serial clock input/output	✓	✓	✓	✓
	$\overline{\text{CSIH4TSSI}}$	I	CSIH4 slave select signal input	✓	✓	✓	✓
	CSIH4TRY	IO	CSIH4 handshake signal input/output	✓	✓	✓	✓
	CSIH4TCSS0	O	CSIH4 slave select signal output	✓	✓	✓	✓
	CSIH4TCSS1	O		✓	✓	✓	✓
	CSIH4TCSS2	O		✓	✓	✓	✓
	CSIH4TCSS3	O		✓	✓	✓	✓

Table 2.63 Pin Function (15/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
CSIH5	CSIH5TSI	I	CSIH5 receive data input	✓	✓	✓	✓
	CSIH5TSO	O	CSIH5 transmit data output	✓	✓	✓	✓
	CSIH5TSCK	IO	CSIH5 serial clock input/output	✓	✓	✓	✓
	CSIH5TSSI	I	CSIH5 slave select signal input	✓	✓	✓	✓
	CSIH5TRY	IO	CSIH5 handshake signal input/output	✓	✓	✓	✓
	CSIH5TCSS0	O	CSIH5 slave select signal output	✓	✓	✓	✓
	CSIH5TCSS1	O		✓	✓	✓	✓
	CSIH5TCSS2	O		✓	✓	✓	✓
	CSIH5TCSS3	O		✓	✓	✓	✓
HSSPI0	HSSPI0_CLKN	IO	HSSPI clock input/output	✓	✓	✓	✓
	HSSPI0_CLKP	IO	HSSPI clock input/output	✓	✓	✓	✓
	HSSPI0_RXDN	I	HSSPI receive data input	✓	✓	✓	✓
	HSSPI0_RXDP	I	HSSPI receive data input	✓	✓	✓	✓
	HSSPI0_TXDN	O	HSSPI transmit data output	✓	✓	✓	✓
	HSSPI0_TXDP	O	HSSPI transmit data output	✓	✓	✓	✓
	HSSPI0_SSLI	I	HSSPI salve select signal input	✓	✓	✓	✓
	HSSPI0_SSL	O	HSSPI salve select signal output	✓	✓	✓	✓
	HSSPI0_CLK	IO	HSSPI single mode clock input/output	✓	✓	✓	✓
	HSSPI0_RXD	I	HSSPI single mode receive data input	✓	✓	✓	✓
	HSSPI0_TXD	O	HSSPI single mode transmit data output	✓	✓	✓	✓
HSIF0	HSIF0_REFCLK	IO	RHSIF0 Reference clock input/output	✓	✓	✓	✓
	HSIF0_TXDN	O	RHSIF0 transmission data differential output	✓	✓	✓	✓
	HSIF0_TXDP	O		✓	✓	✓	✓
	HSIF0_RXDN	I	RHSIF0 receive data differential input	✓	✓	✓	✓
	HSIF0_RXDP	I		✓	✓	✓	✓
PSI5S	PSISCLK	O	UART clock	✓	✓	✓	✓
	PSISRX	I	UART Rx data	✓	✓	✓	✓
	PSISTX	O	UART Tx data	✓	✓	✓	✓
RSCAN0	CTX0	O	CAN0 transmit data output	✓	✓	✓	✓
	CRX0	I	CAN0 receive data input	✓	✓	✓	✓
RSCAN1	CTX1	O	CAN1 transmit data output	✓	✓	✓	✓
	CRX1	I	CAN1 receive data input	✓	✓	✓	✓
RSCAN2	CTX2	O	CAN2 transmit data output	✓	✓	✓	✓
	CRX2	I	CAN2 receive data input	✓	✓	✓	✓
RSCAN3	CTX3	O	CAN3 transmit data output	✓	✓	✓	✓
	CRX3	I	CAN3 receive data input	✓	✓	✓	✓
RSCAN4	CTX4	O	CAN4 transmit data output	✓	✓	✓	✓
	CRX4	I	CAN4 receive data input	✓	✓	✓	✓

Table 2.63 Pin Function (16/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
FlexRay0	FR0RXDA	I	FlexRay0 channel A receive data input	✓	✓	✓	✓
	FR0TXDA	O	FlexRay0 channel A transmit data output	✓	✓	✓	✓
	FR0ENA	O	FlexRay0 channel A transmit data enable	✓	✓	✓	✓
	FR0RXDB	I	FlexRay0 channel B receive data input	✓	✓	✓	✓
	FR0TXDB	O	FlexRay0 channel B transmit data output	✓	✓	✓	✓
	FR0ENB	O	FlexRay0 channel B transmit data enable	✓	✓	✓	✓
RHSB0	RHSB0FCLP	O	RHSB0 differential clock output	✓	✓	✓	✓
	RHSB0FCLN	O		✓	✓	✓	✓
	RHSB0SOP	O	RHSB0 differential data output	✓	✓	✓	✓
	RHSB0SON	O		✓	✓	✓	✓
	RHSB0CSD0	O	RHSB0 chip select output	✓	✓	✓	✓
	RHSB0CSD1	O		✓	✓	✓	✓
	RHSB0SI0	I	RHSB0 serial input	✓	✓	✓	✓
	RHSB0SI1	I		✓	✓	✓	✓
	RHSB0EMRG	I	RHSB0 emergency signal input	✓	✓	✓	✓
RHSB1	RHSB1FCLP	O	RHSB1 differential clock output	✓	✓	✓	✓
	RHSB1FCLN	O		✓	✓	✓	✓
	RHSB1SOP	O	RHSB1 differential data output	✓	✓	✓	✓
	RHSB1SON	O		✓	✓	✓	✓
RHSB1	RHSB1CSD0	O	RHSB1 chip select output	✓	✓	✓	✓
	RHSB1CSD1	O		✓	✓	✓	✓
	RHSB1SI0	I	RHSB1 serial input	✓	✓	✓	✓
	RHSB1SI1	I		✓	✓	✓	✓
	RHSB1EMRG	I	RHSB1 emergency signal input	✓	✓	✓	✓
FastEther (MII)	MII_TX_CLK	I	Transmit clock signal	✓	✓	✓	✓
	MII_RX_CLK	I	Receive clock signal	✓	✓	✓	✓
	MII_TX_EN	O	Transmit data enable output	✓	✓	✓	✓
	MII_TXD3	O	Transmit data output	✓	✓	✓	✓
	MII_TXD2	O		✓	✓	✓	✓
	MII_TXD1	O		✓	✓	✓	✓
	MII_TXD0	O		✓	✓	✓	✓
	MII_TX_ER	O	Transmit error output	✓	✓	✓	✓
	MII_RX_DV	I	Receive data valid input	✓	✓	✓	✓
	MII_RXD3	I	Receive data input	✓	✓	✓	✓
	MII_RXD2	I		✓	✓	✓	✓
	MII_RXD1	I		✓	✓	✓	✓
	MII_RXD0	I		✓	✓	✓	✓
	MII_RX_ER	I	Receive data error input	✓	✓	✓	✓
	MII_CRS	I	Carrier detect input	✓	✓	✓	✓
MII_COL	I	Collision detect input	✓	✓	✓	✓	

Table 2.63 Pin Function (17/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
FastEther (RMII)	REF50CK	I	Timing reference signal	✓	✓	✓	✓
	RMII_TXD1	O	Transmit data output	✓	✓	✓	✓
	RMII_TXD0	O		✓	✓	✓	✓
	RMII_TX_EN	O	Transmit data enable	✓	✓	✓	✓
	RMII_RXD1	I	Receive data input	✓	✓	✓	✓
	RMII_RXD0	I		✓	✓	✓	✓
	RMII_RX_ER	I	Receive data error input	✓	✓	✓	✓
	RMII_CRS_DV	I	Carrier detect input	✓	✓	✓	✓
FastEther (COMMON)	WOL	O	Wakeup on LAN signal	✓	✓	✓	✓
	MDC	O	PHY chip setting I/F clock output	✓	✓	✓	✓
	MDIO	IO	PHY chip setting I/F data input/output	✓	✓	✓	✓
	LINKSTA	I	Link status input	✓	✓	✓	✓
	EXOUT	O	External output port	✓	✓	✓	✓
RSENT0	RSENT0RX	I	RSENT0 receive data input	✓	✓	✓	✓
	RSENT0SPCO	O	RSENT0 SPC extension output	✓	✓	✓	✓
RSENT1	RSENT1RX	I	RSENT1 receive data input	✓	✓	✓	✓
	RSENT1SPCO	O	RSENT1 SPC extension output	✓	✓	✓	✓
RSENT2	RSENT2RX	I	RSENT2 receive data input	✓	✓	✓	✓
	RSENT2SPCO	O	RSENT2 SPC extension output	✓	✓	✓	✓
RSENT3	RSENT3RX	I	RSENT3 receive data input	✓	✓	✓	✓
	RSENT3SPCO	O	RSENT3 SPC extension output	✓	✓	✓	✓
RSENT4	RSENT4RX	I	RSENT4 receive data input	✓	✓	✓	✓
	RSENT4SPCO	O	RSENT4 SPC extension output	✓	✓	✓	✓
RSENT5	RSENT5RX	I	RSENT5 receive data input	✓	✓	✓	✓
	RSENT5SPCO	O	RSENT5 SPC extension output	✓	✓	✓	✓
RSENT6	RSENT6RX	I	RSENT6 receive data input	✓	✓	✓	✓
	RSENT6SPCO	O	RSENT6 SPC extension output	✓	✓	✓	✓
RSENT7	RSENT7RX	I	RSENT7 receive data input	✓	✓	✓	✓
	RSENT7SPCO	O	RSENT7 SPC extension output	✓	✓	✓	✓
RSENT8	RSENT8RX	I	RSENT8 receive data input	✓	✓	✓	✓
	RSENT8SPCO	O	RSENT8 SPC extension output	✓	✓	✓	✓
RSENT9	RSENT9RX	I	RSENT9 receive data input	✓	✓	✓	✓
	RSENT9SPCO	O	RSENT9 SPC extension output	✓	✓	✓	✓
RSENT10	RSENT10RX	I	RSENT10 receive data input	✓	✓	✓	✓
	RSENT10SPCO	O	RSENT10 SPC extension output	✓	✓	✓	✓
RSENT11	RSENT11RX	I	RSENT11 receive data input	✓	✓	✓	✓
	RSENT11SPCO	O	RSENT11 SPC extension output	✓	✓	✓	✓
RSENT12	RSENT12RX	I	RSENT12 receive data input	✓	✓	✓	✓
	RSENT12SPCO	O	RSENT12 SPC extension output	✓	✓	✓	✓
RSENT13	RSENT13RX	I	RSENT13 receive data input	✓	✓	✓	✓
	RSENT13SPCO	O	RSENT13 SPC extension output	✓	✓	✓	✓
RSENT14	RSENT14RX	I	RSENT14 receive data input	✓	✓	✓	✓
	RSENT14SPCO	O	RSENT14 SPC extension output	✓	✓	✓	✓

Table 2.63 Pin Function (18/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
RSENT15	RSENT15RX	I	RSENT15 receive data input	✓	✓	✓	✓
	RSENT15SPCO	O	RSENT15 SPC extension output	✓	✓	✓	✓
RSENT16	RSENT16RX	I	RSENT16 receive data input	✓	✓	✓	✓
	RSENT16SPCO	O	RSENT16 SPC extension output	✓	✓	✓	✓
ADC0	AN000	I	ADC0 analog input	✓	✓	✓	✓
	AN001	I		✓	✓	✓	✓
	AN002	I		✓	✓	✓	✓
	AN003	I		✓	✓	✓	✓
	AN010	I		✓	✓	✓	✓
	AN011	I		✓	✓	✓	✓
	AN012	I		✓	✓	✓	✓
	AN013	I		✓	✓	✓	✓
	AN020	I		✓	✓	✓	✓
	AN021	I		✓	✓	✓	✓
	AN022	I		✓	✓	✓	✓
	AN023	I		✓	✓	✓	✓
	AN030	I		✓	✓	✓	✓
	AN031	I		✓	✓	✓	✓
	AN032	I		✓	✓	✓	✓
	AN033	I		✓	✓	✓	✓
	AN040	I		✓	✓	✓	✓
	AN041	I		✓	✓	✓	✓
	AN042	I		✓	✓	✓	✓
	AN043	I		✓	✓	✓	✓
	AN050	I		✓	✓	✓	✓
	AN051	I		✓	✓	✓	✓
	AN052	I		✓	✓	✓	✓
AN053	I	✓	✓	✓	✓		
AN060	I	✓	✓	✓	✓		
AN061	I	✓	✓	✓	✓		
AN062	I	✓	✓	✓	✓		
AN063	I	✓	✓	✓	✓		
ADC0	ADTRG0	I	ADC0 conversion startup trigger input	✓	✓	✓	✓
	ADEND0	O	ADC0 conversion timing monitor output	✓	✓	✓	✓
	ADMPX00	O	ADC0 ADMPX0 output	✓	✓	✓	✓
	ADMPX01	O	ADC0 ADMPX1 output	✓	✓	✓	✓
	ADMPX02	O	ADC0 ADMPX2 output	✓	✓	✓	✓
	ADMPX03	O	ADC0 ADMPX3 output	✓	✓	✓	✓
	ADMPX04	O	ADC0 ADMPX4 output	✓	✓	✓	✓

Table 2.63 Pin Function (19/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M		
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292	
ADC1	AN100	I	ADC1 analog input	✓	—	✓	—	
	AN101	I		✓	—	✓	—	
	AN102	I		✓	—	✓	—	
	AN103	I		✓	—	✓	—	
	AN110	I		✓	—	✓	—	
	AN111	I		✓	—	✓	—	
	AN112	I		✓	—	✓	—	
	AN113	I		✓	—	✓	—	
	AN120	I		✓	—	✓	—	
	AN121	I		✓	—	✓	—	
	AN122	I		✓	—	✓	—	
	AN123	I		✓	—	✓	—	
	AN130	I		✓	—	✓	—	
	AN131	I		✓	—	✓	—	
	AN132	I		✓	—	✓	—	
	AN133	I		✓	—	✓	—	
	AN140	I		✓	—	✓	—	
	AN141	I		✓	—	✓	—	
	AN142	I		✓	—	✓	—	
	AN143	I		✓	—	✓	—	
	ADTRG1	I		ADC1 conversion startup trigger input	✓	—	✓	—
	ADEND1	O		ADC1 conversion timing monitor output	✓	—	✓	—
	ADMPX10	O		ADC1 ADMPX0 output	✓	—	✓	—
ADMPX11	O	ADC1 ADMPX1 output	✓	—	✓	—		
ADMPX12	O	ADC1 ADMPX2 output	✓	—	✓	—		
ADMPX13	O	ADC1 ADMPX3 output	✓	—	✓	—		
ADMPX14	O	ADC1 ADMPX4 output	✓	—	✓	—		
ADC2	AN200	I	ADC2 analog input	✓	✓	✓	✓	
	AN201	I		✓	✓	✓	✓	
	AN202	I		✓	✓	✓	✓	
	AN203	I		✓	✓	✓	✓	
	AN210	I		✓	✓	✓	✓	
	AN211	I		✓	✓	✓	✓	
	AN212	I		✓	✓	✓	✓	
	AN213	I		✓	✓	✓	✓	
	AN220	I		✓	✓	✓	✓	
	AN221	I		✓	✓	✓	✓	
	AN222	I		✓	✓	✓	✓	
	AN223	I		✓	✓	✓	✓	
	AN230	I		✓	✓	✓	✓	
	AN231	I		✓	✓	✓	✓	
	AN232	I		✓	✓	✓	✓	
	AN233	I		✓	✓	✓	✓	

Table 2.63 Pin Function (20/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
ADC2	AN240	I	ADC2 analog input	✓	✓	✓	✓
	AN241	I		✓	✓	✓	✓
	AN242	I		✓	✓	✓	✓
	AN243	I		✓	✓	✓	✓
	AN250	I		✓	✓	✓	✓
	AN251	I		✓	✓	✓	✓
	AN252	I		✓	✓	✓	✓
	AN253	I		✓	✓	✓	✓
	AN260	I		✓	—	✓	—
	AN261	I		✓	—	✓	—
	AN262	I		✓	—	✓	—
	AN263	I		✓	—	✓	—
	ADTRG2	I		ADC2 conversion startup trigger input	✓	✓	✓
	ADEND2	O	ADC2 conversion timing monitor output	✓	✓	✓	✓
	ADMPX20	O	ADC2 ADMPX0 output	✓	✓	✓	✓
	ADMPX21	O	ADC2 ADMPX1 output	✓	✓	✓	✓
	ADMPX22	O	ADC2 ADMPX2 output	✓	✓	✓	✓
	ADMPX23	O	ADC2 ADMPX3 output	✓	✓	✓	✓
	ADMPX24	O	ADC2 ADMPX4 output	✓	✓	✓	✓
	AN300	I	ADC3 analog input	✓	—	✓	—
	AN301	I		✓	—	✓	—
	AN302	I		✓	—	✓	—
	AN303	I		✓	—	✓	—
	AN310	I		✓	—	✓	—
	AN311	I		✓	—	✓	—
	AN312	I		✓	—	✓	—
	AN313	I		✓	—	✓	—
	AN320	I		✓	—	✓	—
	AN321	I		✓	—	✓	—
	AN322	I		✓	—	✓	—
	AN323	I		✓	—	✓	—
	AN330	I		✓	—	✓	—
	AN331	I		✓	—	✓	—
	AN332	I		✓	—	✓	—
	AN333	I		✓	—	✓	—
	AN340	I		✓	—	✓	—
	AN341	I		✓	—	✓	—
	AN342	I		✓	—	✓	—
	AN343	I		✓	—	✓	—
	ADTRG3	I	ADC3 conversion startup trigger input	✓	—	✓	—
	ADEND3	O	ADC3 conversion timing monitor output	✓	—	✓	—
	ADMPX30	O	ADC3 ADMPX0 output	✓	—	✓	—
ADMPX31	O	ADC3 ADMPX1 output	✓	—	✓	—	

Table 2.63 Pin Function (21/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
ADC2	ADMPX32	O	ADC3 ADMPX2 output	✓	—	✓	—
	ADMPX33	O	ADC3 ADMPX3 output	✓	—	✓	—
	ADMPX34	O	ADC3 ADMPX4 output	✓	—	✓	—
DSADC00	DSAN000P	I	DSADC00 analog input	✓	✓	✓	✓
	DSAN000N	I		✓	✓	✓	✓
	DSAN001P	I		✓	✓	✓	✓
	DSAN001N	I		✓	✓	✓	✓
	DSAN002P	I		✓	✓	✓	✓
	DSAN002N	I		✓	✓	✓	✓
	DSAN003P	I		✓	✓	✓	✓
	DSAN003N	I		✓	✓	✓	✓
	DSADTRG00	I		DSADC00 conversion startup trigger input	✓	✓	✓
	DSADEND00	O	DSADC00 conversion timing monitor output	✓	✓	✓	✓
	DSADC10	DSAN100P	I	DSADC10 analog input	✓	✓	✓
DSAN100N		I	✓		✓	✓	✓
DSAN101P		I	✓		✓	✓	✓
DSAN101N		I	✓		✓	✓	✓
DSADTRG10		I	DSADC10 conversion startup trigger input	✓	✓	✓	✓
DSADEND10		O	DSADC10 conversion timing monitor output	✓	✓	✓	✓
DSADC11	DSAN110P	I	DSADC11 analog input	✓	✓	✓	✓
	DSAN110N	I		✓	✓	✓	✓
	DSAN111P	I		✓	✓	✓	✓
	DSAN111N	I		✓	✓	✓	✓
	DSADTRG11	I	DSADC11 conversion startup trigger input	✓	✓	✓	✓
	DSADEND11	O	DSADC11 conversion timing monitor output	✓	✓	✓	✓
DSADC12	DSAN120P	I	DSADC12 analog input	✓	✓	✓	✓
	DSAN120N	I		✓	✓	✓	✓
	DSAN121P	I		✓	✓	✓	✓
	DSAN121N	I		✓	✓	✓	✓
	DSADTRG12	I	DSADC12 conversion startup trigger input	✓	✓	✓	✓
	DSADEND12	O	DSADC12 conversion timing monitor output	✓	✓	✓	✓
DSADC13	DSAN130P	I	DSADC13 analog input	✓	✓	✓	✓
	DSAN130N	I		✓	✓	✓	✓
	DSAN131P	I		✓	✓	✓	✓
	DSAN131N	I		✓	✓	✓	✓
	DSADTRG13	I	DSADC13 conversion startup trigger input	✓	✓	✓	✓
	DSADEND13	O	DSADC13 conversion timing monitor output	✓	✓	✓	✓

Table 2.63 Pin Function (22/23)

Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
DSADC20	DSAN200P	I	DSADC20 analog input	✓	✓	✓	✓
	DSAN200N	I		✓	✓	✓	✓
	DSADTRG20	I	DSADC20 conversion startup trigger input	✓	✓	✓	✓
	DSADEND20	O	DSADC20 conversion timing monitor output	✓	✓	✓	✓
C-ADC00	CAN000P	I	C-ADC00 analog input	✓	✓	✓	✓
	CAN000N	I		✓	✓	✓	✓
	CAN001P	I		✓	✓	✓	✓
	CAN001N	I		✓	✓	✓	✓
	CAN002P	I		✓	✓	✓	✓
	CAN002N	I		✓	✓	✓	✓
	CAN003P	I		✓	✓	✓	✓
	CAN003N	I		✓	✓	✓	✓
	CADTRG00	I	C-ADC00 conversion startup trigger input	✓	✓	✓	✓
	CADEND00	O	C-ADC00 conversion timing monitor output	✓	✓	✓	✓
ECM	$\overline{\text{ERROROUT_M}}$	O	Error output (master)	✓	✓	✓	✓
	$\overline{\text{ERROROUT_C}}$	O	Error output (checker)	✓	✓	✓	✓
	$\overline{\text{ERRORIN}}$	I	Error input	✓	✓	✓	✓
Interrupt	NMI	I	External interrupt input	✓	✓	✓	✓
	IRQ0	I		✓	✓	✓	✓
	IRQ1	I		✓	✓	✓	✓
	IRQ2	I		✓	✓	✓	✓
	IRQ3	I		✓	✓	✓	✓
	IRQ4	I		✓	✓	✓	✓
	IRQ5	I		✓	✓	✓	✓
	IRQ6	I		✓	✓	✓	✓
	IRQ7	I		✓	✓	✓	✓
Clock	EXTAL	I	External clock	✓	✓	✓	✓
	XTAL	O	Crystal	✓	✓	✓	✓
	CK	O	Peripheral clock	✓	✓	✓	✓
Reset	$\overline{\text{RES_IN}}$	I	External reset input	✓	✓	✓	✓
	$\overline{\text{RES_OUT}}$	O	Reset output	✓	✓	✓	✓
Mode setting	MD0	I	Mode setting	✓	✓	✓	✓
	MD1	I		✓	✓	✓	✓
Nexus/JTAG	$\overline{\text{TRST}}$	I	Reset input	✓	✓	✓	✓
	TDO	O	Serial data output	✓	✓	✓	✓
	TMS	I	Mode select input	✓	✓	✓	✓
	TDI	I	Serial data input	✓	✓	✓	✓
	TCK	I	Clock input	✓	✓	✓	✓
	$\overline{\text{DRDY}}$	O	Ready output	✓	✓	✓	✓
	$\overline{\text{EVTO}}$	O	Event trigger output	✓	✓	✓	✓
	$\overline{\text{EVTI}}$	I	Event trigger input	✓	✓	✓	✓

Table 2.63 Pin Function (23/23)

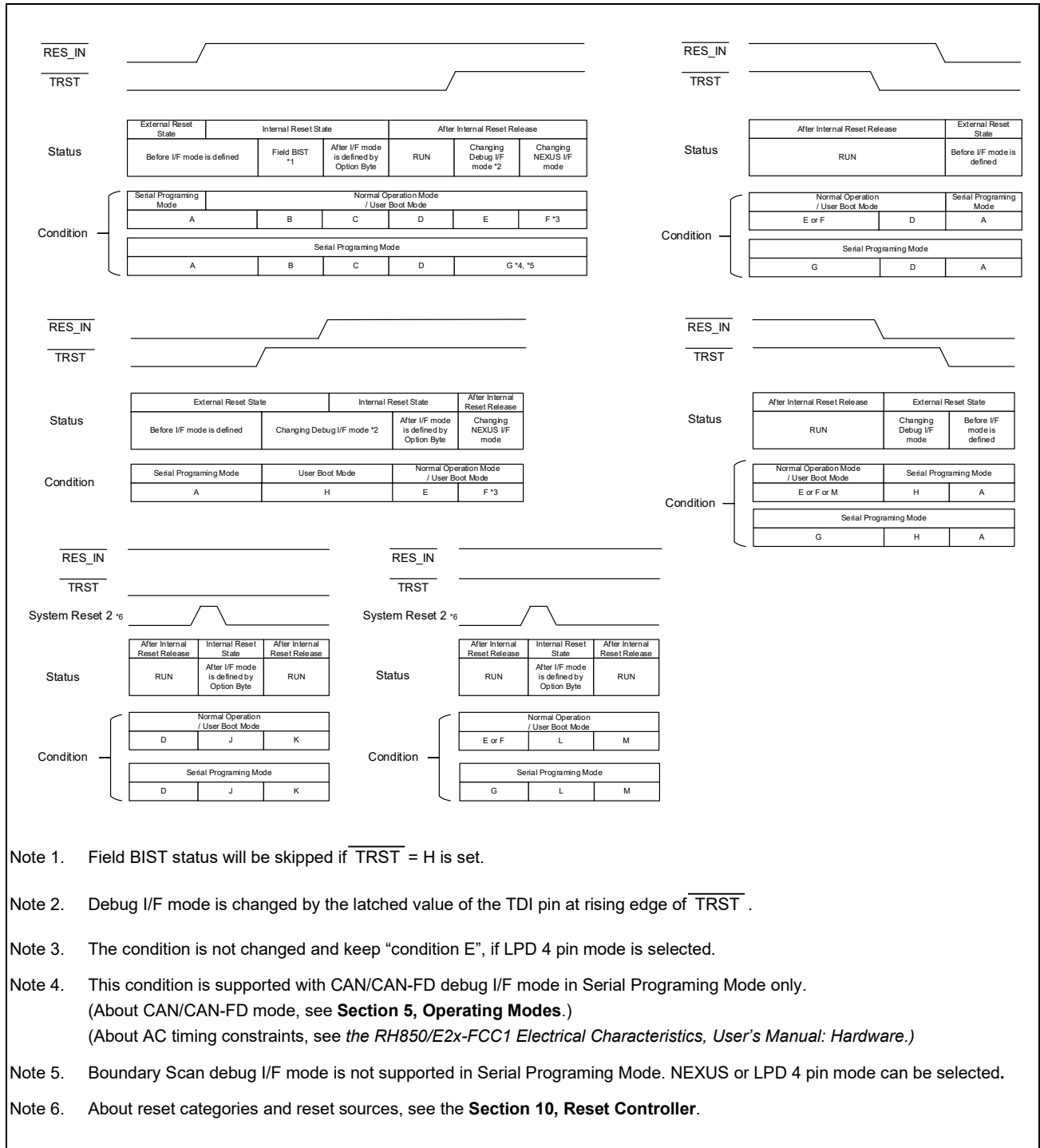
Category	Function Name	IO	Function	E2x-FCC1		E2M	
				BGA 373	BGA292 (E2M)	BGA 373	BGA 292
LPD/DOC	$\overline{\text{LPDRST}}$	I	LPD reset input	✓	✓	✓	✓
	LPDO	O	LPD 4 pin data output	✓	✓	✓	✓
	LPDI	I	LPD 4 pin data input	✓	✓	✓	✓
	LPDCLKO	O	LPD 4 pin clock output	✓	✓	✓	✓
	LPDCLKI	I	LPD 4 pin clock input	✓	✓	✓	✓
AURORA	CICREFP	I	Clock input	✓	✓	—	—
	CICREFN	I		✓	✓	—	—
	TODP0	O	Trace data output	✓	✓	—	—
	TODN0	O		✓	✓	—	—
	$\overline{\text{AURORES}}$	I	AURORA reset	✓	✓	—	—
FLSCI (Writer I/F)	FPDT	O	Transmit data output (2 wire UART, CSI)	✓	✓	✓	✓
	FPDR	I	Receive data input (2 wire UART, CSI)	✓	✓	✓	✓
	FPCK	I	Serial clock input (CSI)	✓	✓	✓	✓

Note 1. These functions are multiplexed via virtual ports, so it can only be used in input mode. (See **Section 2.5.2, List of Alternative Functions** for details). About port group support pins, refer to the **Section 2.1.1, Port Group**.

2.8 Pins Status

See the description of the pin states in **Section 10, Reset Controller** for the definition of each reset state.

The following table describes the condition definition for pin states, such as Condition A is External Reset State.



Note 1. Field BIST status will be skipped if $\overline{\text{TRST}} = \text{H}$ is set.

Note 2. Debug I/F mode is changed by the latched value of the TDI pin at rising edge of $\overline{\text{TRST}}$.

Note 3. The condition is not changed and keep "condition E", if LPD 4 pin mode is selected.

Note 4. This condition is supported with CAN/CAN-FD debug I/F mode in Serial Programming Mode only.

(About CAN/CAN-FD mode, see **Section 5, Operating Modes**.)

(About AC timing constraints, see *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.)

Note 5. Boundary Scan debug I/F mode is not supported in Serial Programming Mode. NEXUS or LPD 4 pin mode can be selected.

Note 6. About reset categories and reset sources, see the **Section 10, Reset Controller**.

Figure 2.19 Pin Status Conditions

Table 2.64 Pin State in Normal Operation Mode, User Boot Mode and Serial Programming Mode

Pin Function		Pin State							
Category	Pin Name	Condition							Power-off Standby Mode
		A	H	B	C	D, E, F, G	J, L	K, M	
Clock	XTAL	O	O	O	O	O	O	O	Z
	EXTAL	I	I	I	I	I	I	I	Z
System control	$\overline{\text{RES_IN}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
	$\overline{\text{RES_OUT}}$	L	L	L	L	L ^{*3}	L	L ^{*3}	L
	MD0	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
	MD1	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
Interrupt	NMI	I (pull-down)	I (pull-down)	pull-down	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	Z
ECM	$\overline{\text{ERROROUT_M}}$	L	L	L	L	L ^{*5}	^{*5}	^{*5}	L
General input/output port	Pn_m	Z	Z	Z	Z	Z	Z	Z	Z
SAR, $\Delta\Sigma$ A/D	ANxxx	Z	Z	Z	Z	Z	Z	Z	Z
Debug system	$\overline{\text{AUORES}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
	CICREFP	Z	Z	Z	Z	Z	Z	Z	Z
	CICREFN	Z	Z	Z	Z	Z	Z	Z	Z
	TODP0	Z	Z	Z	Z	Z	Z	Z	Z
	TODN0	Z	Z	Z	Z	Z	Z	Z	Z

Table 2.65 Pin State in Normal Operation Mode, User Boot Mode (1/2)

Pin Function			Pin State					
Category	IF Mode	Pin Name	Condition					
			A	B	C, J	D, K	E, L, M ^{*6}	F, L, M ^{*6}
Debug System	BSCAN	TDI	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)
		TDO	Z	Z	Z	Z	Z	Z
		TCK	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)
		TMS	Z	Z	Z	Z	I (pull-up)	I (pull-up)
		$\overline{\text{TRST}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{DRDY}} / \text{JP0_5}$	Z	Z	Z	Z	Z	Z
	NEXUS (wo $\overline{\text{DRDY}}$)	TDI	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)
		TDO	Z	Z	Z	Z	Z	Z
		TCK	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)
		TMS	Z	Z	Z	Z	I (pull-up)	I (pull-up)
		$\overline{\text{TRST}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{DRDY}} / \text{JP0_5}$	Z	Z	Z	Z	Z	Z
	NEXUS (w $\overline{\text{DRDY}}$)	TDI	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)
		TDO	Z	Z	Z	Z	Z	Z
		TCK	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)
		TMS	Z	Z	Z	Z	I (pull-up)	I (pull-up)
		$\overline{\text{TRST}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{DRDY}}$	Z	Z	Z	Z	Z	H ^{*4}
	LPD-4 pin	TDI/LPDI	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)
		TDO/LPDO	Z	Z	Z	Z	H	H
		TCK/LPDCLKI	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)
		TMS/JP0_3	Z	Z	Z	Z	Z	Z
		$\overline{\text{TRST}} / \overline{\text{LPDRST}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{DRDY}} / \overline{\text{LPDCLKO}}$	Z	Z	Z	Z	L	L
	GPIO (Write I/F)	TDI/FPDR	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)
		TDO/FPDT	Z	Z	Z	Z	Z	Z
		TCK/FPCK	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)
TMS/JP0_3		Z	Z	Z	Z	Z	Z	
$\overline{\text{TRST}}$		I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	
$\overline{\text{DRDY}} / \text{JP0_5}$		Z	Z	Z	Z	Z	Z	

Table 2.65 Pin State in Normal Operation Mode, User Boot Mode (2/2)

Pin Function			Pin State				Power-off Standby Mode
Category	IF Mode	Pin Name	Condition				
			A	H	E, L, M *6	F, L, M *6	
Debug System	BSCAN	TDI	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TDO	Z	Z	Z	Z	Z
		TCK	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TMS	Z	I (pull-up)	I (pull-up)	I (pull-up)	Z
		$\overline{\text{TRST}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{DRDY}} / \text{JP0_5}$	Z	Z	Z	Z	Z
	NEXUS (wo $\overline{\text{DRDY}}$)	TDI	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TDO	Z	Z	Z	Z	Z
		TCK	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TMS	Z	I (pull-up)	I (pull-up)	I (pull-up)	Z
		$\overline{\text{TRST}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{DRDY}} / \text{JP0_5}$	Z	Z	Z	Z	Z
	NEXUS (w $\overline{\text{DRDY}}$)	TDI	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TDO	Z	Z	Z	Z	Z
		TCK	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TMS	Z	I (pull-up)	I (pull-up)	I (pull-up)	Z
		$\overline{\text{TRST}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{DRDY}}$	Z	Z ^{*7}	Z	H ^{*4}	Z
	LPD-4 pin	TDI/LPDI	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TDO/LPDO	Z	Z ^{*8}	Z	H	Z
		TCK/LPDCCKI	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TMS/JP0_3	Z	Z	Z	Z	Z
		$\overline{\text{TRST}} / \overline{\text{LPDRST}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{DRDY}} / \overline{\text{LPDCKO}}$	Z	Z ^{*8}	Z	L	Z
	GPIO (Write I/F)	TDI/FPDR	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TDO/FPDT	Z	Z	Z	Z	Z
		TCK/FPCK	I (pull-up)	I (pull-up)	I (pull-up)	I (pull-up)	Z
TMS/JP0_3		Z	Z	Z	Z	Z	
$\overline{\text{TRST}}$		I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	
$\overline{\text{DRDY}} / \text{JP0_5}$		Z	Z	Z	Z	Z	

Table 2.66 Pin State in Serial Programming Mode

Pin Function			Pin State					Power-off Standby Mode
Category	IF Mode	Pin Name	Condition					
			A	B	C, J	D, K	G, L, M*6	
Debug System*1	NEXUS (wo $\overline{\text{DRDY}}$)*2	TDI	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TDO	Z	Z	Z	Z	Z	Z
		TCK	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TMS	Z	Z	Z	Z	I (pull-up)	Z
		$\overline{\text{TRST}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{DRDY}}$ / JP0_5	Z	Z	Z	Z	Z	Z
	NEXUS (w $\overline{\text{DRDY}}$)*2	TDI	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TDO	Z	Z	Z	Z	Z	Z
		TCK	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TMS	Z	Z	Z	Z	I (pull-up)	Z
		$\overline{\text{TRST}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{DRDY}}$	Z	Z	Z	Z	H*4	Z
	LPD-4pin*2	TDI/LPDI	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TDO/LPDO	Z	Z	Z	Z	H	Z
		TCK/LPDCLKI	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TMS/JP0_3	Z	Z	Z	Z	Z	Z
		$\overline{\text{TRST}}$ / $\overline{\text{LPDRST}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{DRDY}}$ / LPDCLKO	Z	Z	Z	Z	L	Z
	GPIO (Write I/F)	TDI/FPDR	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TDO/FPDT	Z	Z	Z	Z	Z	Z
		TCK/FPCK	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)	Z
		TMS/JP0_3	Z	Z	Z	Z	Z	Z
		$\overline{\text{TRST}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{DRDY}}$ / JP0_5	Z	Z	Z	Z	Z	Z

Note: I: Input

O: Output

H: Output High

L: Output Low

Z: High impedance

Pull-up: On-chip pull-up resistor

Pull-down: On-chip pull-down resistor

Note: The port pin's input structures is designed in a way that during reset and after reset, unconnected pins do not tend to oscillate, draw excessive current, or exhibit unintended crosscoupling effects on other port pins.

Note 1. Boundary Scan I/F mode is not support with Serial Programming Mode.

Note 2. This condition is support with CAN/CAN-FD debug I/F mode in Serial Programming Mode only (About CAN/CAN-FD mode, see **Section 5, Operating Modes** for details.).

Note 3. When the Option Byte PFC_RESO_CFG = 0, $\overline{\text{RES_OUT}}$ pin output is changed to high after internal reset release. (See **Section 2.6.8, RES_OUT Function** for details.)

Note 4. $\overline{\text{DRDY}}$ pin status depends on the NEXUS/JTAG state. WAIT state is high, and IDLE state is low.

Note 5. $\overline{\text{ERROROUT_M}}$ pin status depends on the output signal from Error Control Module. (See **Section 39, Error Control Module (ECM)** for details.)

Note 6. The condition M is the same as the previous condition just before system reset 2 is asserted.

Note 7. When the status is changed from condition F, $\overline{\text{DRDY}}$ pin depends on the NEXUS/JTAG state. WAIT state is high, and IDLE state is low.

Note 8. When the status is changed from condition E or F or M, TDO and $\overline{\text{DRDY}}$ pins are the same as the previous condition.

2.9 Handling of The Unused Pins

Examples of the handling of unused pins are listed in following table.

Table 2.67 Examples of the Handling of Unused Pins (1/3)

Classification	Pin Name	IO	Examples of Handling of Unused Pins	On-chip Pull-down/Pull-up Resistor
Clock	XTAL	O	This pin can be left open.	None
	EXTAL	I	(This pin will always be used.)	None
System control	RES_IN	I	(This pin will always be used.)	This pin is equipped with an internal pull-down resistor.
	RES_OUT	O	This pin can be left open.	None
	MD0	I	(This pin will always be used.)	This pin is equipped with an internal pull-down resistor.
	MD1	I	This pin must be connected to VSS via individual resistor.	This pin is equipped with an internal pull-down resistor.
Interrupt	NMI	I	This pin must be connected to VSS via individual resistor.	This pin is equipped with an internal pull-down resistor.
ECM	ERROROUT_M	O	This pin can be left open.	None
General input/output port	Pn_m	IO	<p>[Input mode] These pins can be left open and the settings are to disable input (values after reset are PMCn_m = 0, PMn_m = 1, PIBCn_m = 0) These pins can be left open and the settings are to enable on-chip pull-up/pull-down resistors (use PUn_m, PDn_m) These pins must be connected to power supply or ground of each pin via individual resistor.</p> <p>[Output mode] These pins can be left open.</p>	The pins are equipped with internal pull-up or pull-down resistors which can be enabled by register settings.
ADC, DS-ADC	Analog input-only AN0xx, AN1xx, AN2xx, AN3xx (excluding AN00x, AN01x, AN20x, AN21x)	I	These pins can be left open.	None
	Alternative analog input and digital input. AN00x, AN01x, AN20x, AN21x	I	These pins can be left open with the settings are to disable input (values after reset are PMCn_m = 0, and PIBCn_m = 0)	None

Table 2.67 Examples of the Handling of Unused Pins (2/3)

Classification	Pin Name	IO	Examples of Handling of Unused Pins	On-chip Pull-down/Pull-up Resistor	
Debug	AUORES ^{*1}	I	Refer to Section 2.9.1, Handling of Thermal Area Pins.	Refer to Section 2.9.1, Handling of Thermal Area Pins.	
	CICREFF ^{*1}	I	Refer to Section 2.9.1, Handling of Thermal Area Pins.	Refer to Section 2.9.1, Handling of Thermal Area Pins.	
	CICREFN ^{*1}	I	Refer to Section 2.9.1, Handling of Thermal Area Pins.	Refer to Section 2.9.1, Handling of Thermal Area Pins.	
	TODP0 ^{*1}	O	Refer to Section 2.9.1, Handling of Thermal Area Pins.	Refer to Section 2.9.1, Handling of Thermal Area Pins.	
	TODN0 ^{*1}	O	Refer to Section 2.9.1, Handling of Thermal Area Pins.	Refer to Section 2.9.1, Handling of Thermal Area Pins.	
	TDI/LPDI/JP0_0/FPDR	TDI	I	This pin can be left open. If it is connected, this pin must also be connected to VCC via an individual resistor	This pin is equipped with an internal pull-up resistor.
		LPDI	I	This pin can be left open. If it is connected, this pin must also be connected to VCC via an individual resistor	This pin is equipped with an internal pull-up resistor.
		JP0_0/FPDR	I	Do not select the JP0_0/FPDR pin function if the pin is not use in use.	(The pins are equipped with internal pull-up or pull-down resistors which can be enabled by register settings.) The internal pull-up resistor is enabled in booting up by the boot firmware.
	TDO/LPDO / JP0_1/FPDT	TDO	O	This pin can be left open	None
		LPDO	O	This pin can be left open	None
		JP0_1/FPDT	O	Do not select the JP0_1/FPDT pin function if the pin is not use in use.	(The pins are equipped with internal pull-up or pull-down resistors which can be enabled by register settings.) None
	TCK/LPDCLKI/JP0_2/FPCK	TCK	I	This pin can be left open. If it is connected, this pin must also be connected to VCC via an individual resistor	This pin is equipped with an internal pull-up resistor.
		LPDCLKI	I	This pin can be left open. If it is connected, this pin must also be connected to VCC via an individual resistor	This pin is equipped with an internal pull-up resistor.
		JP0_2/FPCK	I	Do not select the JP0_2/FPCK pin function if the pin is not use in use.	(The pins are equipped with internal pull-up resistors which can be enabled by register settings.) None
	TMS / JP0_3	TMS	I	This pin can be left open. If it is connected, this pin must also be connected to VCC via an individual resistor	This pin is equipped with an internal pull-up resistor.
TRST / LPDRST		I	This pin must be connected to VSS via individual resistor.	This pin is equipped with an internal pull-down resistor.	
DRDY / LPDCKLO / JP0_5	DRDY	O	This pin can be left open.	None	
	LPDCKLO	O	This pin can be left open.	None	
NC	NC	—	This pin must be open and soldering with printed circuit board is recommended.	None	
VSS(NC)	VSS(NC)	—	Connecting this pin to VSS is recommended from the point of view of power stabilization. (It is connected to VSS in PKG.) However, it can be left open. Soldering with printed circuit board is recommended even if open.	None	
A0VSS(NC)	A0VSS(NC)	—	Connecting this pin to A0VSS is recommended from the point of view of power stabilization. (It is connected to A0VSS in PKG.) However, it can be left open. Soldering with printed circuit board is recommended even if open.	None	
ADSVSS(NC)	ADSVSS(NC)	—	Connecting this pin to ADSVSS is recommended from the point of view of power stabilization. (It is connected to ADSVSS in PKG.) However, it can be left open. Soldering with printed circuit board is recommended even if open.	None	
VDD(NC)	VDD(NC)	—	Refer to Section 2.9.1, Handling of Thermal Area Pins.	Refer to Section 2.9.1, Handling of Thermal Area Pins.	

Table 2.67 Examples of the Handling of Unused Pins (3/3)

Classification	Pin Name	Examples of Handling of Unused Pins	On-chip Pull-down/Pull-up Resistor
Power supply	E0VCC/E1VCC/E2VCC	(This pin will always be used.)	—
	SYSVCC	(This pin will always be used.)	—
	VCC	(This pin will always be used.)	—
	VDD	(This pin will always be used.)	—
	EMUVCC ^{*1}	Refer to Section 2.9.1, Handling of Thermal Area Pins.	Refer to Section 2.9.1, Handling of Thermal Area Pins.
	EMUVDD ^{*1}	Refer to Section 2.9.1, Handling of Thermal Area Pins.	Refer to Section 2.9.1, Handling of Thermal Area Pins.
	RAMSVCL	(This pin will always be used.)	—
	VSS	(This pin will always be used.)	—
	LVDVCC	This pin must be connected to E0VCC with the settings are to disable LVDS for RHSIF (values after reset are LVDSCTRLA = 0000 0000 _h).	—
	ADSVCC, ADSVSS, ADSVREFH, ADSVREFL, ADSVCL	Connections when the DS-ADC and CADC is not in use but the ADCH module is in use are listed below. ADSVCC: This pin must be connected to A0VCC. ADSVSS: This pin must be connected to A0VSS. ADSVREFH: This pin must be connected to A0VCC. ADSVREFL: This pin must be connected to A0VSS. ADSVCL: This pin can be left open. Connections when neither the DSADC nor CADC nor the ADCH module is in use are listed below. ADSVCC: This pin must be connected to E0VCC. ADSVSS: This pin must be connected to VSS. ADSVREFH: This pin must be connected to E0VCC. ADSVREFL: This pin must be connected to VSS. ADSVCL: This pin can be left open	—
A0VCC A1VCC A2VCC A3VCC A0VREFH A1VREFH A2VREFH A3VREFH A0VSS A1VSS A2VSS A3VSS	Connections when the ADCH module is not in use but the DS-ADC or CADC is in use are listed below. A0VCC: This pin must be connected to ADSVCC. A0VSS: This pin must be connected to ADSVSS. A0VREFH: This pin must be connected to ADSVCC. A1VCC: This pin must be connected to ADSVCC. A1VSS: This pin must be connected to ADSVSS. A1VREFH: This pin must be connected to ADSVCC. A2VCC: This pin must be connected to ADSVCC. A2VSS: This pin must be connected to ADSVSS. A2VREFH: This pin must be connected to ADSVCC. A3VCC: This pin must be connected to ADSVCC. A3VSS: This pin must be connected to ADSVSS. A3VREFH: This pin must be connected to ADSVCC. Connections when neither the DSADC nor the ADCH nor the CADC module is in use are listed below. A0VCC: This pin must be connected to E0VCC. A0VSS: This pin must be connected to VSS. A0VREFH: This pin must be connected to E0VCC. A1VCC: This pin must be connected to E0VCC. A1VSS: This pin must be connected to VSS. A1VREFH: This pin must be connected to E0VCC. A2VCC: This pin must be connected to E0VCC. A2VSS: This pin must be connected to VSS. A2VREFH: This pin must be connected to E0VCC. A3VCC: This pin must be connected to E0VCC. A3VSS: This pin must be connected to VSS. A3VREFH: This pin must be connected to E0VCC.	—	

Note: For the correspondences between the various pins and the power supply or ground, see *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware.*

Note: Use a resistor with a value of at least 1 kΩ for externally pulling pins up or down.

Note: Take care regarding the resistive division effect when using an external resistor to pull up or pull down a pin that is equipped with an internal pull-up or pull-down resistor.

Note 1. The pin is implemented only in E2x-FCC1.

2.9.1 Handling of Thermal Area Pins

2.9.1.1 The Case of using Emulation/Instrumentation RAM (SYSVCC and EMUVDD are always supplied)

Emulation/Instrumentation RAM is available.

Emulation/Instrumentation RAM can be retained when power-off standby (see **Section 16.2.1, Power Off Standby Mode** for details).

Aurora I/F is can not be used ($\overline{\text{AURORES}} = \text{VSS}$).

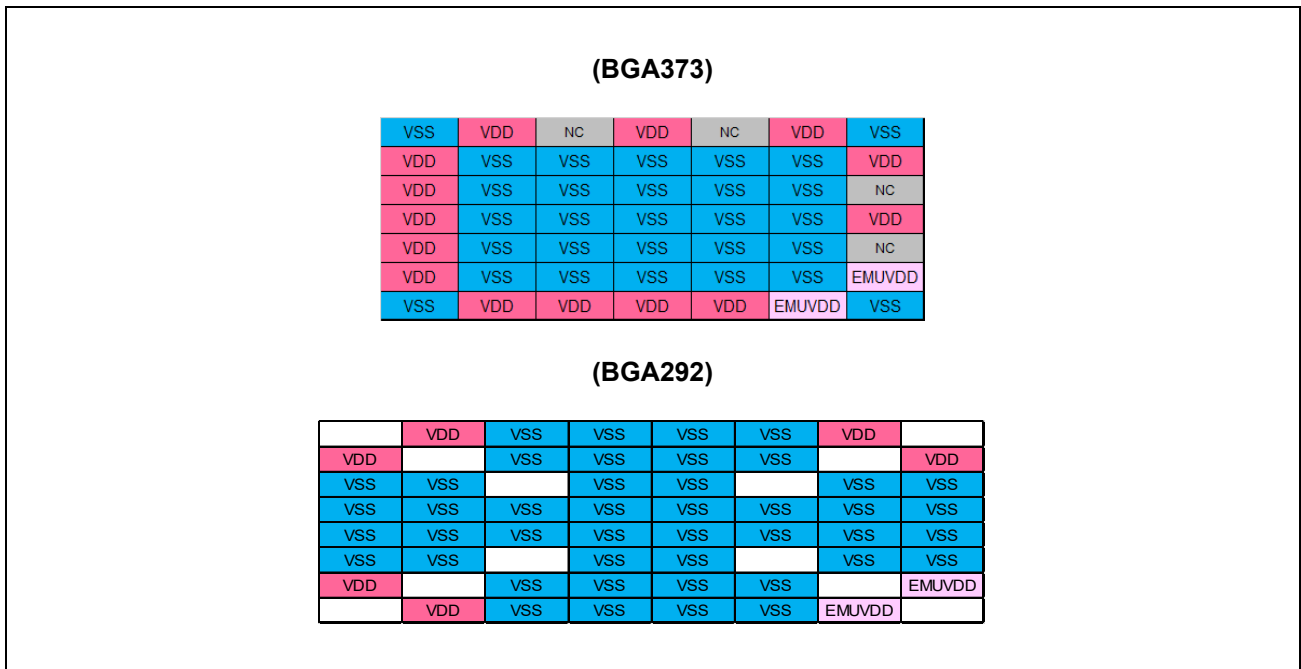


Figure 2.20 Thermal Area Pin Connection Diagram

Table 2.68 Example of the Handling of Unused Pins for Thermal Area

Classification	Pin Name	Debug tool is used.	Debug tool is not used.*1	On-chip Pull-down / Pull-up Resistor
Power supply	EMUVDD	(This pin will always be used.)	This pin can be left open.	—

Note 1. It indicate that Emulation/Instrumentation RAM is not need to be retained. If some of these pins are left open, then all of these pins must be left open.

2.9.1.2 The Case of using Emulation/Instrumentation RAM and Aurora I/F (EMUVCC and EMUVDD are always supplied)

Emulation/Instrumentation RAM and Aurora I/F is available.

Emulation/Instrumentation RAM can be retained when Emulation/Instrumentation RAM and Aurora Retention Mode. (see **Section 16.2.2, Emulation/Instrumentation RAM and Aurora Retention Mode** for details).

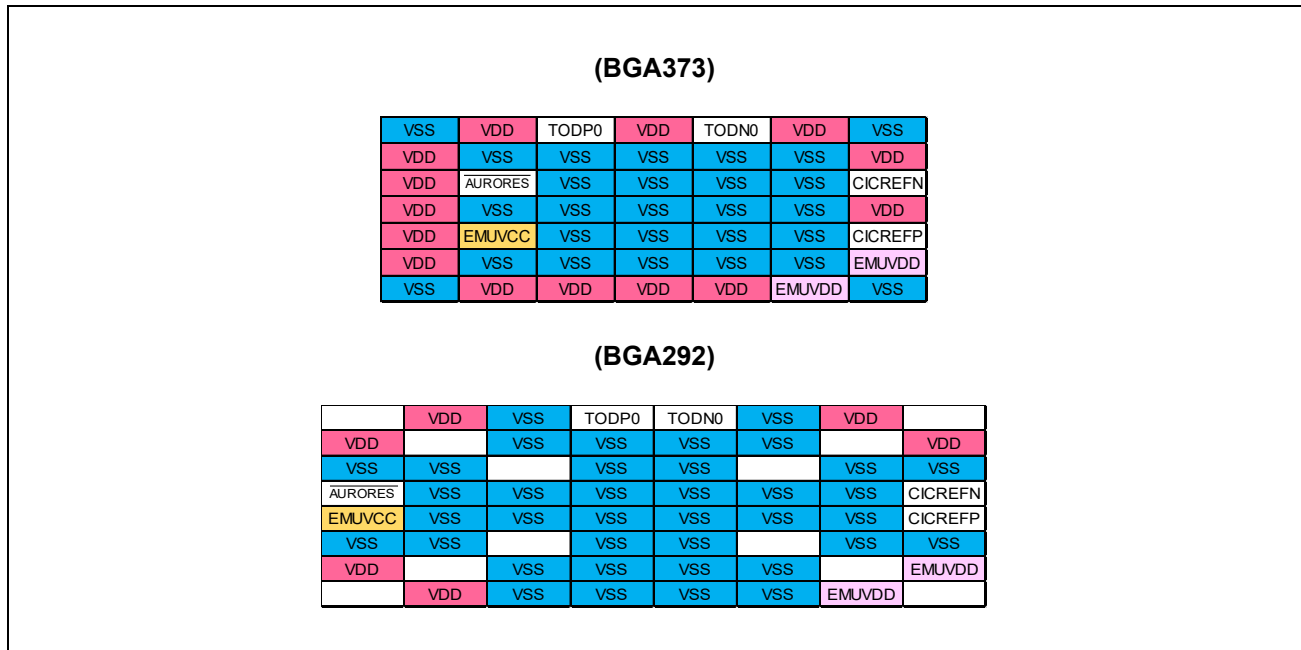


Figure 2.21 Thermal Area Pin Connection Diagram

Table 2.69 Examples of the Handling of Unused Pins for Thermal Area

Classification	Pin Name	Debug tool is used.	Debug tool is not used. *1	On-chip Pull-down / Pull-up Resistor
Debug	AURORES	(This pin will always be used.)	This pin can be left open.	This pin is equipped with an internal pull-down resistor.
	CICREFP	(This pin will always be used.)	This pin can be left open.	—
	CICREFN			
	TODP0 TODN0			
Power supply	EMUVCC	(This pin will always be used.)	This pin can be left open.	—
	EMUVDD	(This pin will always be used.)	This pin can be left open.	—

Note 1. It indicate that AURORA I/F is not used and Emulation/Instrumentation RAM is not need to be retained. If some of these pins are left open, then all of these pins must be left open.

2.9.1.3 The Case of using Emulation/Instrumentation RAM (EMUVCC and EMUVDD are always supplied)

Emulation/Instrumentation RAM is available.

Emulation/Instrumentation RAM can be retained when Emulation/Instrumentation RAM and Aurora Retention Mode. (see **Section 16.2.2, Emulation/Instrumentation RAM and Aurora Retention Mode** for details).

Aurora I/F can not be used ($\overline{\text{AURORES}} = \text{VSS}$).

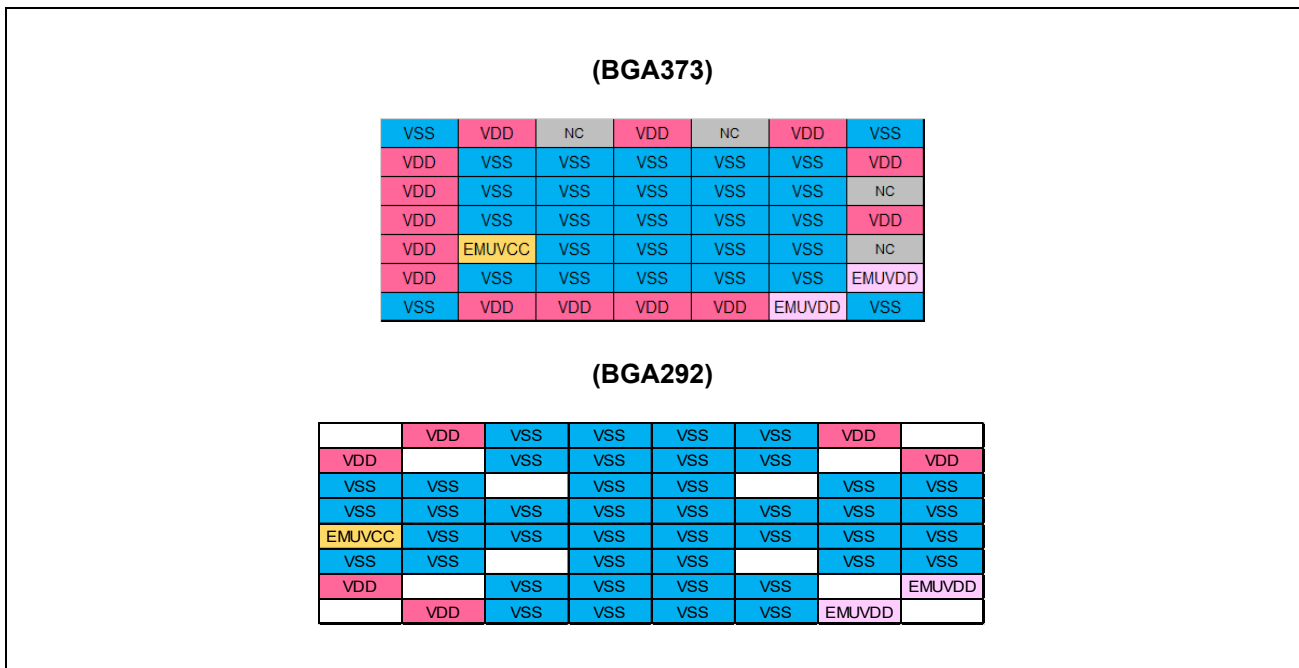


Figure 2.22 Thermal Area Pin Connection Diagram

Table 2.70 Examples of the Handling of Unused Pins for Thermal Area

Classification	Pin Name	Debug tool is used.	Debug tool is not used.*1	On-chip Pull-down / Pull-up Resistor
Power supply	EMUVCC	(This pin will always be used.)	This pin can be left open.	—
	EMUVDD	(This pin will always be used.)	This pin can be left open.	—

Note 1. It indicate that the Emulation/Instrumentation RAM and Aurora configuration are not need to be retained. If some of these pins are left open, then all of these pins must be left open.

2.9.1.4 The Case of Pin Assignment for Mass Production

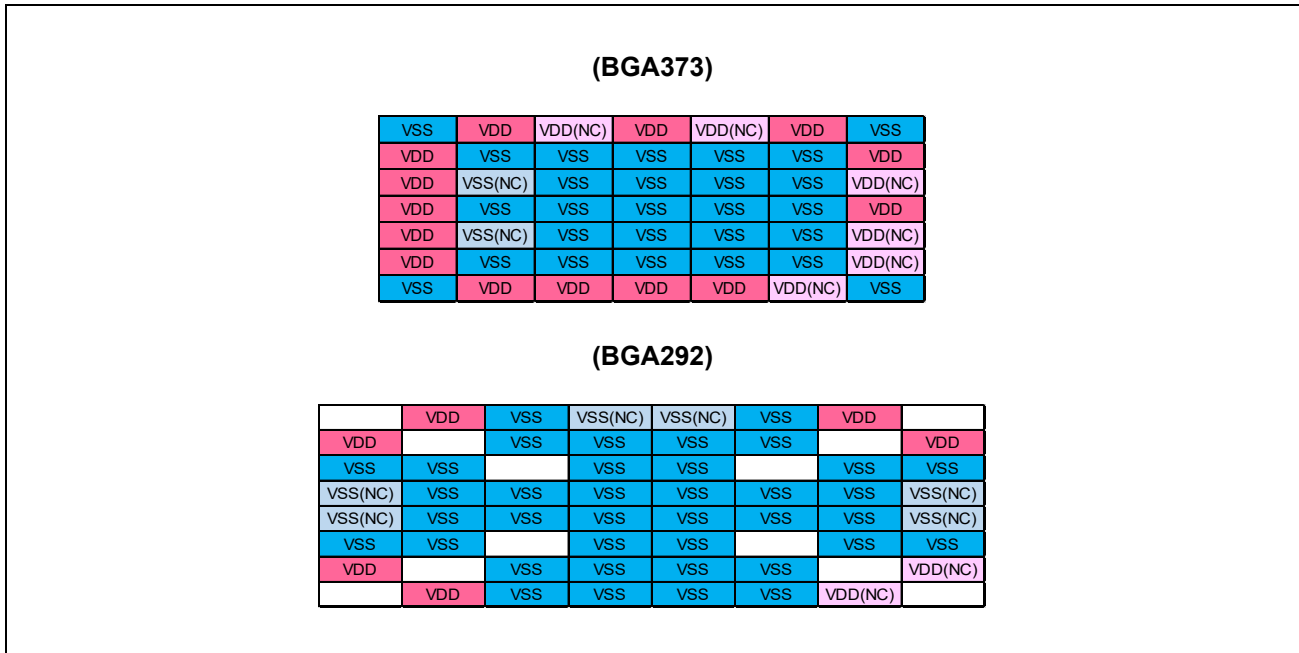


Figure 2.23 Thermal Area Pin Connection Diagram

Table 2.71 Examples of the Handling of Unused Pins for Thermal Area

Classification	Pin Name	Examples of Handling of Unused Pins.	On-chip Pull-down / Pull-up Resistor
Power supply	VDD(NC)	Connecting this pin to VDD is recommended from the point of view of power stabilization. (It is connected to VDD in PKG.) However, it can be left open. Soldering with printed circuit board is recommended even if open.	—
	VSS(NC)	Connecting this pin to VSS is recommended from the point of view of power stabilization. (It is connected to VSS in PKG.) However, it can be left open. Soldering with printed circuit board is recommended even if open.	—

Section 3 CPU System

3.1 Overview

3.1.1 Block Diagram

For the block diagram of the CPU system, see **Figure 1.5** and **Figure 1.6** in **Section 1, Overview**.

The CPU system consists of the following modules.

CPU0 (PE0), CPU1 (PE1)

The RH850 G4MH core is included as the main CPU. The CPUs also include a checker core for safety assurance.

ICUMD

Refer to *the E2x ICUMD User's Manual*.

Local RAM

Each PE has a high-speed accessible local RAM.

Cluster RAM

The cluster RAM is a mass on-chip RAM that all PEs share. Some areas of this RAM function as the retention RAM.

Code Flash

A mass code flash is included for program storage. PE0, PE1 and ICUMD share the code flash and they are connected with each other via the code flash interface.

Emulation RAM

This is RAM to emulate the code flash. The programs can be replaced by control from an external tool without rewriting the code flash. It is supported in E2x-FCC1.

Data Flash

This is a flash memory rewritable by the CPU.

P-Bus and H-Bus and I-Bus

Peripheral IPs are connected on P-Bus, H-Bus and I-Bus.

E2x-FCC1: The P-Bus consists of 10 groups, Peripheral Group 0 to 9.

The H-Bus consists of 3 groups, H-Bus Group 0 to 2.

The I-Bus consists of 4 groups, I-Bus Group 0 to 3.

E2M: The P-Bus consists of 10 groups, Peripheral Group 0 to 9.

The H-Bus consists of 3 groups, H-Bus Group 0 to 2.

The I-Bus consists of 4 groups, I-Bus Group 0 to 3.

Refer to each section on peripheral IPs for information on bus groups.

INTC1, INTC2

INTC1 is an interrupt controller exclusive to each PE. INTC2 is a common interrupt controller that all PEs can share, which is able to set the destination PE for binding interrupt requests by the registers.

DMA

Two DMA transfer modules, sDMAC and DTS, are included. DTS includes the checker logic for safety assurance.

3.2 CPU

3.2.1 Core Functions

3.2.1.1 Features

Table 3.1 lists the features of the RH850G4MH core. See *the RH850G4MH User's Manual: Software*.

Table 3.1 Features of the RH850G4MH Core

Item	Feature
CPU	<ul style="list-style-type: none"> • High performance 32-bit architecture for embedded control • 32-bit internal data bus • Thirty-two 32-bit general-purpose registers • RISC-type instruction sets <ul style="list-style-type: none"> – Long/short type load/store instructions – Three-operand instructions – Instruction sets based on C • CPU operating modes <ul style="list-style-type: none"> – User mode, supervisor mode • Address space: 4-GB linear space for both data and instructions • Out-of-Order execution
Coprocessor	<ul style="list-style-type: none"> • A floating point operation coprocessor (FPU) <ul style="list-style-type: none"> – Supports single precision (32-bits) and double precision (64-bits). – Supports IEEE754-compliant data types and exceptions. – Rounding mode: Nearest, 0 direction, $+\infty$ direction, and $-\infty$ direction – Handling of non-normalized numbers: These are truncated to 0, or an exception is reported to comply with IEEE754. • An extended floating-point operation coprocessor (FXU) <ul style="list-style-type: none"> – Supports 4 single-precision (32-bit) parallel operations (FP-SIMD) – Supports IEEE754-compatible data types and exceptions – Rounding modes: Nearest, 0 direction, $+\infty$ direction, and $-\infty$ direction – Handling of non-normalized numbers: These are truncated to 0, or an exception is reported to comply with IEEE754.
Exception/Interrupt	<ul style="list-style-type: none"> • 16-level interrupt priority that can be specified for each channel • Vector selection method that can be selected according to performance requirements and the amount of consumed memory <ul style="list-style-type: none"> – Direct branch method exception vector (direct vector method) – Address-table-referencing indirect branch method exception vector (table reference method) • Support for high-speed context save and restore processing on interrupt by using dedicated instructions (PUSHSP, POPSP) • Support for high-speed context save on interrupt by using the register bank feature • Support for restoration from the register bank using a dedicated instruction (RESBANK)
Memory Management	<ul style="list-style-type: none"> • Memory protection function (MPU): 24 areas settable.
Cache	<ul style="list-style-type: none"> • Instruction cache 16 KB

3.2.2 Processor Model

This CPU defines a processor model that has basic operation functions, registers, and an exception management function.

This section describes the unique features of the processor model of this CPU.

3.2.2.1 CPU Operating Modes

This CPU has two operating modes of the supervisor mode (SV) and the user mode (UM). Whether the system is in supervisor mode or user mode is indicated by the UM bit in the PSW register.

- Supervisor mode (PSW.UM = 0): All hardware functions can be managed or used.
- User mode (PSW.UM = 1): The usable hardware functions are restricted.

(1) Definition of CPU Operating Modes

(a) Supervisor Mode (SV)

All hardware functions can be managed or used in this mode. The system always starts up in supervisor mode after a reset.

(b) User Mode (UM)

This operating mode makes up a pair with the supervisor mode. In user mode, address spaces to which access is permitted by the supervisor and the system registers defined as user resources can be used. Supervisor-privileged instructions cannot be executed and result in exceptions.

Restriction in user mode (PSW.UM = 1)

- Privileged instruction violations due to SV-privileged-instruction operating restrictions
(→ PIE exceptions)

For details about privileged-instruction operating restrictions, see **Section 3.2.2.1(3), CPU Operating Mode and Privileges**.

(2) CPU Operating Mode Transition

The CPU operating mode changes due to three events.

(a) Change due to Acknowledging an Exception

When an exception is acknowledged in user mode, the CPU operating mode changes to privileged mode.

When an exception is acknowledged in supervisor mode, the CPU operating mode remains as is.

(b) Change due to a Return Instruction

When a return instruction is executed, the PSW value is restored according to the value of the corresponding bit backed up to EIPSW and FEPSW.

(c) Change due to a System Register Instruction

The CPU operating mode changes when an LDSR instruction is used to directly overwrite the PSW operating mode bits.

CAUTION

The CPU operating mode cannot be changed in user mode because the higher-order 31 to 5 bits of the PSW register cannot be overwritten while the mode can be changed in supervisor mode. This CPU guarantees that if an LDSR instruction is used to update the PSW register, the new setting will be reflected when the subsequent instruction is executed. However, this CPU does not guarantee that the new setting will be reflected in the memory protection by the MPU for instruction fetch of the subsequent instruction. Therefore, for changing the higher-order 31 to 5 bits of the PSW register, it is recommended to use a return instruction. For details, see **Section 3.2.7.3, Hazard Management after System Register Update**.

(3) CPU Operating Mode and Privileges

In this CPU, the usable functions can be restricted according to usage permission settings for specific resources and the CPU operating mode. Specific instructions (including instructions that update specific system registers) can only be executed in the defined operating mode. The permissions necessary to execute these specific instructions are called “privileges”. In operating modes that do not have privileges, these instructions are not executed and exceptions occur.

This CPU defines the following two types of privileges (usage permission).

- Supervisor (SV) privilege: Privilege necessary for important system resources operation, fatal error processing, and user-mode program execution management
- Coprocessor use permissions: Permissions necessary to use a coprocessor

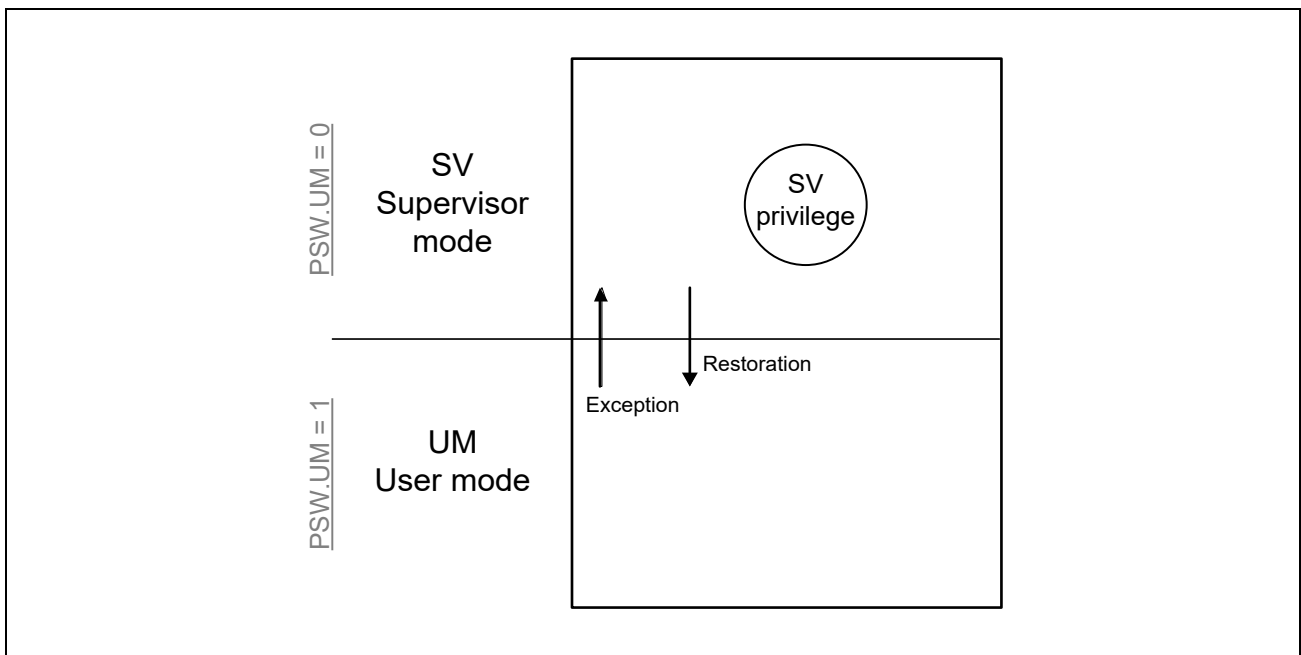


Figure 3.1 CPU Operating Modes and Privileges

(a) Supervisor Privilege (SV Privilege)

The privilege necessary to perform the operation for important system resources, fatal error processing, and user-mode program execution management is called the supervisor privilege (SV privilege). This privilege is available in supervisor mode. The SV privilege is generally necessary to execute instructions used to perform the operation for important system resources, and these instructions are sometimes called SV privileged instructions.

(b) Coprocessor Use Permissions

Regardless of the CPU operating mode, it is possible to specify whether coprocessors can be used.

The CU2 to CU0 bits in the PSW register are used by a supervisor to specify whether coprocessors can be used by each program. If the CU 2 to CU0 bits are not set to 1, a coprocessor unusable exception occurs when the corresponding coprocessor instruction is executed or the system register is accessed.

If no coprocessor is installed, it is not possible to set the corresponding CU bits to 1. The setting of the CU2 to CU0 bits is valid regardless of the CPU operating mode, and, if the supervisor accesses coprocessor system registers, it is necessary to set the CU2 to CU0 bits to enable coprocessor use.

(c) Operation when there is a Privilege Violation

When a privileged instruction is executed by someone who does not have the required privilege, or when a system register for which access permission is specified is accessed by someone who does not have the required permission, a PIE exception or UCPOP exception occurs.

The relationship between instructions that can be executed according to operating mode and use permission are shown in **Table 3.2**, and the relationship between system registers that can be accessed according to operating mode and use permission are shown in **Table 3.3**.

Table 3.2 Operation when Violating Execution Permission

Instruction		PSW				Whether Instructions can be Executed
Execution Permission	Classification	UM	CU2	CU1	CU0	
SV privilege	*1	0	—	—	—	Possible
		1	—	—	—	Not possible (PIE exception occurs)
User	Coprocessor 0 instruction	—	—	—	0	Not possible (UCPOP exception occurs)
		—	—	—	1	Possible
	Coprocessor 1 instruction	—	—	0	—	Not possible (UCPOP exception occurs)
		—	—	1	—	Possible
	Coprocessor 2 instruction	—	0	—	—	Not possible (UCPOP exception occurs)
		—	1	—	—	Possible
Other than above	—	—	—	—	Possible	

Remark: —: 0 or 1

Note 1. Coprocessor instructions with SV privilege are not defined in this CPU.

Table 3.3 Operation When Violating Access Permission to System Registers

System Register		PSW				Whether Instructions can be Executed
Access Permission	Classification	UM	CU2	CU1	CU0	
SV permission	Coprocessor 0 Permission	0	—	—	0	Inaccessible (UCPOP exception occurs)
		0	—	—	1	Accessible
		1	—	—	0	Inaccessible (UCPOP exception occurs)
		1	—	—	1	Inaccessible (PIE exception occurs)
	Coprocessor 1 Permission	0	—	0	—	Inaccessible (UCPOP exception occurs)
		0	—	1	—	Accessible
		1	—	0	—	Inaccessible (UCPOP exception occurs)
		1	—	1	—	Inaccessible (PIE exception occurs)
	Coprocessor 2 Permission	0	0	—	—	Inaccessible (UCPOP exception occurs)
		0	1	—	—	Accessible
		1	0	—	—	Inaccessible (UCPOP exception occurs)
		1	1	—	—	Inaccessible (PIE exception occurs)
	Other than above	0	—	—	—	Accessible
		1	—	—	—	Inaccessible (PIE exception occurs)
UM permission	Coprocessor 0 Permission	—	—	—	0	Inaccessible (UCPOP exception occurs)
		—	—	—	1	Accessible
	Coprocessor 1 Permission	—	—	0	—	Inaccessible (UCPOP exception occurs)
		—	—	1	—	Accessible
	Coprocessor 2 Permission	—	0	—	—	Inaccessible (UCPOP exception occurs)
		—	1	—	—	Accessible
	Other than above	—	—	—	—	Accessible

Remark: —: 0 or 1

(4) Halt State by a HALT Instruction

Executing a HALT instruction brings the CPU core to a HALT state.

A HALT state is terminated when a request for a terminating exception is made to the CPU. At this point, if the conditions for acknowledging the terminating exception are met, control is transferred to the exception handler address. Even if they are not met, the HALT state is terminated when the request is made; therefore, operation is resumed from the instruction following the HALT instruction.

(5) Temporary Halt State by a SNOOZE instruction

A SNOOZE instruction is an instruction to reduce the use of bus bandwidth during a spinlock. After this instruction is completed, the CPU core is brought to a temporary halt state to restrict the execution of the subsequent instructions. Programmers can avoid the unnecessary use of bus bandwidth that results from short-term repetition of a locking process, by inserting this instruction into a spinlock loop.

A temporary halt state is terminated if any of the following conditions is met:

- After the time specified by the SNZCFG register elapsed, the execution of the subsequent instructions is resumed.
- The occurrence of a terminating exception causes control to be transferred to the exception handler address. If the conditions for acknowledging the exception are not met, execution is resumed from the instruction following the SNOOZE instruction. Also, if the exception is acknowledged, the instruction following the SNOOZE instruction is interrupted, and the PC restored indicates the instruction following the SNOOZE instruction.

3.2.2.2 Instruction Execution

The instruction execution flow of this CPU is shown below.

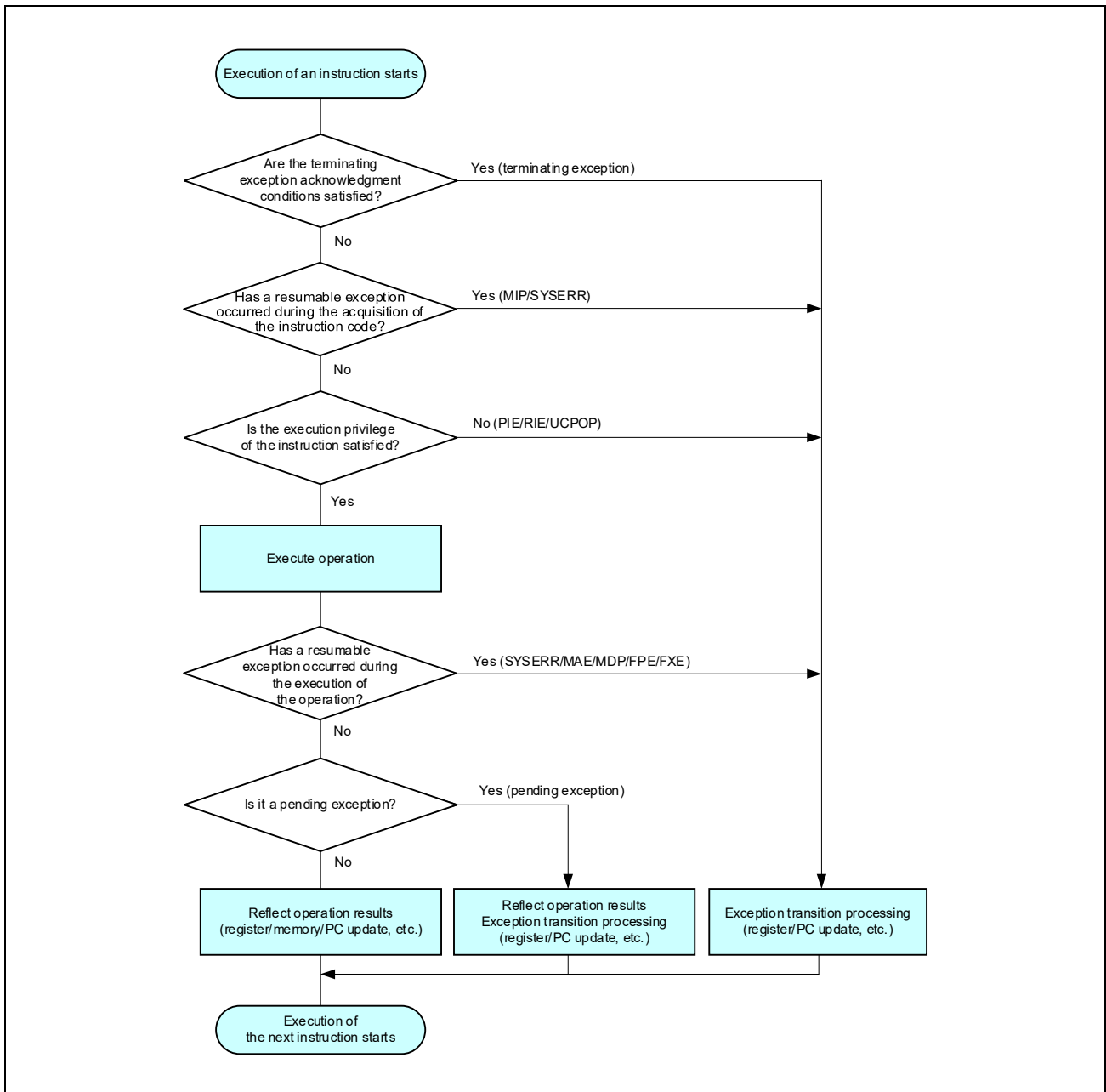


Figure 3.2 Instruction Execution Flow

If terminating exceptions can be acknowledged, if an exception is detected during the acquisition of instruction code, or if the execution privilege of the instruction is not satisfied, an exception occurs before the instruction is executed. If a resumable exception occurs while the CPU is executing the operation, it interrupts the execution of the operation and acknowledges the exception. In these cases, the result of instruction operation is, in principle, not reflected in registers or memory and the CPU retains its state that is established before executing the instruction.*¹

For a pending exception such as a software exception, the exception is acknowledged after the result of instruction execution has been reflected.

For details of the types of exceptions, refer to **Section 3.2.2.3, Exceptions and Interrupts**.

Note 1. If an exception is acknowledged during the execution of the following instructions, intermediate results may be applied to memory or general-purpose registers. However, SP/EP is not updated.

- PREPARE, DISPOSE, PUSHSP, POPSP

3.2.2.3 Exceptions and Interrupts

Exceptions and interrupts are exceptional events that cause the program under execution to branch to another program. Exceptions and interrupts are triggered by various sources, including interrupts from peripherals and program abnormalities.

For details, see **Section 3.2.2.3, Exceptions and Interrupts**.

(1) Types of Exceptions

The exceptions of this CPU are divided into the following three types according to the purpose of the exceptions.

- Terminating exception
- Resumable exception
- Pending exception

(a) Terminating Exception

A terminating exception is an exception that is acknowledged by interrupting an instruction before the execution of the operation for the instruction. Terminating exceptions include interrupts, etc.

This type of exceptions, such as an interrupt or hardware error, start another program irrelevant to the currently processed program.

(b) Resumable Exception

A resumable exception is an exception that occurs during the execution of the operation for an instruction and that is acknowledged without the completion of the instruction. It is also referred to as a precise exception because it is precisely acknowledged without the completion of the relevant instruction or the execution of the subsequent instruction.

Unlike a terminating exception, a resumable exception occurs during the execution of an instruction, and cancel the execution of the instruction, and so it allows the instruction to be executed again after exception handling. Therefore, it is possible to perform complex memory management while maintaining the consistency of the logical operation of programs, by executing the same instruction again after appropriate configuration by exception handling.

(c) Pending Exception

A pending exception is an exception that occurs as a result of the execution of the operation for an instruction, and that is acknowledged after the completion of the instruction. Pending exceptions include software exceptions, etc.

The occurrence of a pending exception is defined as a normal operation of an instruction; therefore, unlike a resumable exception, a pending exception normally completes the instruction that caused it, and never re-executes the instruction. This exception is intended mainly for use in call gates, for example, for calling a management program.

(2) Exception Level

In this CPU, if an exception with a high degree of urgency occurs while another exception is being processed, the urgent exception will be processed. To make it possible to return to the interrupted exception handling after acknowledging the urgent exception, even if the context had not been saved to the memory, exception causes are managed in the following two hierarchical levels.

- EI level exception
- FE level exception

EI level exceptions are used for processing such as regular user processing, interrupt servicing, and OS processing. FE level exceptions are used to enable interrupts with a high degree of urgency for the system or exceptions from the memory management function that might occur during OS processing to be acknowledged even while an EI level exception is being processed.

3.2.2.4 Coprocessors

In this CPU, single-precision and double-precision floating-point unit (FPU) and extended floating-point operation unit (FXU) are incorporated. Note that these coprocessors may not be available depending on the specification of the product.

(1) Coprocessor Use Permissions

To execute a coprocessor instruction, permission to use the corresponding coprocessor instruction is necessary. Coprocessor use permissions are specified by the PSW.CU2 to PSW.CU0 bits, and, if an attempt is made to execute an instruction for which the corresponding coprocessor use permission is cleared to 0, a coprocessor unusable exception (UCPOP) occurs.

In the following cases, the values of the PSW.CU2 to CU0 bits are fixed at 0 and cannot be changed.

- Coprocessor functions are not incorporated in the product
- Coprocessor functions are made unavailable according to the functions of the product

(2) Correspondences between Coprocessor Use Permissions and Coprocessors

This CPU defines coprocessor use permissions to control the availability of the coprocessor for each program during CPU operation. There are three coprocessor use permissions (CU0 to CU2), and their correspondences with the coprocessors are shown in the following table. This CPU does not have a coprocessor function with the coprocessor use permission CU2. The coprocessor use permission CU2 is a function reserved for future CPUs that have compatibility with this CPU.

Table 3.4 Correspondences between Coprocessor Use Permissions and Coprocessors

Coprocessor Use Permission	Coprocessor	Exception Cause Code
CU0	Single-precision FPU	80 _H
	Double-precision FPU	
CU1	FXU	81 _H
CU2	Reserved	82 _H

(3) Coprocessor Unusable Exceptions

A coprocessor unusable exception occurs if an attempt is made to execute a coprocessor instruction or access a system register of the coprocessor without having the corresponding coprocessor use permission (PSW.CUn = 0).

For details about the opcodes of the coprocessor instructions, see *the RH850G4MH User's Manual: Software*. For details about the system registers of the coprocessors, see **Section 3.2.3.4, FPU Function Registers** and **Section 3.2.3.5, FXU Function Registers**.

If the register bank function is used, the FPSR register, which is a system register of the FPU, is accessed when the automatic context saving and the RESBANK instruction is executed. In this case, even if the coprocessor use permission is not given (PSW.CU0 = 0), a coprocessor unusable exception does not occur.

(4) System Registers

Some coprocessor functions have system registers as the part of their functions. The coprocessor use permission is necessary to access the system register of a coprocessor function. For some system registers, the supervisor privilege (SV privilege) is necessary in addition to the coprocessor use permission.

For details about the permissions necessary to access system registers, see **Section 3.2.2.5, Registers**.

3.2.2.5 Registers

This CPU defines program registers (general-purpose registers and the program counter PC) and system registers for controlling the status and storing exception information.

(1) Program Registers

The program registers include general-purpose registers (r0 to r31) and the program counter (PC).

Table 3.5 Program Registers

Category	Access Permission	Name
Program counter	UM	PC
General-purpose registers	UM	r0-r31

Note: Access to the registers with UM (user mode) access permission is always allowed.

For details about program registers, see **Section 3.2.3.1, Program Registers**.

(2) System Registers

System registers are placed in dedicated address spaces defined based on two types of address information: selection ID and register number. Up to 32 selection ID can be defined, and one selection ID includes up to 32 system registers. Therefore, up to 1024 system registers can be defined in the address spaces for system registers. Basically this CPU allocates selection ID as shown below:

- Selection ID 0 to 3: Registers related to basic functions
- Selection ID 4 and 5: Registers related to the memory management function
- Selection ID 10 and 11: Registers related to functions expanded from the legacy architecture
- Selection ID 12 and 13: Registers related to this CPU specific hardware functions
- Other ID: Reserved for future expansion of CPUs compatible with this CPU

For details about system registers, see the relevant sections in **Section 3.2.3, Register Set**.

(3) Register Updating

There are several methods used to update registers. Normally, no particular restrictions apply when updating register by using an instruction. However, when updating registers by using the following instructions, some restrictions might apply, depending on the operating mode.

- LDSR
- STSR

(a) LDSR and STSR

The LDSR and STSR instructions can access all the system registers.

However, if a system register is accessed without the proper permission, a PIE exception or UCPOP exception might occur. For details about the access permission for each register, see the description of system registers in **Section 3.2.3, Register Set**. For details about behaviors when a privilege violation occurs, see **Section 3.2.2.1(3), CPU Operating Mode and Privileges**.

Figure 3.3 shows the flow of executing the LDSR and STSR instructions.

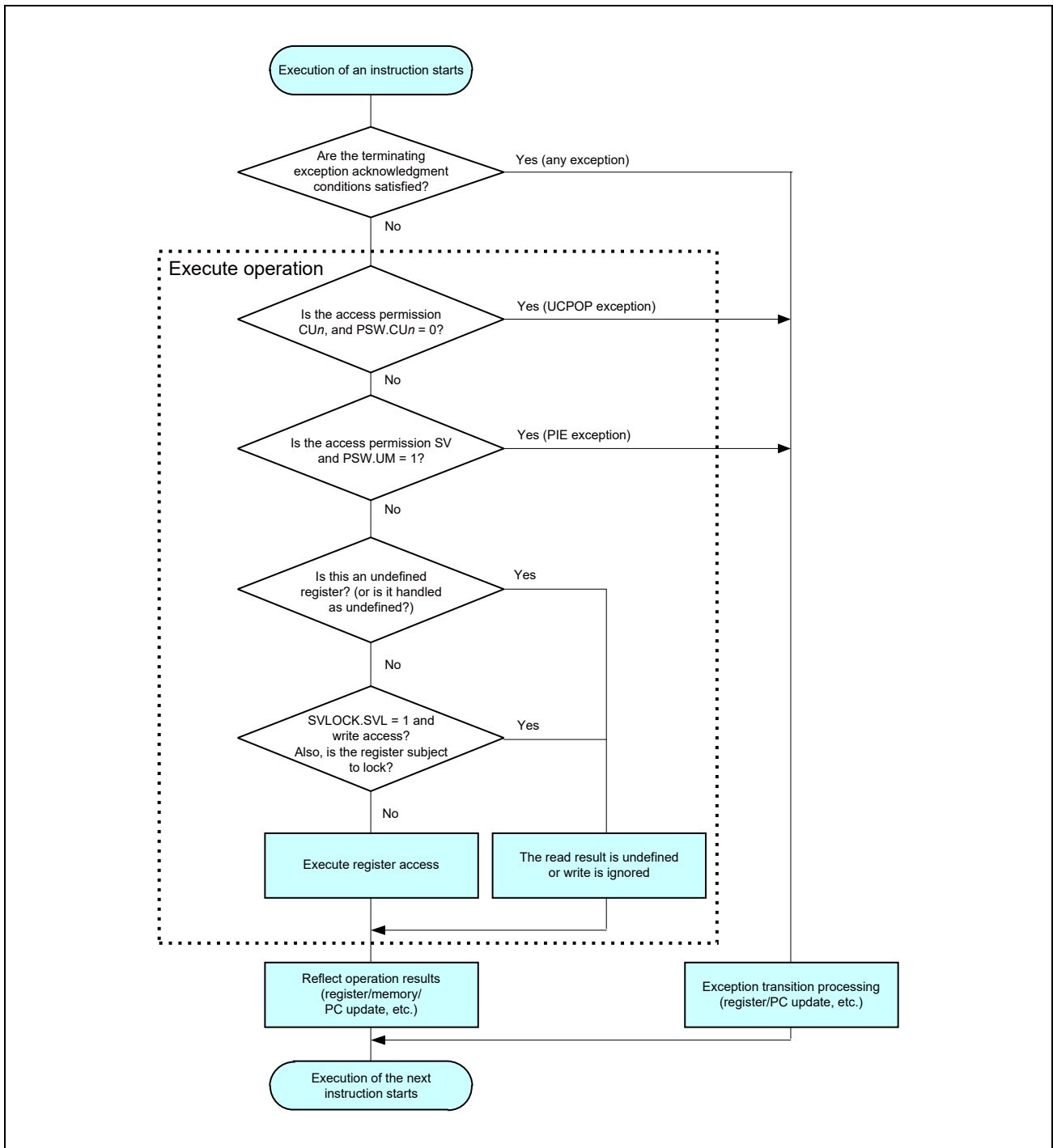


Figure 3.3 Flow of Executing the LDSR and STSR Instructions

(4) Accessing Undefined Registers

If a system register number without any register assigned is accessed or if an inaccessible register is accessed, the following results occur.

- Undefined registers are handled as having the SV permission. When they are accessed by an LDSR or STSR instruction in user mode (PSW.UM = 1), a PIE exception occurs.
- For a read operation, the read result is undefined. If the read value is used in a program, unexpected behaviors might occur.
- For a write operation, the write operation is ignored.

(5) Supervisor Lock Setting

If the SVLOCK.SVL bit is set (1), the following system registers cannot be updated even in supervisor mode:

SPID, MPM, MPLA, MPUA, MPAT, and MPIDn

However, if the SVLOCK.SVL bit is set (1), an attempt to update the system registers above does not cause a PIE exception.

Also, even if the SVLOCK.SVL bit is set (1), it is possible to read the system registers above.

This function is intended for the safety of systems that incorporate this CPU. It prevents a program running in supervisor mode from abruptly changing memory protection setting due to a programming error, etc., to make illegal memory access possible, and to impair safety.

3.2.2.6 Data Types

(1) Data Formats

This CPU handles data in little endian format. This means that byte 0 of a halfword or a word is always the least significant (rightmost) byte.

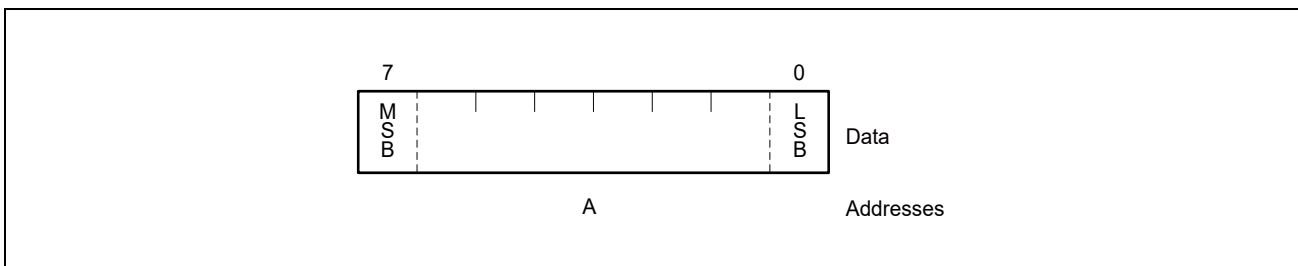
The supported data format is as follows.

- Byte (8-bit data)
- Halfword (16-bit data)
- Word (32-bit data)
- Double-word (64-bit data)
- Quad-word (128-bit data)
- Bit (1-bit data)

In this CPU, a memory access of double-word data is divided into two non-atomic memory accesses of word data, and a memory access of quad-word data is divided into four non-atomic memory accesses of word data.

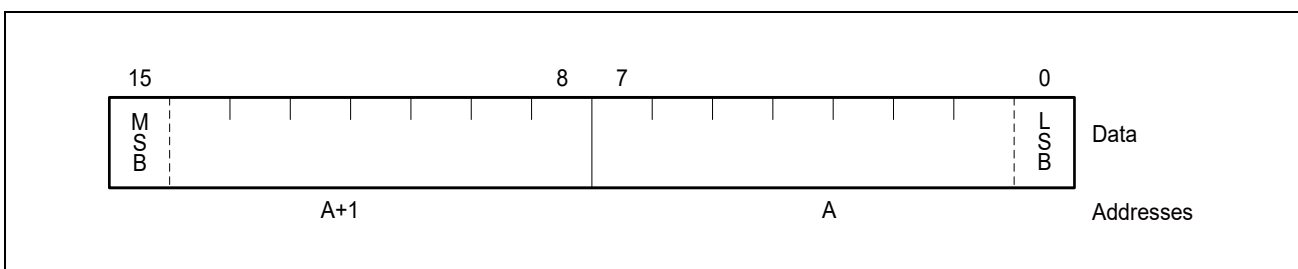
(a) Byte

A byte is 8 consecutive bits of data that starts from any byte boundary. Numbers from 0 to 7 are assigned to these bits, with bit 0 as the LSB (least significant bit) and bit 7 as the MSB (most significant bit). The byte address is specified as “A”.



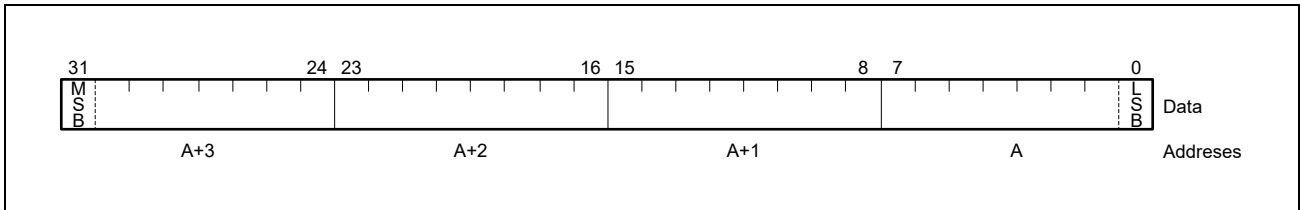
(b) Halfword

A halfword is two consecutive bytes (16 bits) of data that starts from any byte boundary. Numbers from 0 to 15 are assigned to these bits, with bit 0 as the LSB and bit 15 as the MSB. The bytes in a halfword are specified using address “A”, so that the two addresses comprise byte data of “A” and “A + 1”.



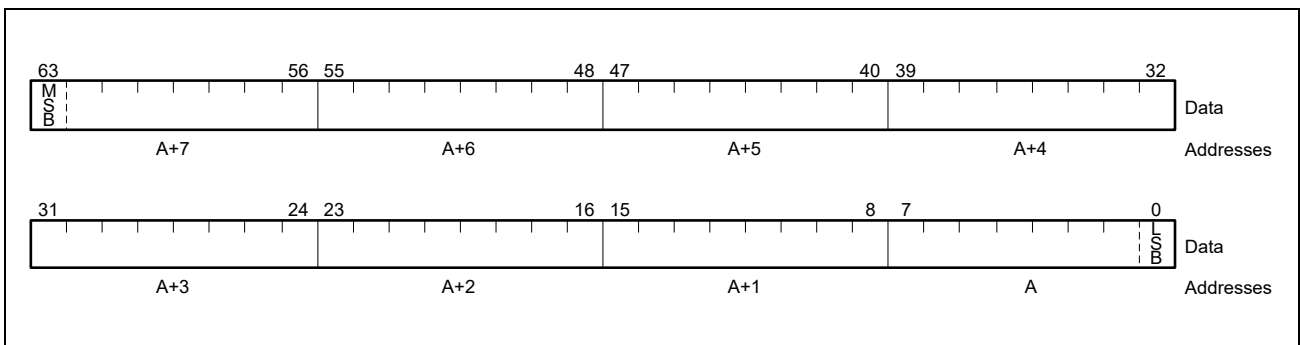
(c) Word

A word is four consecutive bytes (32 bits) of data that starts from any byte boundary. Numbers from 0 to 31 are assigned to these bits, with bit 0 as the LSB (least significant bit) and bit 31 as the MSB (most significant bit). A word is specified by address “A” and consists of byte data of four addresses: “A”, “A + 1”, “A + 2”, and “A + 3”.



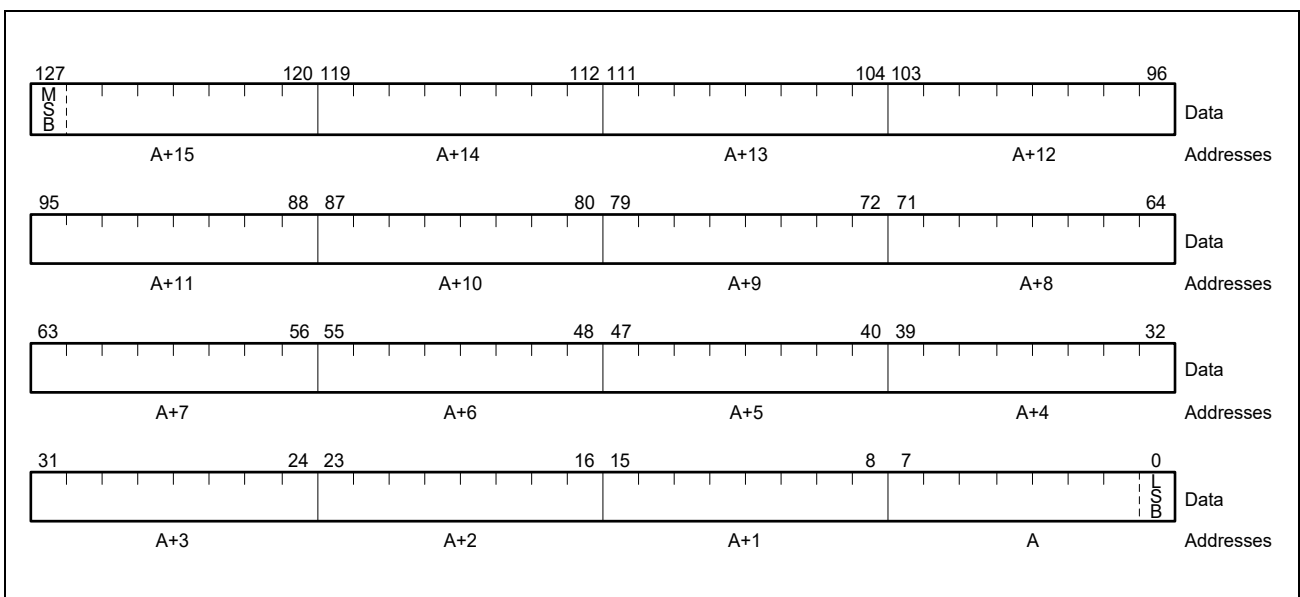
(d) Double-word

A double-word is eight consecutive bytes (64 bits) that start from any 4-byte boundary. Numbers from 0 to 63 are assigned to these bits, with bit 0 as the LSB and bit 63 as the MSB. A double-word is specified by address “A” and consists of byte data of eight addresses: “A”, “A + 1”, “A + 2”, “A + 3”, “A + 4”, “A + 5”, “A + 6”, and “A + 7”.



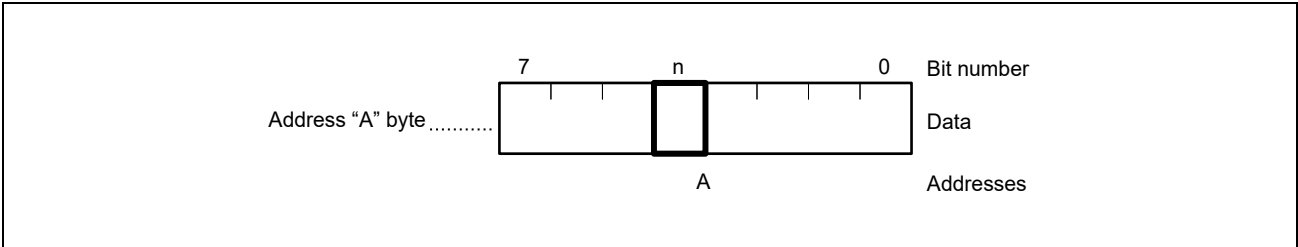
(e) Quad-word

A quad-word is sixteen consecutive bytes (128 bits) that start from any 8-byte boundary. Numbers from 0 to 127 are assigned to these bits, with bit 0 as the LSB and bit 127 as the MSB. A quad-word is specified by address “A” and consists of byte data of sixteen addresses: “A”, “A + 1”, ..., and “A + 15”.



(f) Bit

A bit is bit data at the nth bit within 8-bit data that starts from any byte boundary. Each bit is specified using its byte address “A” and its bit number “n” (n = 0 to 7).



(2) Data Representation

(a) Integers

Integers are represented as binary values using 2's complement, and are used in one of four lengths: 64 bits, 32 bits, 16 bits, or 8 bits. Regardless of the length of an integer, its place uses bit 0 as the LSB, and this place gets higher as the bit number increases. Because this is a 2's complement representation, the MSB is used as a sign bit.

The integer ranges for various data lengths are as follows.

- Double-word (64 bits): -9223372036854775808 to +9223372036854775807
- Word (32 bits): -2147483648 to +2147483647
- Halfword (16 bits): -32768 to +32767
- Byte (8 bits): -128 to +127

Although data format of quad-word (128 bits) is defined, data representation of quad-word is not used in this CPU. This is because quad-word data consists of four-word data or two-double-word data, but this CPU does not process quad-word data directly.

(b) Unsigned Integers

In contrast to “integers” which are data that can take either a positive or negative sign, “unsigned integers” are never negative integers. Like integers, unsigned integers are represented as binary values, and are used in one of four lengths: 64 bits, 32 bits, 16 bits, or 8 bits. Also like integers, the positioning of unsigned integers uses bit 0 as the LSB and gets higher as the bit number increases. However, unsigned integers do not use a sign bit.

The unsigned integer ranges for various data lengths are as follows.

- Double-word (64 bits): 0 to 18446744073709551615
- Word (32 bits): 0 to 4294967295
- Halfword (16 bits): 0 to 65535
- Byte (8 bits): 0 to 255

(c) Bits

Bit data are handled as single-bit data with either of two values: cleared (0) or set (1). There are four types of bit-related operations (listed below), which target only single-byte data in the memory space.

- Set
- Clear
- Invert
- Test

(3) Data Alignment

In this CPU, misaligned data allocation is inhibited. When the result of address calculation is a misaligned address, a misalignment exception (MAE) occurs.

Misaligned access indicates the accesses to the data size with the addresses listed below:

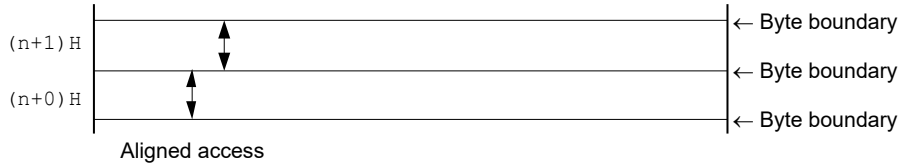
- Halfword size: The access to an address that is not at the halfword boundary (where LSB of the address = 0)
- Word size: The access to an address that is not at the word boundary (where the lowest two bits of the address = 0).
- Double-word size: The access to an address that is not at the double-word boundary (where the lowest three bits of the address = 0).
- Quad-word size: The access to an address that is not at the quad-word boundary (where the lowest four bits of the address = 0)

For the double-word format only, a misaligned access exception does not occur when data is placed at the word boundary but not at the double-word boundary, and data can be normally accessed in double word.

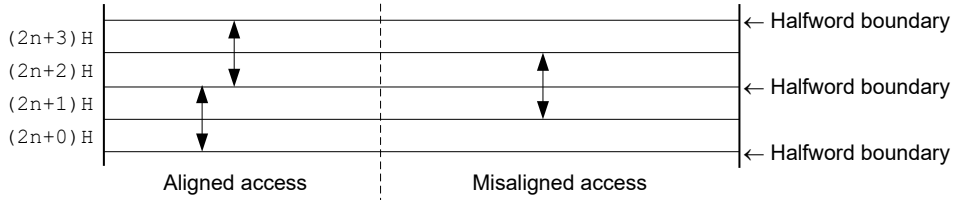
CAUTIONS

1. The following instructions might possibly cause misaligned access. For details, see the relevant descriptions in *the RH850G4MH User's Manual: Software*.
 - LD.H, LD.HU, LD.W, LD.DW
 - SLD.H, SLD.HU, SLD.W
 - ST.H, ST.W, ST.DW
 - SST.H, SST.W
 - LDL.HU, LDL.W, STC.H, STC.W, CAXI
 - LDV.W, LDV.DW, LDV.QW, STV.W, STV.DW, STV.QW
 - LDVZ.H4, STVZ.H4
2. The following instructions do not cause misaligned access, because the address is rounded as the instruction specification when a misaligned address is specified.
 - PREPARE, DISPOSE
 - PUSHSP, POPSP

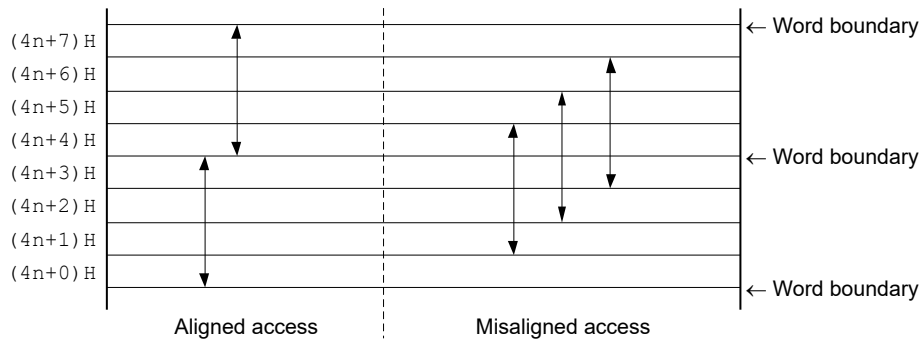
(a) Byte Access



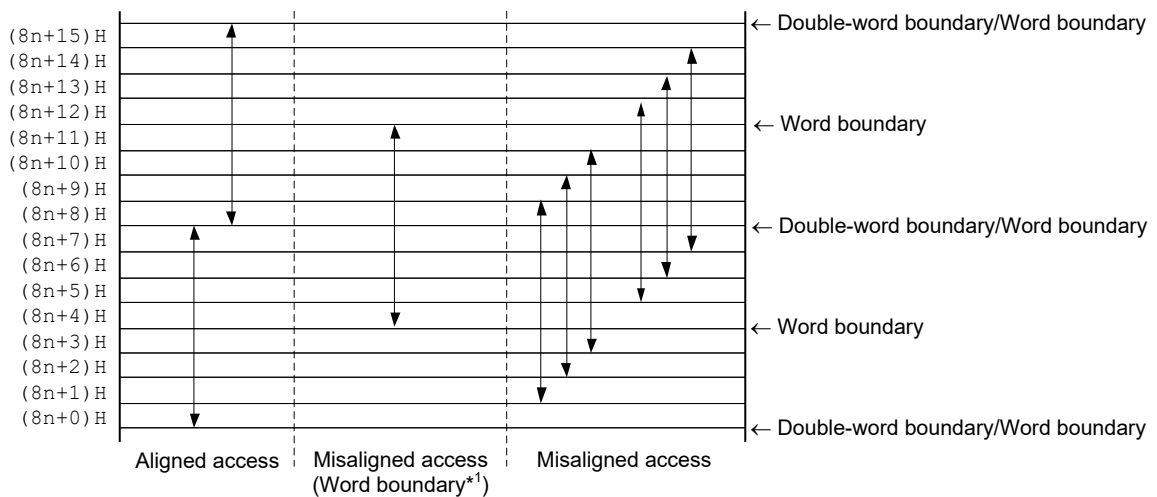
(b) Halfword Access



(c) Word Access



(d) Double-word Access



Note 1. No misalignment exception (MAE) occurs for LD.DW and ST.DW instructions. For details, see LD.DW and ST.DW in the *RH850G4MH User's Manual: Software*. An MAE occurs for other double-word access instructions.

Figure 3.4 Example of Data Placement for Misaligned Access (1/2)

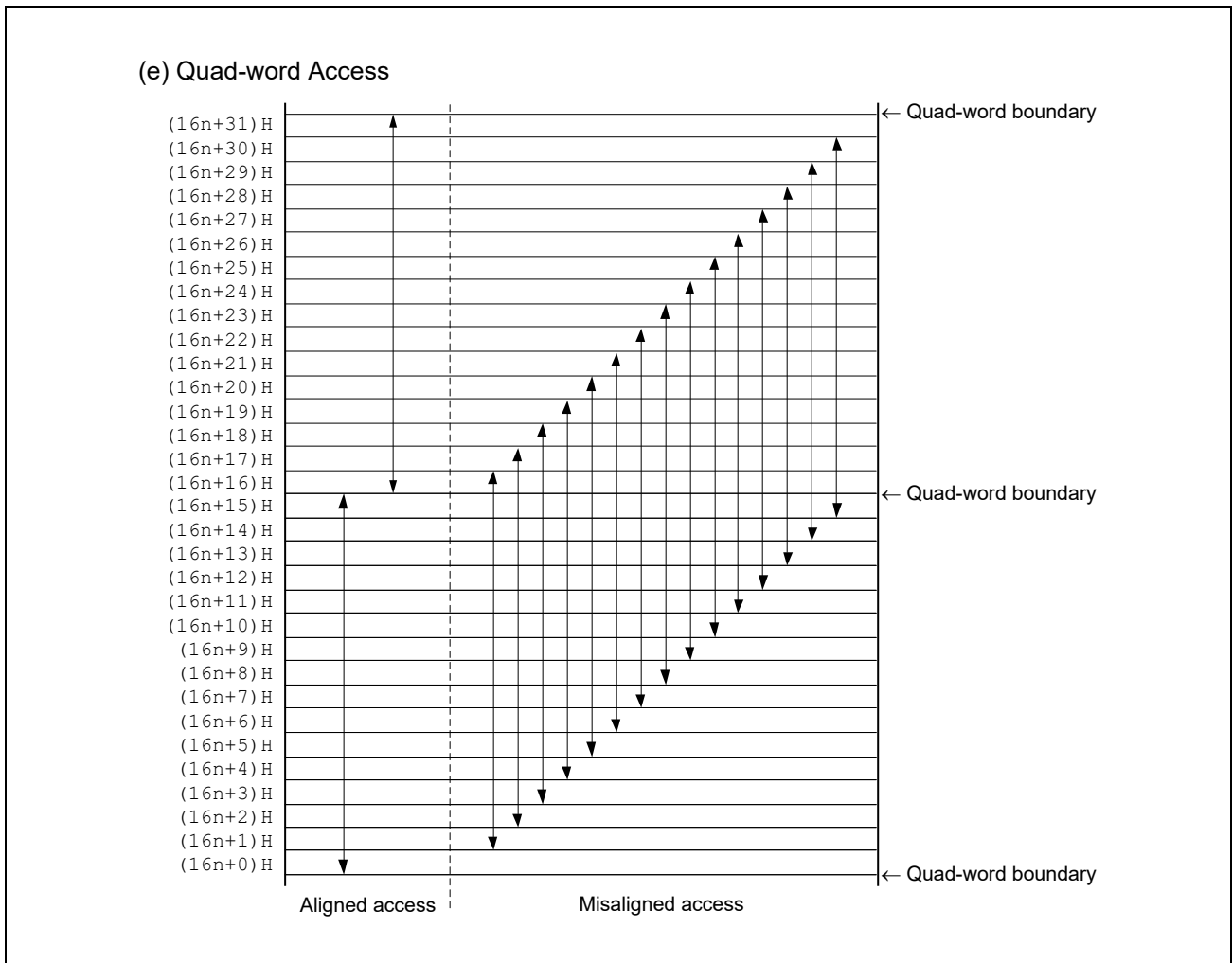


Figure 3.4 Example of Data Placement for Misaligned Access (2/2)

3.2.2.7 Address Space

This CPU supports a linear address space of up to 4 Gbytes. Both memory and I/O can be mapped to this address space (using the memory mapped I/O method). The CPU outputs a 32-bit address for memory and I/O, in which the highest address number is “ $2^{32} - 1$ ”.

The byte data placed at various addresses is defined with bit 0 as the LSB and bit 7 as the MSB. When the data is comprised of multiple bytes, it is defined so that the byte data at the lowest address is the LSB and the byte data at the highest address is the MSB (i.e., in little endian format).

This manual stipulates that, when representing data comprised of multiple bytes, the right edge must be represented as the lower-order addresses and the left side as the higher-order addresses, as shown below.

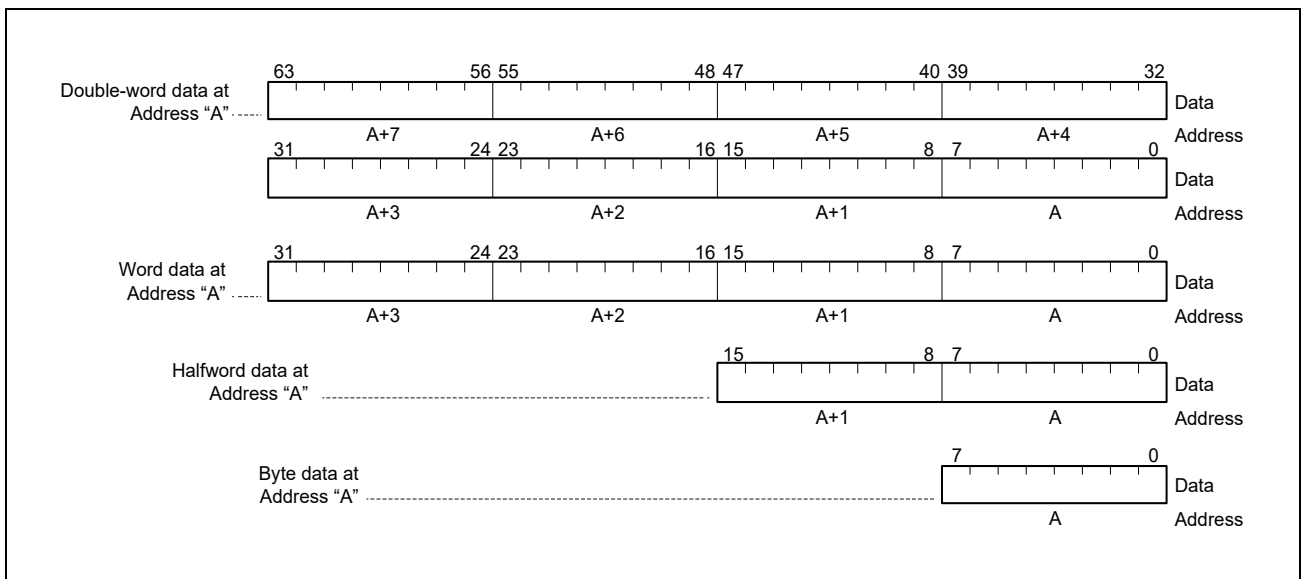


Figure 3.5 Address Space Byte Format

(1) Memory Map

This CPU is 32-bit architecture and supports a linear address space of up to 4 Gbytes. The whole range of this 4-Gbyte address space can be addressed by instruction addressing (instruction fetch access) and data addressing (data access).

A memory map is shown in **Figure 3.6**.

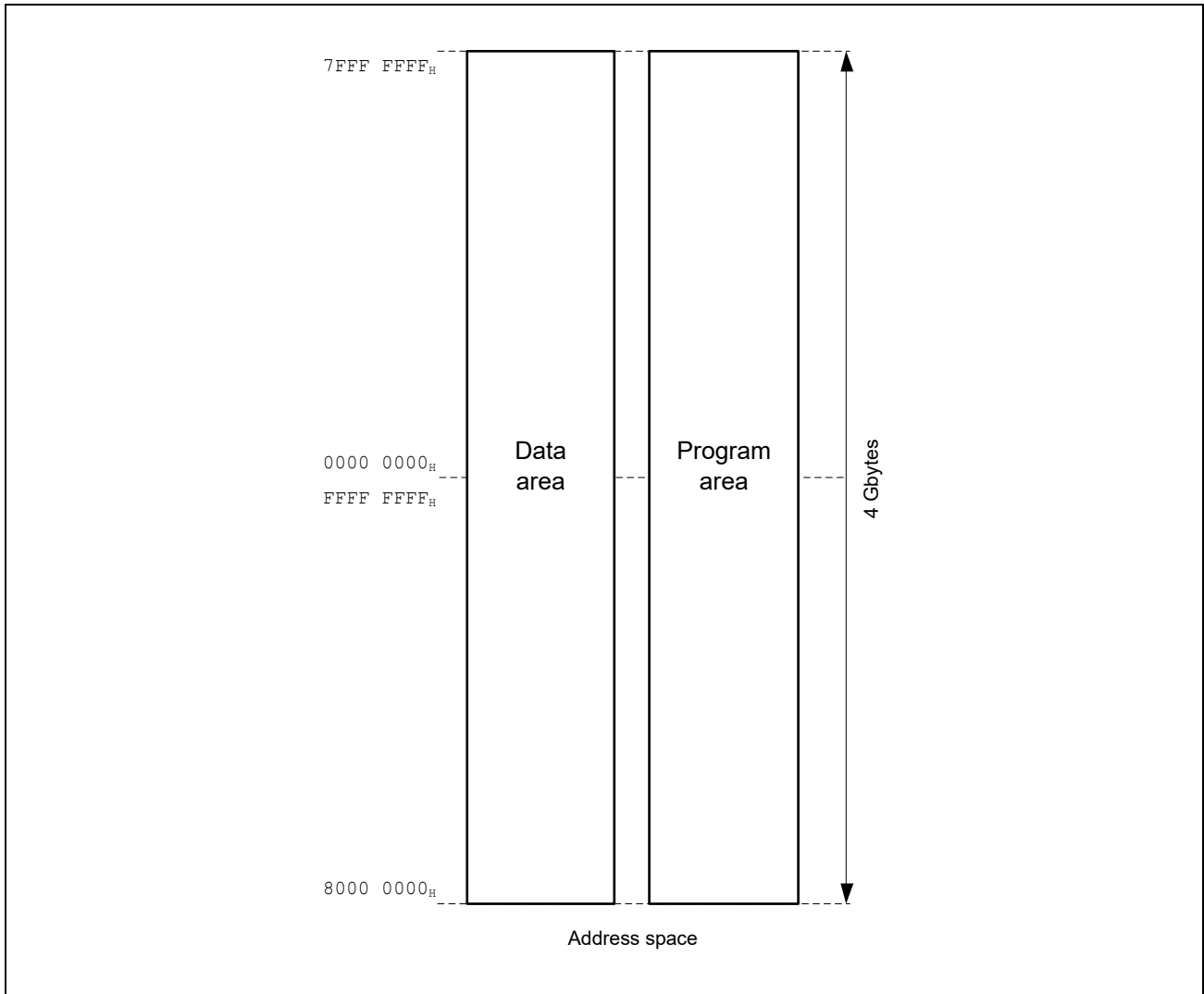


Figure 3.6 Memory Map (Address Space)

(2) Instruction Addressing

The instruction address is determined based on the contents of the program counter (PC), and is automatically incremented according to the number of bytes in the executed instruction. When a branch instruction is executed, the addressing shown below is used to set the branch destination address to the PC.

If the result of address calculation exceeds the positive maximum value $FFFF\ FFFF_H$ by addition, it is wrapped around to $0000\ 0000_H$. If the result of address calculation falls below the positive minimum value $0000\ 0000_H$ by subtraction, it is wrapped around to $FFFF\ FFFF_H$.

(a) Relative Addressing (PC Relative)

Signed N-bit data (displacement: $disp\ N$) in the instruction code is added to the program counter (PC). In this case, displacement is handled as 2's complement data, and the MSB is a sign bit (S). If the displacement is less than 32 bits, the higher-order bits are sign-extended (N differs from one instruction to another).

The JARL, JR, and Bcond instructions are used with this type of addressing.

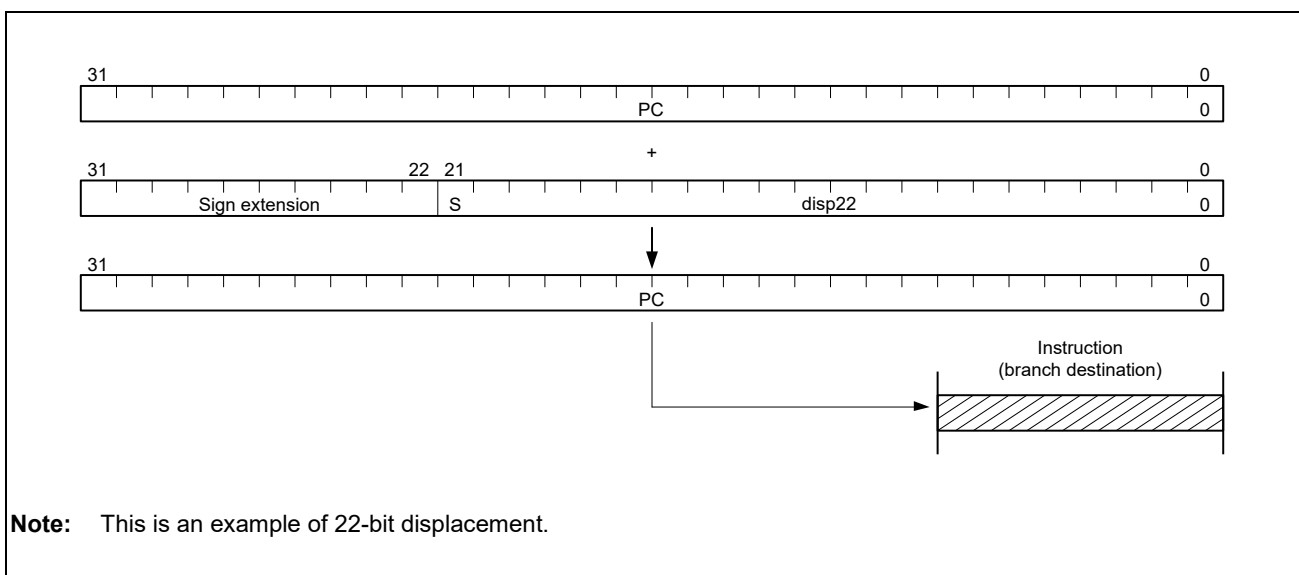


Figure 3.7 Relative Addressing

(b) Register Addressing (Register Indirect)

The contents of the general-purpose register (reg1) or system register (regID) specified by the instruction are transferred to the program counter (PC).

The JMP, CTRET, EIRET, FERET, and DISPOSE instructions are used with this type of addressing.

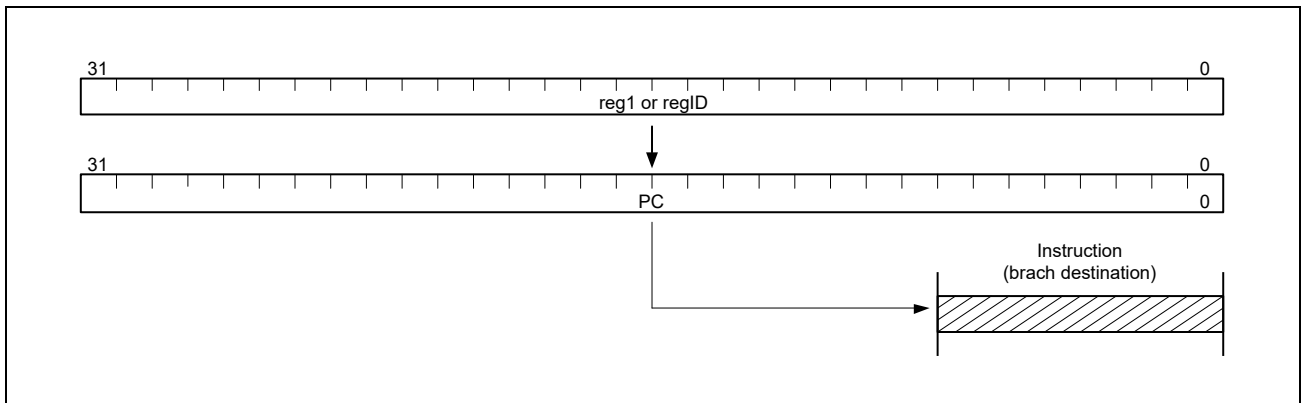


Figure 3.8 Register Addressing

(c) Based Addressing

The contents of the general-purpose register (reg1) and the N-bit displacement (dispN) specified by the instruction are added and transferred to the program counter (PC). At this time, the displacement is handled as a 2's complement data, and the MSB is a sign bit (S). If the displacement is less than 32 bits, the higher bits are sign-extended (N differs from one instruction to another).

The JMP instruction is used with this type of addressing.

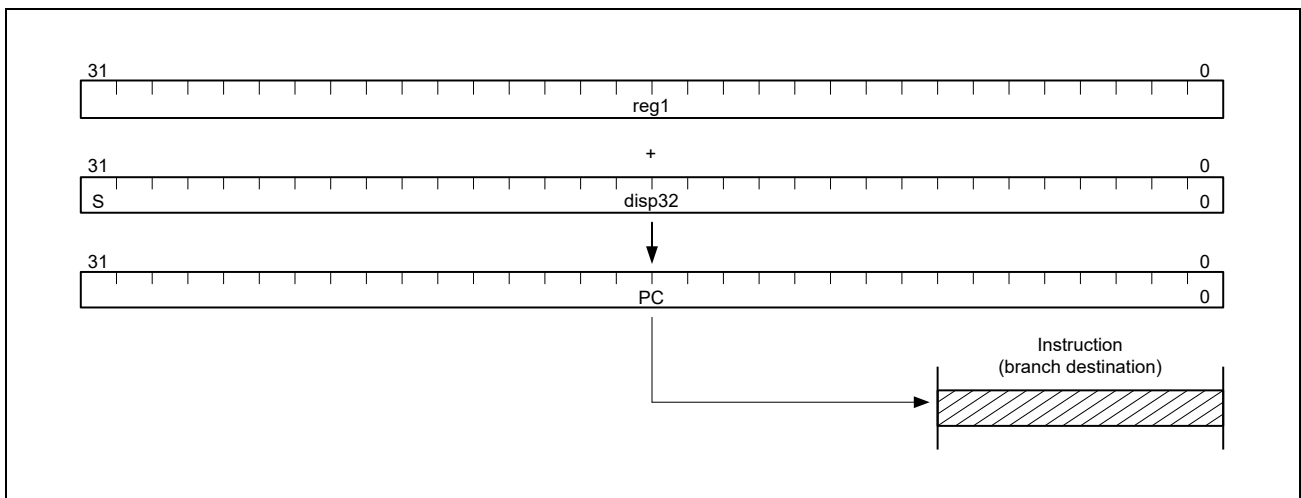


Figure 3.9 Based Addressing

(d) Other Addressing

A value specified by an instruction is transferred to the program counter (PC). How a value is specified is explained in [Operation] or [Description] of each instruction.

The CALLT, SYSCALL, TRAP, FETRAP, and RIE instructions, and branch in case of an exception are used with this type of addressing.

(3) Data Addressing

The following methods can be used to access the target registers or memory when executing an instruction.

If the result of address calculation exceeds the positive maximum value $FFFF\ FFFF_H$ by addition, it is wrapped around to $0000\ 0000_H$. If the result of address calculation falls below the positive minimum value $0000\ 0000_H$ by subtraction, it is wrapped around to $FFFF\ FFFF_H$.

(a) Register Addressing

This addressing method accesses the general-purpose register or system register specified in the general-purpose register field as an operand.

Any instruction that includes the operand `reg1`, `reg2`, `reg3`, or `regID` is used with this type of addressing.

(b) Immediate Addressing

This address mode uses arbitrary size data as the operation target in the instruction code.

Any instruction that includes the operand `imm5`, `imm16`, `vector`, or `cccc` is used with this type of addressing.

NOTE

vector: This is immediate data that specifies the exception vector (00_H to $1F_H$), and is an operand used by the TRAP, FETRAP, and SYSCALL instructions. The data width differs from one instruction to another.

cccc: This is 4-bit data that specifies a condition code, and is an operand used in the Bcond instruction, CMOV instruction, SASF instruction, and SETF instruction.

(c) Based Addressing

There are two types of based addressing, as described below.

3.2.2.7.3.c.1 Type 1

The contents of the general-purpose register (reg1) specified at the addressing specification field in the instruction code are added to the N-bit displacement (dispN) data sign-extended to word length to obtain the operand address, and the target memory is accessed using the operand address. At this time, the displacement is handled as a 2's complement data, and the MSB is a sign bit (S). If the displacement is less than 32 bits, the higher bits are sign-extended (N differs from one instruction to another).

The LD, ST, LDV and STV instructions are used with this type of addressing.

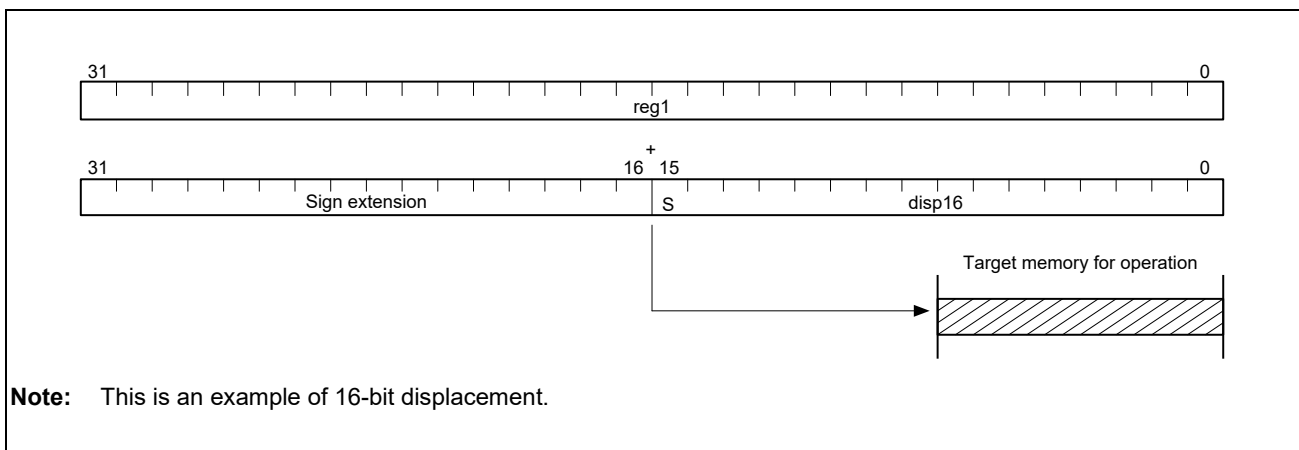


Figure 3.10 Based Addressing (Type 1)

3.2.2.7.3.c.2 Type 2

The contents of the element pointer (r30) are added to the N-bit displacement data (dispN) zero-extended to a word length to obtain the operand address, and target memory is accessed using the operand address. If the displacement is less than 32 bits, the higher bits are zero-extended (N differs from one instruction to another).

The SLD and SST instructions are used with this type of addressing.

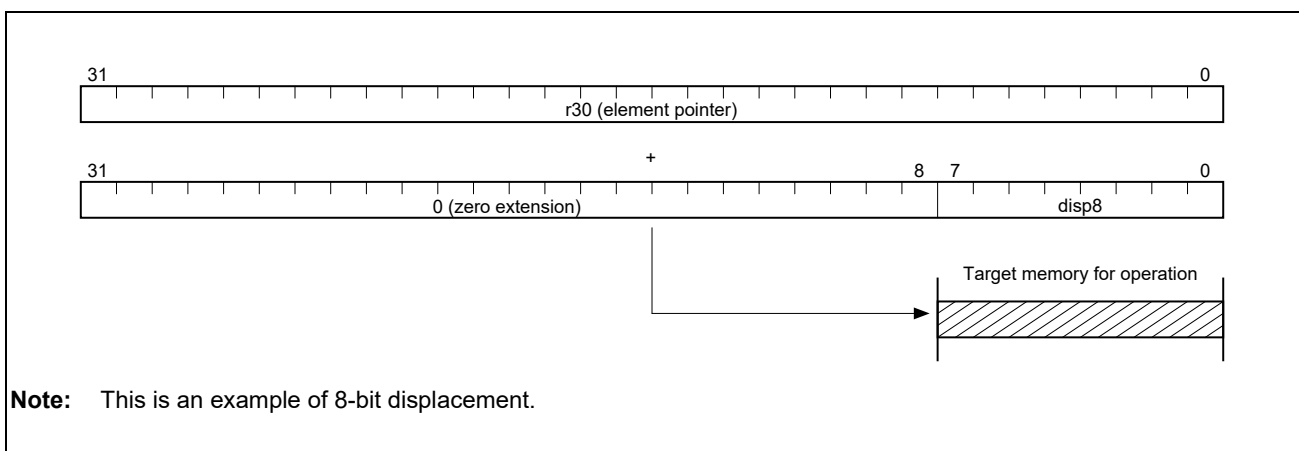


Figure 3.11 Based Addressing (Type 2)

(d) Bit Addressing

The contents of the general-purpose register (reg1) are added to the N-bit displacement (dispN) data sign-extended to word length to obtain the operand address, and one bit (as specified by 3-bit data “bit #3”) in one byte of the target memory is accessed using the operand address. At this time, the displacement is handled as a 2’s complement data, and the MSB is a sign bit (S). If the displacement is less than 32 bits, the higher bits are sign-extended (N differs from one instruction to another).

The CLR1, SET1, NOT1, and TST1 instructions are used with this type of addressing.

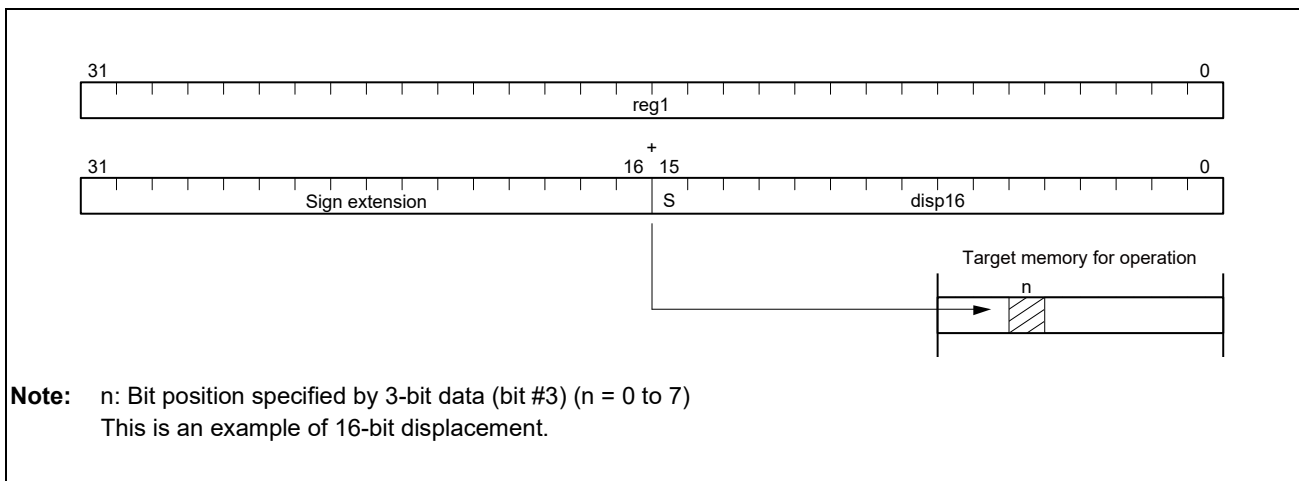


Figure 3.12 Bit Addressing

(e) Post Index Increment/Decrement Addressing

The contents of the general-purpose register (reg1) are used as an operand address to access the target memory, and then the general-purpose register (reg1) is updated. There are two types of the general-purpose register updating, by incrementing or decrementing.

If the result of incrementing the general-purpose register (reg1) value exceeds the positive maximum value $FFFF\ FFFF_H$, the result wraps around to $0000\ 0000_H$, and, if the result of decrementing the general-purpose register value is less than the positive minimum value $0000\ 0000_H$, the result wraps around to $FFFF\ FFFF_H$.

3.2.2.7.3.e.1 Type 1

The general-purpose register (reg1) is updated by adding a constant that depends on the type of accessed data (the size of the accessed data) to the contents of the general-purpose register (reg1). If the type of accessed data is a byte, 1 is added, if the type is a halfword, 2 is added, if the type is a word, 4 is added, and if the type is a double-word, 8 is added.

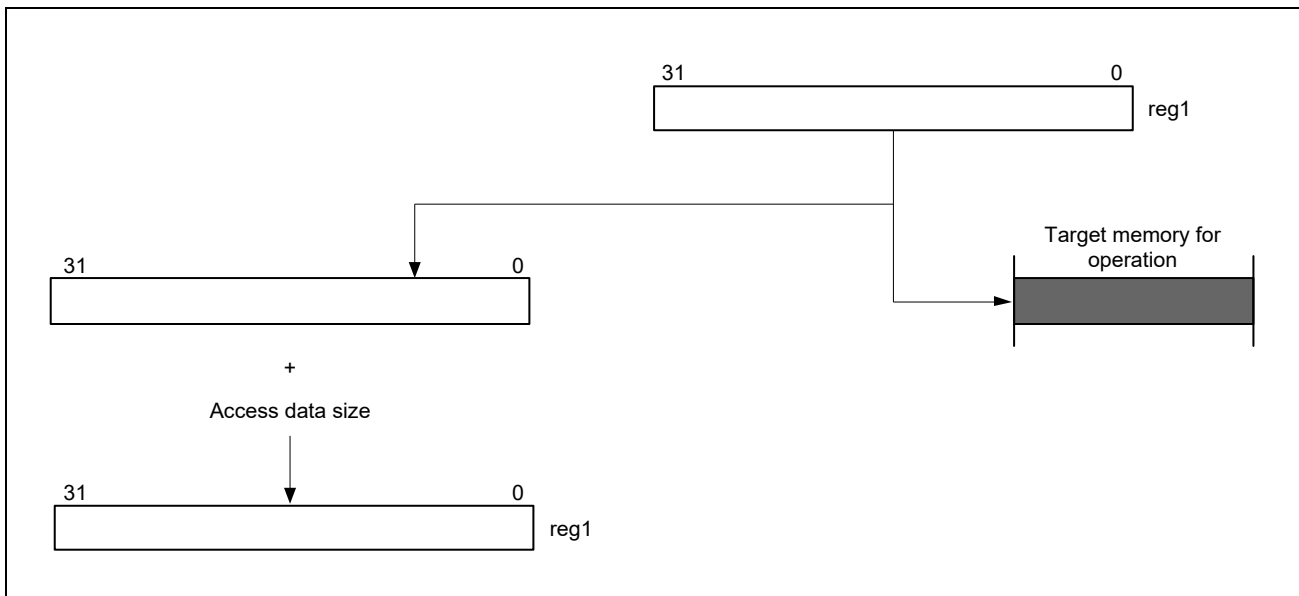


Figure 3.13 Post Index Increment/Decrement Addressing (Type 1)

3.2.2.7.3.e.2 Type 2

The general-purpose register (reg1) is updated by subtracting a constant that depends on the size of the accessed data from the contents of the general-purpose register (reg1). If the size of accessed data is a byte, 1 is subtracted, if the size is a halfword, 2 is subtracted, if the size is a word, 4 is subtracted, and if the size is a double-word, 8 is subtracted.

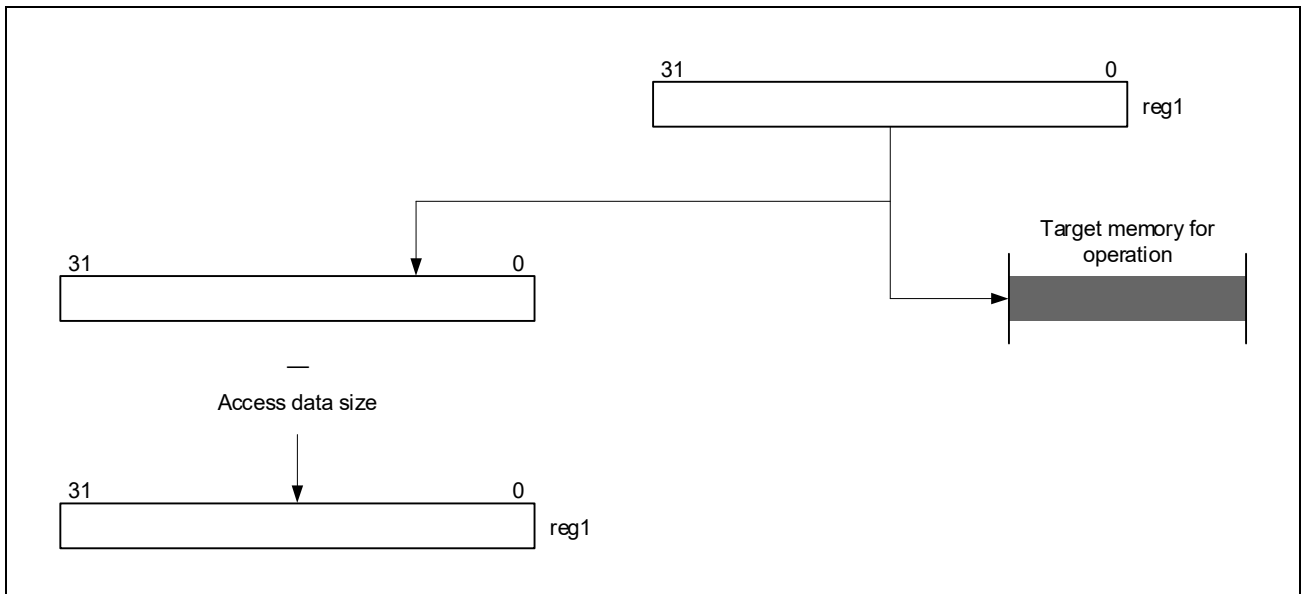


Figure 3.14 Post Index Increment/Decrement Addressing (Type 2)

(f) Other Addressing

The target memory is accessed using a value specified by an instruction as the operand address. How a value is specified is explained in [Operation] or [Description] of each instruction.

The SWITCH, CALLT, SYSCALL, PREPARE, DISPOSE, PUSHSP, and POPSP instructions are used with this type of addressing.

3.2.2.8 Execution Timing of a Store Instruction

Completion of writing by a store instruction may differ depending on the type of the memory to be accessed. For details, see **Section 3.2.7.2, Guaranteeing the Completion of Store Instruction.**

3.2.2.9 Memory Ordering

This CPU guarantees that memories are accessed in the programmed order. However, in the system that incorporates multiple bus masters such as the bus system with DMA or multi-core, the order of accesses to memories needs to be considered. For these cases, see **Section 3.2.7.1, Synchronization Processing.**

3.2.2.10 Acquiring the CPU Number

This CPU provides a method for identifying CPUs in a multi-processor system.

In the multi-processor configuration, you can identify which CPU core is running a program by referencing the PEID register. A unique number within a multi-processor system is assigned to the PEID register according to the specification of the product.

3.2.2.11 System Protection Identifier (SPID)

In this CPU, memory resources and peripheral devices are managed by system protection groups. By specifying the group to which the program being executed belongs, you can assign accessible memory resources and peripheral devices to the program.

The program being executed belongs to the group specified by the SPID, and whether the memory resources and peripheral devices are accessible is decided using the SPID. A value can be set to the SPID register by the supervisor.

CAUTION

According to the value of the SPID, how operations are assigned to memory resources and peripheral devices is determined by the specifications of the product.

3.2.2.12 Timestamp Counter

This CPU has a 64-bit timestamp counter. It can measure long time, and so can be used as specific information for time identification.

Since the timestamp counter is allocated to system registers, it can be accessed quickly by using a LDSR/STSR instruction.

If an overflow occurs during counting operation, no exception will occur.

(1) How to Operate the Timestamp Counter

The value of the timestamp counter is initialized to 0 by a reset. Therefore, if you want to retain the value of the counter across a reset, then before the reset, save the value of the counter in a memory that is not initialized by the reset, restore the value to the counter after the reset, and restart counting.

The counter is 64-bit width, so the counter consists of two 32-bit system registers, TSCOUNTL and TSCOUNTH. Both registers need to be accessed by using LDSR/STSR instructions.

When the counter is not running (the value of the TSCTRL.CEN bit is 0), no special care is needed to access the two registers.

However, if the counter is running (the value of the TSCTRL.CEN bit is 1), it is not recommended to update the counter by using LDSR instructions. In this case, the timing of update of the counter is not guaranteed. It is recommended to update the counter when it is not running.

Also, when reading the value of the counter by using STSR instructions while it is running, it is recommended to follow the procedure below.

TSCNTRD:			
STSR	1, r21, 11	--	Read the upper side of the counter
STSR	0, r20, 11	--	Read the lower side of the counter
STSR	1, r22, 11	--	Read again the upper side of the counter
CMP	r21, r22	--	Compare the two values read from the upper side of the counter
BNE	TSCNTRD	--	If they are not identical, a carry has occurred. Read again.

Even if the CPU core is not in operation after the execution of a HALT or SNOOZE instruction, the timestamp counter continues counting.

3.2.2.13 Performance Measurement Function

This CPU has the performance measurement function. The performance measurement function can measure the performance of programs executed, the effects of interrupts generated during operation, etc. by counting the occurrence of the event specified by the PMCTRLn.CND bit.

The system registers used by the performance measurement function can be accessed only in supervisor mode after a reset. However, it can be accessed in user mode by changing the setting of the PMUMCTRL register.

The performance measurement function itself works in user mode regardless of the setting of the PMUMCTRL register. Even if all performance measurement channels are made inaccessible in user mode by using the PMUMCTRL register, configuration in supervisor mode allows performance measurement during operation in user mode.

This CPU has four channels of system register set for the performance measurement function.

3.2.3 Register Set

This chapter describes the program register and system register mounted on this CPU.

3.2.3.1 Program Registers

Program registers includes general-purpose registers (r0 to r31) and the program counter (PC). r0 always retains 0. The values of the general-purpose registers r1 to r31 after a reset are undefined. The value of the PC after a reset is the value of RBASE register.

Table 3.6 Program Registers

Program Register	Name	Function	Description
General-purpose registers	r0	Zero register	Always retains 0
	r1	Assembler reserved register	Used as working register for generating addresses
	r2	Register for address and data variables (used when the real-time OS used does not use this register)	
	r3	Stack pointer (SP)	Used for generating a stack frame when a function is called
	r4	Global pointer (GP)	Used for accessing a global variable in the data area
	r5	Text pointer (TP)	Used as a register that indicates the start of the text area (area where program code is placed)
	r6 to r29	Register for addresses and data variables	
	r30	Element pointer (EP)	Used as a base pointer for generating addresses when accessing memory
	r31	Link pointer (LP)	Used when the compiler calls a function
Program counter	PC	Retains instruction addresses during execution of programs	

Note: For further descriptions of r1, r3 to r5, and r31 used for an assembler and/or C compiler, see the manual of each software development environment.

(1) General-purpose Registers

A total of 32 general-purpose registers (r0 to r31) are provided. All of these registers can be used for either data variables or address variables.

Of the general-purpose registers, r0 to r5, r30, and r31 are assumed to be used for special purposes in software development environments, so it is necessary to note the following when using them.

(a) r0, r3, and r30

These registers are implicitly used by instructions.

r0 is a register that always retains 0. It is used for operations that use 0, addressing with base address being 0, etc.

r3 is implicitly used by the PREPARE, DISPOSE, PUSHSP, and POPSP instructions.

r30 is used as a base pointer when the SLD instruction or SST instruction accesses memory.

(b) r1, r4, r5, and r31

These registers are implicitly used by the assembler and C compiler.

When using these registers, register contents must first be saved so they are not lost and can be restored after the registers are used.

(c) r2

This register is used by a real-time OS in some cases. If the real-time OS that is being used is not using r2, r2 can be used as a register for address variables or data variables.

(2) PC — Program Counter

The PC retains the address of the instruction being executed.

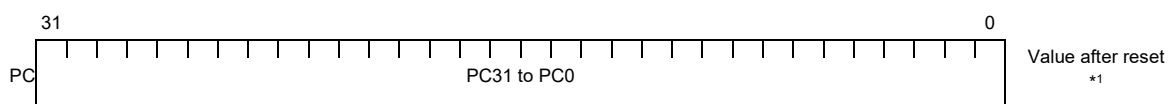


Table 3.7 PC Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 1	PC31 to PC1	These bits indicate the address of the instruction being executed.	R/W	*1
0	PC0	This bit is fixed to 0. Branching to an odd number address is disabled.	R/W	0

Note 1. For details, see **Section 4, Address MAP**.

3.2.3.2 Basic System Registers

The basic system registers are used to control CPU status and to retain exception information.

The basic system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.8 Basic System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR0, 0	EIPC	Status save registers when acknowledging EI level exception	SV
SR1, 0	EIPSW	Status save registers when acknowledging EI level exception	SV
SR2, 0	FEPC	Status save registers when acknowledging FE level exception	SV
SR3, 0	FEPSW	Status save registers when acknowledging FE level exception	SV
SR5, 0	PSW	Program status word	*1
SR6, 0	FPSR	(See Section 3.2.3.4, FPU Function Registers)	CU0 and SV
SR7, 0	FPEPC	(See Section 3.2.3.4, FPU Function Registers)	CU0 and SV
SR8, 0	FPST	(See Section 3.2.3.4, FPU Function Registers)	CU0
SR9, 0	FPCC	(See Section 3.2.3.4, FPU Function Registers)	CU0
SR10, 0	FPCFG	(See Section 3.2.3.4, FPU Function Registers)	CU0
SR13, 0	EIIC	EI level exception cause	SV
SR14, 0	FEIC	FE level exception cause	SV
SR16, 0	CTPC	CALLT execution status save register	UM
SR17, 0	CTPSW	CALLT execution status save register	UM
SR20, 0	CTBP	CALLT base pointer	UM
SR21, 0	SNZCFG	SNOOZE control register	SV
SR28, 0	EIWR	EI level exception working register	SV
SR29, 0	FEWR	FE level exception working register	SV
SR0, 1	SPID	System protection identifier	SV
SR1, 1	SPIDLIST	List of system protection identifiers that can be specified in SPID	SV
SR2, 1	RBASE	Reset vector base address	SV
SR3, 1	EBASE	Exception handler vector address	SV
SR4, 1	INTBP	Base address of the interrupt handler "address" table	SV
SR5, 1	MCTL	CPU control	SV
SR6, 1	PID	Processor ID	SV
SR8, 1	SVLOCK	Supervisor lock	SV
SR11, 1	SCCFG	SYSCALL operation setting	SV
SR12, 1	SCBP	SYSCALL base pointer	SV
SR0, 2	PEID	Processor element identifier	SV
SR1, 2	BMID	Bus master identifier	SV
SR6, 2	MEA	Memory error address	SV
SR8, 2	MEI	Memory error information	SV
SR15, 2	RBCR0	Register bank control 0	SV
SR16, 2	RBCR1	Register bank control 1	SV
SR17, 2	RBNR	Register bank number	SV
SR18, 2	RBIP	Register bank initial pointer	SV

Note 1. The access permission differs depending on the bit. For details, see **Section 3.2.3.2(5), PSW — Program Status Word**.

(1) EIPC — Status Save Register when Acknowledging EI Level Exception

When an EI level exception is acknowledged, the address of the instruction that was being executed when the EI level exception occurred, or of the next instruction, is saved to the EIPC register (see **Section 3.2.4.1(3), Types of Exceptions**).

Because there is only one pair of EI level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

Be sure to set an even-numbered address to the EIPC register. An odd-numbered address must not be specified.

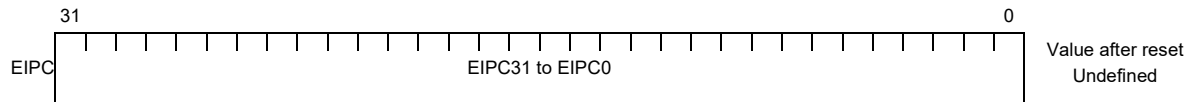


Table 3.9 EIPC Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 1	EIPC31 to EIPC1	These bits indicate the PC saved when an EI level exception is acknowledged.	R/W	Undefined
0	EIPC0	This bit indicates the PC restored when an EI level exception is acknowledged. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the EIRET instruction is executed is 0.	R/W	Undefined

(2) EIPSW — Status Save Register when Acknowledging EI Level Exception

When an EI level exception is acknowledged, the current PSW setting is saved to the EIPSW register.

Because there is only one pair of EI level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

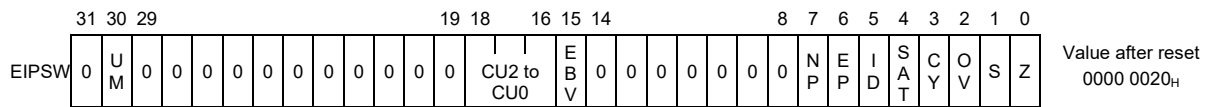


Table 3.10 EIPSW Register Contents

Bit	Name	Description	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
30	UM	This bit stores the PSW.UM bit setting when an EI level exception is acknowledged.	R/W	0
29 to 19	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
18 to 16	CU2 to CU0	These bits store the PSW.CU2-0 field setting when an EI level exception is acknowledged.*1	R/W	0
15	EBV	This bit stores the PSW.EBV bit setting when an EI level exception is acknowledged.	R/W	0
14 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	NP	This bit stores the PSW.NP bit setting when an EI level exception is acknowledged.	R/W	0
6	EP	This bit stores the PSW.EP bit setting when an EI level exception is acknowledged.	R/W	0
5	ID	This bit stores the PSW.ID bit setting when an EI level exception is acknowledged.	R/W	1
4	SAT	This bit stores the PSW.SAT bit setting when an EI level exception is acknowledged.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when an EI level exception is acknowledged.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when an EI level exception is acknowledged.	R/W	0
1	S	This bit stores the PSW.S bit setting when an EI level exception is acknowledged.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when an EI level exception is acknowledged.	R/W	0

Note 1. CU2 is reserved for future CPUs that are to be made compatible with this CPU. It is always set to 0 in this CPU.

(3) FEPC — Status Save Register when Acknowledging FE Level Exception

When an FE level exception is acknowledged, the address of the instruction that was being executed when the FE level exception occurred, or of the next instruction, is saved to the FEPC register (see **Section 3.2.4.1(3), Types of Exceptions**). Because there is only one pair of FE level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

Be sure to set an even-numbered address to the FEPC register. An odd-numbered address must not be specified.



Table 3.11 FEPC Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 1	FEPC31 to FEPC1	These bits indicate the PC saved when an FE level exception is acknowledged.	R/W	Undefined
0	FEPC0	This bit indicates the PC saved when an FE level exception is acknowledged. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the FERET instruction is executed is 0.	R/W	Undefined

(4) FEPSW — Status Save Register when Acknowledging FE Level Exception

When an FE level exception is acknowledged, the current PSW setting is saved to the FEPSW register.

Because there is only one pair of FE level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

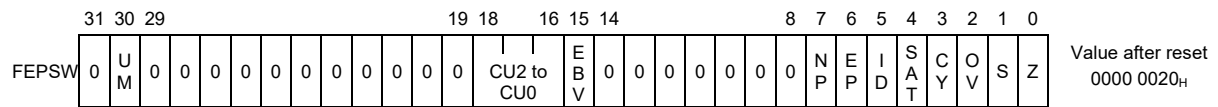


Table 3.12 FEPSW Register Contents

Bit	Name	Description	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
30	UM	This bit stores the PSW.UM bit setting when an FE level exception is acknowledged.	R/W	0
29 to 19	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
18 to 16	CU2 to CU0	These bits store the PSW.CU2-0 field setting when an FE level exception is acknowledged*1.	R/W	0
15	EBV	This bit stores the PSW.EBV bit setting when an FE level exception is acknowledged.	R/W	0
14 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	NP	This bit stores the PSW.NP bit setting when an FE level exception is acknowledged.	R/W	0
6	EP	This bit stores the PSW.EP bit setting when an FE level exception is acknowledged.	R/W	0
5	ID	This bit stores the PSW.ID bit setting when an FE level exception is acknowledged.	R/W	1
4	SAT	This bit stores the PSW.SAT bit setting when an FE level exception is acknowledged.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when an FE level exception is acknowledged.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when an FE level exception is acknowledged.	R/W	0
1	S	This bit stores the PSW.S bit setting when an FE level exception is acknowledged.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when an FE level exception is acknowledged.	R/W	0

Note 1. CU2 is reserved for future CPUs that are to be made compatible with this CPU. It is always set to 0 in this CPU.

(5) PSW — Program Status Word

PSW (program status word) is a set of flags that indicate the program status (instruction execution result) and bits that indicate the operation status of the CPU (flags are bits in the PSW that are referenced by a condition instruction (Bcond, CMOV, etc.)).

CAUTIONS

1. When the LDSR instruction is used to change the contents of this register, the changed contents become valid from the subsequent instruction. For details, see **Section 3.2.7.3, Hazard Management after System Register Update**.
2. The access permission for the PSW register differs depending on the bit. All bits can be read, but some bits can only be written under certain conditions. See **Table 3.13** for the access permission for each bit.

Table 3.13 Access Permission for PSW Register

Bit		Access Permission when Reading	Access Permission when Writing
30	UM	UM	SV*1
18 to 16	CU2 to CU0		SV*1
15	EBV		SV*1
7	NP		SV*1
6	EP		SV*1
5	ID		SV*1
4	SAT		UM
3	CY		UM
2	OV		UM
1	S		UM
0	Z		UM

Note 1. The access permission for the whole PSW register is UM, so the PIE exception does not occur even if the register is written by using an LDSR instruction when PSW.UM is 1. In this case, writing is ignored.

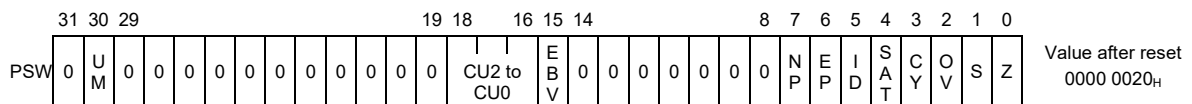


Table 3.14 PSW Register Contents (1/2)

Bit	Name	Description	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
30	UM	This bit indicates that the CPU is in user mode (in UM mode). 0: Supervisor mode 1: User mode	R/W	0
29 to 19	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
18 to 16	CU2 to CU0	These bits indicate the coprocessor use permissions. When the bit corresponding to the coprocessor is 0, a coprocessor unusable exception occurs if an instruction for the coprocessor is executed or a coprocessor resource (system register) is accessed. Bit 18 (CU2): Fixed to 0*1. Bit 17 (CU1): FXU Bit 16 (CU0): FPU CU2 to CU0 are fixed to 0 in the devices that do not have corresponding coprocessors.	R/W	000

Table 3.14 PSW Register Contents (2/2)

Bit	Name	Description	R/W	Value after Reset
15	EBV	This bit indicates the reset vector and exception vector operation. See Section 3.2.3.2(16), RBASE — Reset Vector Base Address and Section 3.2.3.2(17), EBASE — Exception Handler Vector Address .	R/W	0
14 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	NP	This bit disables the acknowledgement of FE level exception. When an FE level exception is acknowledged, this bit is set to 1 to disable the acknowledgement of EI level and FE level exceptions. As for the exceptions which the NP bit disables the acknowledgment, see Table 3.95, Exception Cause List . 0: The acknowledgement of FE level exception is enabled. 1: The acknowledgement of FE level exception is disabled.	R/W	0
6	EP	This bit indicates that an exception other than an interrupt is being serviced. It is set to 1 when the corresponding exception occurs. This bit does not affect acknowledging an exception request even when it is set to 1. 0: An exception other than an interrupt is not being serviced. 1: An exception other than an interrupt is being serviced.	R/W	0
5	ID	This bit disables the acknowledgement of EI level exception. When an EI level or FE level exception is acknowledged, this bit is set to 1 to disable the acknowledgement of EI level exception. As for the exceptions which the ID bit disables the acknowledgment, see Table 3.95, Exception Cause List . This bit is also used to disable EI level exceptions from being acknowledged as a critical section while an ordinary program or interrupt is being serviced. It is set to 1 when the DI instruction is executed, and cleared to 0 when the EI instruction is executed. The change of the ID bit by the EI or ID instruction will be enabled from the next instruction. 0: The acknowledgement of EI level exception is enabled. 1: The acknowledgement of EI level exception is disabled.	R/W	1
4	SAT* ²	This bit indicates that a saturation arithmetic operation instruction resulted in overflow and saturation processing is applied to the result. This is a cumulative flag, that is, it is set (1) once a saturation occurs and not cleared (0) by subsequent instructions with unsaturated results. This bit is cleared by the LDSR instruction. Note that execution of an arithmetic operation instruction neither set nor clear this flag. 0: The result was not saturated 1: The result was saturated	R/W	0
3	CY	This bit indicates whether a carry or borrow has occurred in the operation result. 0: Carry and borrow have not occurred. 1: Carry or borrow has occurred.	R/W	0
2	OV* ²	This bit indicates whether or not an overflow has occurred during an operation. 0: Overflow has not occurred. 1: Overflow has occurred.	R/W	0
1	S* ²	This bit indicates whether or not the result of an operation is negative. 0: Result of operation is positive or 0. 1: Result of operation is negative.	R/W	0
0	Z	This bit indicates whether or not the result of an operation is 0. 0: Result of operation is not 0. 1: Result of operation is 0.	R/W	0

Note 1. The coprocessor use permission CU2 is reserved for future CPUs that are to be made compatible with this CPU.

Note 2. Saturation processing is applied to the operation result in accordance with the contents of the OV and S flags. The SAT flag is set (1) only when the OV flag is set (1) in the saturation arithmetic operation.

Operation Result Status	Flag Status			Operation Result after Saturation Processing
	SAT	OV	S	
Exceeded positive maximum value	1	1	0	7FFF FFFF _H
Exceeded negative maximum value	1	1	1	8000 0000 _H
Positive (maximum value not exceeded)	Value prior to operation is retained	0	0	Operation result as is
Negative (maximum value not exceeded)			1	

(6) EIIC — EI Level Exception Cause

The EIIC register retains the cause of any EI level exception that occurs. The value retained in this register is an exception code corresponding to a specific exception cause (see **Table 3.95, Exception Cause List**).

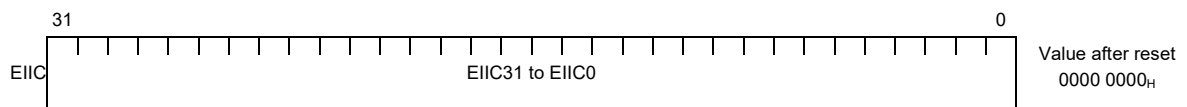


Table 3.15 EIIC Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 0	EIIC31 to EIIC0	These bits store the exception cause code when an EI level exception occurs. The EIIC15-0 field stores the exception cause codes shown in Table 3.95 . The EIIC31-16 field stores detailed exception cause codes defined individually for each exception. If there is no particular definition, these bits are set to 0.	R/W	0

(7) FEIC — FE Level Exception Cause

The FEIC register retains the cause of any FE level exception that occurs. The value retained in this register is an exception code corresponding to a specific exception cause (see **Table 3.95, Exception Cause List**).



Table 3.16 FEIC Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 0	FEIC31 to FEIC0	These bits store the exception cause code when an FE level exception occurs. The FEIC15-0 field stores the exception cause codes shown in Table 3.95 . The FEIC31-16 field stores detailed exception cause codes defined individually for each exception. If there is no particular definition, these bits are set to 0.	R/W	0

(8) CTPC — Status Save Register when Executing CALLT

When a CALLT instruction is executed, the address of the next instruction after the CALLT instruction is saved to CTPC.

Be sure to set an even-numbered address to the CTPC register. An odd-numbered address must not be specified.

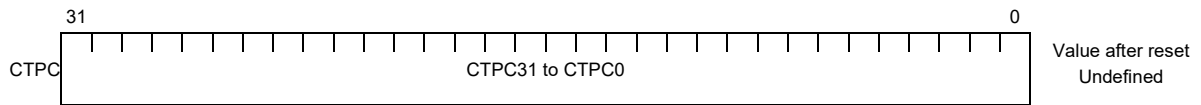


Table 3.17 CTPC Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 1	CTPC31 to CTPC1	These bits indicate the PC of the instruction after the CALLT instruction.	R/W	Undefined
0	CTPC0	This bit indicates the PC of the instruction after the CALLT instruction. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the CTRET instruction is executed is 0.	R/W	Undefined

(9) CTPSW — Status Save Register when Executing CALLT

When a CALLT instruction is executed, some of the PSW (program status word) settings are saved to CTPSW.

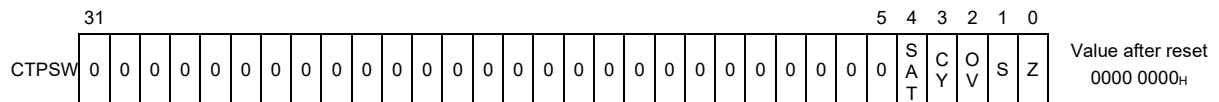


Table 3.18 CTPSW Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4	SAT	This bit stores the PSW.SAT bit setting when the CALLT instruction is executed.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when the CALLT instruction is executed.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when the CALLT instruction is executed.	R/W	0
1	S	This bit stores the PSW.S bit setting when the CALLT instruction is executed.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when the CALLT instruction is executed.	R/W	0

(10) CTBP — CALLT Base Pointer

The CTBP register is used to specify table addresses of the CALLT instruction and generate target addresses.

Be sure to set the CTBP register to a halfword address.

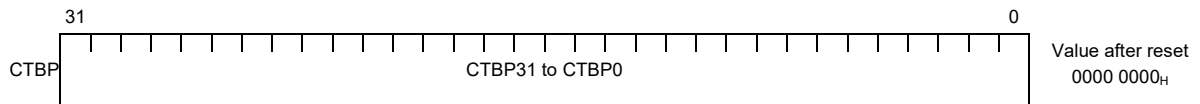


Table 3.19 CTBP Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 1	CTBP31 to CTBP1	These bits indicate the base pointer address of the CALLT instruction. These bits indicate the start address of the table used by the CALLT instruction.	R/W	0
0	CTBP0	This bit indicates the base pointer address of the CALLT instruction. This bit indicates the start address of the table used by the CALLT instruction. Always set this bit to 0.	R	0

(11) SNZCFG — SNOOZE Configuration

The SNZCFG register is used to configure the operation of the SNOOZE instruction.

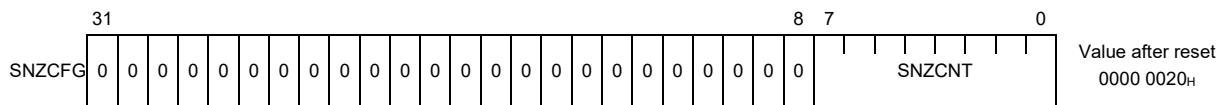


Table 3.20 SNZCFG Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7 to 0	SNZCNT	These bits specify the number of clock cycles in which execution is to be temporarily halted by the SNOOZE instruction.	R/W	20 _H

(12) EIWR — EI Level Exception Working Register

The EIWR register is used as a working register when an EI level exception has occurred.

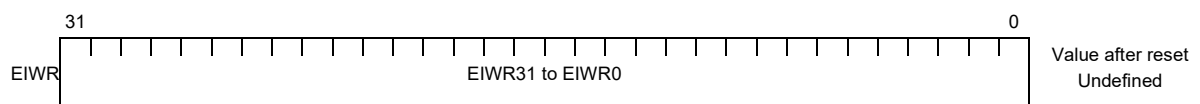


Table 3.21 EIWR Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 0	EIWR31 to EIWR0	These bits constitute a working register that can be used for any purpose during the processing of an EI level exception. Use this register for purposes such as storing the values of general-purpose registers.	R/W	Undefined

(13) FEWR — FE Level Exception Working Register

The FEWR register is used as a working register when an FE level exception has occurred.

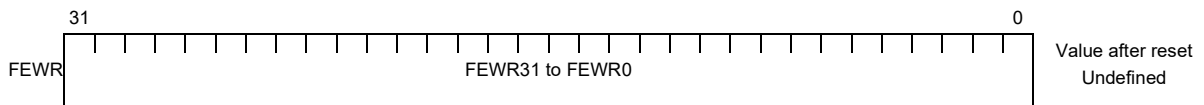


Table 3.22 FEWR Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 0	FEWR31 to FEWR0	These bits constitute a working register that can be used for any purpose during the processing of an FE level exception. Use this register for purposes such as storing the values of general-purpose registers.	R/W	Undefined

(14) SPID — System Protection Identifier

The SPID register holds the system protection identifier of the CPU.

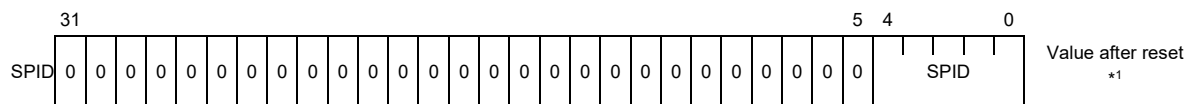


Table 3.23 SPID Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	SPID	These bits indicate the system protection identifier. The system protection identifier is a variable ID that is used for access protection in a product which consists of two or more bus masters including this CPU. For its uses and constraints on its value, see Section 3.8.7, Product Information of Initial Value for G4MH Register . Within this CPU, the SPID is used to check for area matching by the MPU. It allows the system specifications defined for the product to be reflected in the MPU's protection feature. The settable system protection identifiers are given by the SPIDLIST register. If an attempt is made to set an illegal system protection identifier, the SPID register is not updated and retains the original value.	R/W	*1

Note 1. See **Section 3.8.7, Product Information of Initial Value for G4MH Register**.

(15) SPIDLIST — Legitimate System Protection Identifier List

The SPIDLIST register contains a list of system protection identifiers that can be set to the SPID register.

The bits corresponding to the settable system protection identifiers are set to 1. The bits corresponding to illegal system protection identifiers are cleared to 0. These values are set outside the CPU as system specifications and cannot be altered by this CPU.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
SPIDLIST	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	Value after reset *1
	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		

Table 3.24 SPIDLIST Register Contents

Bit	Name	Description	R/W	Value after Reset
31	SL31	This bit indicates whether or not 31 can be set as a system protection identifier.	R	*1
30	SL30	This bit indicates whether or not 30 can be set as a system protection identifier.	R	*1
29	SL29	This bit indicates whether or not 29 can be set as a system protection identifier.	R	*1
28	SL28	This bit indicates whether or not 28 can be set as a system protection identifier.	R	*1
27	SL27	This bit indicates whether or not 27 can be set as a system protection identifier.	R	*1
26	SL26	This bit indicates whether or not 26 can be set as a system protection identifier.	R	*1
25	SL25	This bit indicates whether or not 25 can be set as a system protection identifier.	R	*1
24	SL24	This bit indicates whether or not 24 can be set as a system protection identifier.	R	*1
23	SL23	This bit indicates whether or not 23 can be set as a system protection identifier.	R	*1
22	SL22	This bit indicates whether or not 22 can be set as a system protection identifier.	R	*1
21	SL21	This bit indicates whether or not 21 can be set as a system protection identifier.	R	*1
20	SL20	This bit indicates whether or not 20 can be set as a system protection identifier.	R	*1
19	SL19	This bit indicates whether or not 19 can be set as a system protection identifier.	R	*1
18	SL18	This bit indicates whether or not 18 can be set as a system protection identifier.	R	*1
17	SL17	This bit indicates whether or not 17 can be set as a system protection identifier.	R	*1
16	SL16	This bit indicates whether or not 16 can be set as a system protection identifier.	R	*1
15	SL15	This bit indicates whether or not 15 can be set as a system protection identifier.	R	*1
14	SL14	This bit indicates whether or not 14 can be set as a system protection identifier.	R	*1
13	SL13	This bit indicates whether or not 13 can be set as a system protection identifier.	R	*1
12	SL12	This bit indicates whether or not 12 can be set as a system protection identifier.	R	*1
11	SL11	This bit indicates whether or not 11 can be set as a system protection identifier.	R	*1
10	SL10	This bit indicates whether or not 10 can be set as a system protection identifier.	R	*1
9	SL9	This bit indicates whether or not 9 can be set as a system protection identifier.	R	*1
8	SL8	This bit indicates whether or not 8 can be set as a system protection identifier.	R	*1
7	SL7	This bit indicates whether or not 7 can be set as a system protection identifier.	R	*1
6	SL6	This bit indicates whether or not 6 can be set as a system protection identifier.	R	*1
5	SL5	This bit indicates whether or not 5 can be set as a system protection identifier.	R	*1
4	SL4	This bit indicates whether or not 4 can be set as a system protection identifier.	R	*1
3	SL3	This bit indicates whether or not 3 can be set as a system protection identifier.	R	*1
2	SL2	This bit indicates whether or not 2 can be set as a system protection identifier.	R	*1
1	SL1	This bit indicates whether or not 1 can be set as a system protection identifier.	R	*1
0	SL0	This bit indicates whether or not 0 can be set as a system protection identifier.	R	*1

Note 1. See **Section 3.8.7, Product Information of Initial Value for G4MH Register**.

(16) RBASE — Reset Vector Base Address

This register indicates the reset vector address when there is a reset. If the PSW.EBV bit is 0, this register indicates the exception handler vector address and the selection method of exception handler address.

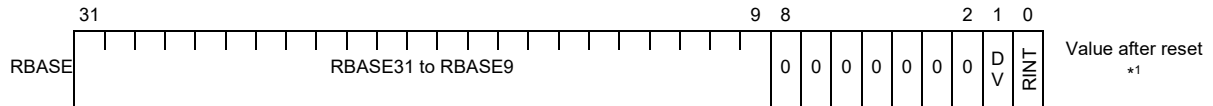


Table 3.25 RBASE Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 9	RBASE31 to RBASE9	These bits indicate the reset vector when there is a reset. When PSW.EBV = 0, this address is also used as the exception vector. The RBASE8 to RBASE0 bits are not assigned as names because these bits are always 0.	R	*1
8 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	DV	When the DV bit is set, the exception handler address for interrupt is determined by using the direct vector method. For details, see Section 3.2.4.4(1)(b), Table Reference Method . This bit is valid when PSW.EBV = 0.	R	*1
0	RINT	When the RINT bit is set, the exception handler address for interrupt processing is reduced. See Section 3.2.4.4(1)(a), Direct Vector Method . This bit is valid when PSW.EBV = 0.	R	*1

Note 1. See **Section 3.8.7, Product Information of Initial Value for G4MH Register**.

(17) EBASE — Exception Handler Vector Address

This register indicates the exception handler vector address and the selection method of exception handler address. This register is valid when the PSW.EBV bit is 1.

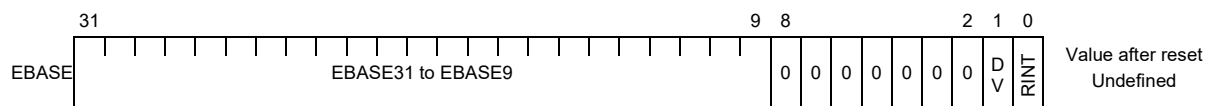


Table 3.26 EBASE Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 9	EBASE31 to EBASE9	The exception handler routine address is changed to the address resulting from adding the offset address of each exception to the base address specified for this register. The EBASE8 to EBASE0 bits are not assigned as names because these bits are always 0.	R/W	Undefined
8 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	DV	When the DV bit is set, the exception handler address for interrupt is determined by using the direct vector method. For details, see Section 3.2.4.4(1)(b), Table Reference Method .	R/W	Undefined
0	RINT	When the RINT bit is set, the exception handler address for interrupt processing is reduced. See Section 3.2.4.4(1)(a), Direct Vector Method .	R/W	Undefined

(18) INTBP — Base Address of the Interrupt Handler Address Table

This register indicates the base address of the table when the table reference method is selected as the interrupt handler address selection method.

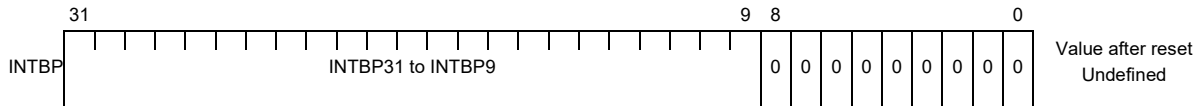


Table 3.27 INTBP Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 9	INTBP31 to INTBP9	These bits indicate the base pointer address for an interrupt when the table reference method is used. The value indicated by these bits is the first address in the table used to determine the exception handler when the interrupt specified by the table reference method (EIINTn) is acknowledged. The INTBP8 to INTBP0 bits are not assigned as names because these bits are always 0.	R/W	Undefined
8 to 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(19) MCTL — Machine Control

The MCTL register is used to control the CPU.

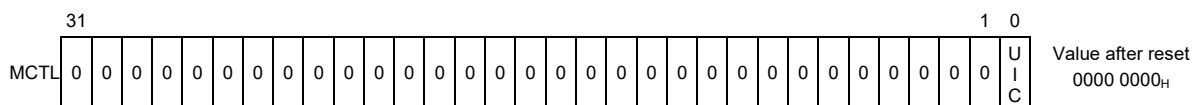


Table 3.28 MCTL Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	UIC	This bit is used to control the interrupt enable/disable operation in user mode. When this bit is set to 1, executing the EI/DI instruction in user mode become possible.	R/W	0

(20) PID — Processor ID

The PID register retains a processor identifier that is unique to the CPU. The PID register is a read-only register.

CAUTION

The PID register indicates information used to identify the incorporated CPU core and CPU core configuration. Usage such that the software behavior varies dynamically according to the PID register information is not assumed.

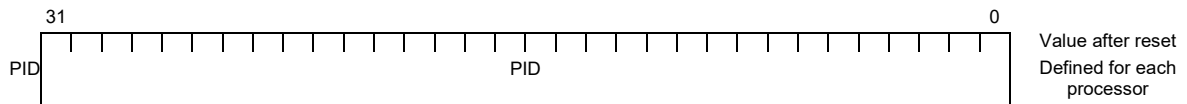


Table 3.29 PID Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 24	PID	Architecture Identifier This identifier indicates the architecture of the processor.	R	*1
23 to 8		Function Identifier This identifier indicates the functions of the processor. These bits indicate whether or not functions defined per bit are implemented (1: implemented, 0: not implemented). Bit 23 to19: Reserved Bit 18: Register bank Bit 17 to12: Reserved Bit 11: Extended floating-point operation function Bit 10: Double-precision floating-point operation function Bit 9: Single-precision floating-point operation function Bit 8: Memory protection unit (MPU) function NOTE: If a double-precision floating-point operation function is implemented (when bit 10 is 1), a single-precision floating-point operation function is also always implemented (bit 9 is 1).	R	*1
7 to 0		Version Identifier This identifier indicates the version of the processor.	R	*1

Note 1. For details, see **Section 3.8.7, Product Information of Initial Value for G4MH Register**.

(21) SVLOCK — Supervisor Lock

The SVLOCK register is used to restrict the CPU operation in supervisor mode.

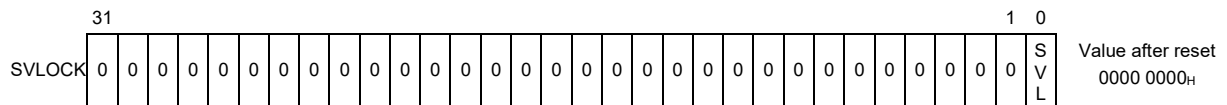


Table 3.30 SVLOCK Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	SVL	This bit specifies whether to restrict the CPU operation in supervisor mode. 0: Does not restrict the CPU operation in supervisor mode. 1: Restrict the CPU operation in supervisor mode. If the SVL bit is set to 1, the following system registers*1 cannot be updated even when the CPU is in supervisor mode: SPID, MPM, MPLA, MPUA, MPAT, MPIDn For details, see Section 3.2.2.5(5), Supervisor Lock Setting.	R/W	0

Note 1. The target system registers are those registers that are associated with memory accessing. This register prevents these registers from being rewritten carelessly and unintentional memory access from being performed outside the CPU.

(22) SCCFG — SYSCALL Operation Setting

This register is used to set operations related to the SYSCALL instruction. Be sure to set an appropriate value to this register before using the SYSCALL instruction.

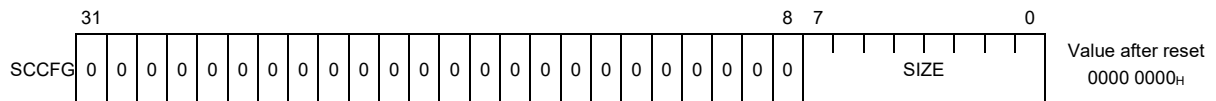


Table 3.31 SCCFG Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7 to 0	SIZE	These bits specify the maximum number of entries of a table that the SYSCALL instruction references. The maximum number of entries the SYSCALL instruction references is 1 if SIZE is 0, and 256 if SIZE is 255. By setting the maximum number of entries appropriately in accordance with the number of functions branched by the SYSCALL instruction, the memory area can be effectively used. If a vector exceeding the maximum number of entries is specified for the SYSCALL instruction, the first entry is selected. Place an error processing routine at the first entry.	R/W	0

(23) SCBP — SYSCALL Base Pointer

The SCBP register is used to specify a table address of the SYSCALL instruction and generate a target address. Be sure to set an appropriate value to this register before using the SYSCALL instruction.

Be sure to set a word address to the SCBP register.

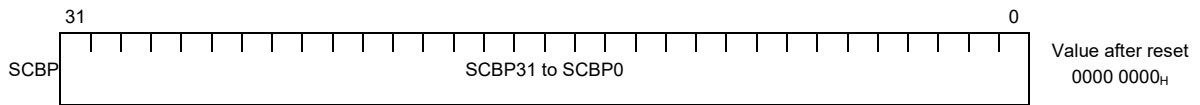


Table 3.32 SCBP Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 2	SCBP31 to SCBP2	These bits indicate the base pointer address of the SYSCALL instruction. These bits indicate the start address of the table used by the SYSCALL instruction.	R/W	0
1, 0	SCBP1, SCBP0	These bits indicate the base pointer address of the SYSCALL instruction. These bits indicate the start address of the table used by the SYSCALL instruction. Always set these bits to 0.	R	0

(24) PEID — Processor Element Identifier

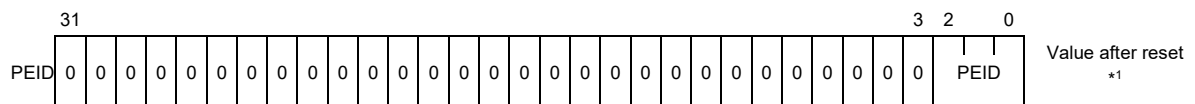


Table 3.33 PEID Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 3	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
2 to 0	PEID	These bits indicate the processor element identifier.	R	*1

Note 1. The processor element identifier of the CPU which is defined by the product specification is read. Writing to these bits is not possible. For details, see **Section 3.8.7, Product Information of Initial Value for G4MH Register**.

(25) BMID — Bus Master Identifier

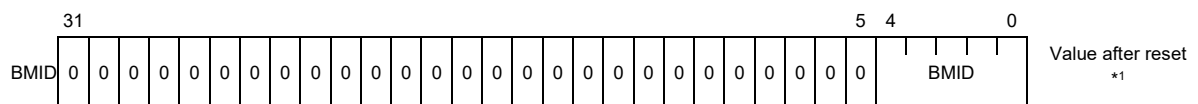


Table 3.34 BMID Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	BMID	These bits indicate the bus master identifier.	R	*1

Note 1. The bus master identifier of the CPU which is defined by the product specification is read. Writing to these bits is not possible. For details, see **Section 3.8.7, Product Information of Initial Value for G4MH Register**.

(26) MEA — Memory Error Address

Table 3.35 MEA Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 0	MEA	These bits holds the address in which an MAE (misalignment) or MPU violation occurred.	R/W	Undefined

(27) MEI — Memory Error Information

The MEI register holds the information about the instruction that caused a misalignment exception (MAE) or memory protection exception (MDP). The information can be used as hint information for the emulation by software.

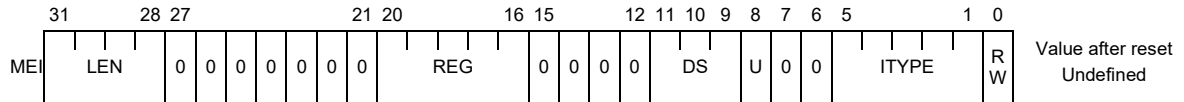


Table 3.36 MEI Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 28	LEN	These bits indicate the code size of the instruction that causes the exception. 0: Non-instruction factor 2: 16 bits 4: 32 bits 6: 48 bits 8 :64 bits Values other than those listed above are reserved for future use and never stored here. For details, see Table 3.37 .	R/W	Undefined
27 to 21	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
20 to 16	REG	These bits indicate the source register number or destination register number of the instruction that caused the exception. For details, see Table 3.37 .	R/W	Undefined
15 to 12	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
11 to 9	DS	These bits indicate the data type of the instruction that caused the exception*1. 0: Byte (8 bits) 1: Halfword (16 bits) 2: Word (32 bits) 3: Double-word (64 bits) 4: Quad-word (128 bits) Values other than those listed above are reserved for future use and never stored here. For details, see Table 3.37 .	R/W	Undefined
8	U	This bit indicates the sign extension method of the instruction that caused the exception. 0: Signed 1: Unsigned For details, see Table 3.37 .	R/W	Undefined
7 to 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5 to 1	ITYPE	These bits indicate the instruction that caused the exception. For details, see Table 3.37 .	R/W	Undefined
0	RW	This bit indicates whether the operation performed by the instruction that caused the exception is a read (Load-memory) or a write (Store-memory). 0: Read (Load-memory) 1: Write (Store-memory) For details, see Table 3.37 .	R/W	Undefined

Note 1. Even if the data is divided and access is made several times due to the specifications of the hardware, the original data type indicated by the instruction is stored.

Table 3.37 Instructions Causing Exceptions and Values of MEI Register (1/2)

Instruction	LEN	REG	DS	U	RW	ITYPE
SLD.B	2 (16 bits)	dst	0 (Byte)	0 (Signed)	0 (Read)	0000 _B
SLD.BU	2 (16 bits)	dst	0 (Byte)	1 (Unsigned)	0 (Read)	0000 _B
SLD.H	2 (16 bits)	dst	1 (Half-word)	0 (Signed)	0 (Read)	0000 _B
SLD.HU	2 (16 bits)	dst	1 (Half-word)	1 (Unsigned)	0 (Read)	0000 _B
SLD.W	2 (16 bits)	dst	2 (Word)	0 (Signed)	0 (Read)	0000 _B
SST.B	2 (16 bits)	src	0 (Byte)	0 (Signed)	1 (Write)	0000 _B
SST.H	2 (16 bits)	src	1 (Half-word)	0 (Signed)	1 (Write)	0000 _B
SST.W	2 (16 bits)	src	2 (Word)	0 (Signed)	1 (Write)	0000 _B
LD.B (disp16)	4 (32 bits)	dst	0 (Byte)	0 (Signed)	0 (Read)	00001 _B
LD.BU (disp16)	4 (32 bits)	dst	0 (Byte)	1 (Unsigned)	0 (Read)	00001 _B
LD.H (disp16)	4 (32 bits)	dst	1 (Half-word)	0 (Signed)	0 (Read)	00001 _B
LD.HU (disp16)	4 (32 bits)	dst	1 (Half-word)	1 (Unsigned)	0 (Read)	00001 _B
LD.W (disp16)	4 (32 bits)	dst	2 (Word)	0 (Signed)	0 (Read)	00001 _B
ST.B (disp16)	4 (32 bits)	src	0 (Byte)	0 (Signed)	1 (Write)	00001 _B
ST.H (disp16)	4 (32 bits)	src	1 (Half-word)	0 (Signed)	1 (Write)	00001 _B
ST.W (disp16)	4 (32 bits)	src	2 (Word)	0 (Signed)	1 (Write)	00001 _B
LD.B (disp23)	6 (48 bits)	dst	0 (Byte)	0 (Signed)	0 (Read)	00010 _B
LD.BU (disp23)	6 (48 bits)	dst	0 (Byte)	1 (Unsigned)	0 (Read)	00010 _B
LD.H (disp23)	6 (48 bits)	dst	1 (Half-word)	0 (Signed)	0 (Read)	00010 _B
LD.HU (disp23)	6 (48 bits)	dst	1 (Half-word)	1 (Unsigned)	0 (Read)	00010 _B
LD.W (disp23)	6 (48 bits)	dst	2 (Word)	0 (Signed)	0 (Read)	00010 _B
LD.DW (disp23)	6 (48 bits)	dst	3 (Double-word)	0 (Signed)	0 (Read)	00010 _B
ST.B (disp23)	6 (48 bits)	src	0 (Byte)	0 (Signed)	1 (Write)	00010 _B
ST.H (disp23)	6 (48 bits)	src	1 (Half-word)	0 (Signed)	1 (Write)	00010 _B
ST.W (disp23)	6 (48 bits)	src	2 (Word)	0 (Signed)	1 (Write)	00010 _B
ST.DW (disp23)	6 (48 bits)	src	3 (Double-word)	0 (Signed)	1 (Write)	00010 _B
LD.B (+)	4 (32 bits)	dst	0 (Byte)	0 (Signed)	0 (Read)	00100 _B
LD.BU (+)	4 (32 bits)	dst	0 (Byte)	1 (Unsigned)	0 (Read)	00100 _B
LD.H (+)	4 (32 bits)	dst	1 (Half-word)	0 (Signed)	0 (Read)	00100 _B
LD.HU (+)	4 (32 bits)	dst	1 (Half-word)	1 (Unsigned)	0 (Read)	00100 _B
LD.W (+)	4 (32 bits)	dst	2 (Word)	0 (Signed)	0 (Read)	00100 _B
ST.B (+)	4 (32 bits)	src	0 (Byte)	0 (Signed)	1 (Write)	00100 _B
ST.H (+)	4 (32 bits)	src	1 (Half-word)	0 (Signed)	1 (Write)	00100 _B
ST.W (+)	4 (32 bits)	src	2 (Word)	0 (Signed)	1 (Write)	00100 _B
LD.B (-)	4 (32 bits)	dst	0 (Byte)	0 (Signed)	0 (Read)	00101 _B
LD.BU (-)	4 (32 bits)	dst	0 (Byte)	1 (Unsigned)	0 (Read)	00101 _B
LD.H (-)	4 (32 bits)	dst	1 (Half-word)	0 (Signed)	0 (Read)	00101 _B
LD.HU (-)	4 (32 bits)	dst	1 (Half-word)	1 (Unsigned)	0 (Read)	00101 _B
LD.W (-)	4 (32 bits)	dst	2 (Word)	0 (Signed)	0 (Read)	00101 _B
ST.B (-)	4 (32 bits)	src	0 (Byte)	0 (Signed)	1 (Write)	00101 _B
ST.H (-)	4 (32 bits)	src	1 (Half-word)	0 (Signed)	1 (Write)	00101 _B
ST.W (-)	4 (32 bits)	src	2 (Word)	0 (Signed)	1 (Write)	00101 _B
LDL.BU	4 (32 bits)	dst	0 (Byte)	1 (Unsigned)	0 (Read)	00111 _B
LDL.HU	4 (32 bits)	dst	1 (Half-word)	1 (Unsigned)	0 (Read)	00111 _B
LDL.W	4 (32 bits)	dst	2 (Word)	0 (Signed)	0 (Read)	00111 _B

Table 3.37 Instructions Causing Exceptions and Values of MEI Register (2/2)

Instruction	LEN	REG	DS	U	RW	ITYPE
STC.B	4 (32 bits)	src	0 (Byte)	0 (Signed)	1 (Write)	00111 _B
STC.H	4 (32 bits)	src	1 (Half-word)	0 (Signed)	1 (Write)	00111 _B
STC.W	4 (32 bits)	src	2 (Word)	0 (Signed)	1 (Write)	00111 _B
CAXI	4 (32 bits)	dst* ¹	2 (Word)* ¹	0 (Signed)* ¹	0 (Read)* ²	01000 _B
SET1	4 (32 bits)	0* ¹	0 (Byte)* ¹	0 (Signed)* ¹	0 (Read)* ²	01001 _B
CLR1	4 (32 bits)	0* ¹	0 (Byte)* ¹	0 (Signed)* ¹	0 (Read)* ²	01001 _B
NOT1	4 (32 bits)	0* ¹	0 (Byte)* ¹	0 (Signed)* ¹	0 (Read)* ²	01001 _B
TST1	4 (32 bits)	0* ¹	0 (Byte)* ¹	0 (Signed)* ¹	0 (Read)	01001 _B
PREPARE	* ⁵ ,* ⁶	src* ¹	2 (Word)* ¹	0 (Signed)* ¹	1 (Write)	01100 _B
DISPOSE	4 (32 bits)	dst* ¹	2 (Word)* ¹	0 (Signed)* ¹	0 (Read)	01100 _B
PUSHSP	4 (32 bits)	src* ¹	2 (Word)* ¹	0 (Signed)* ¹	1 (Write)	01101 _B
POPSP	4 (32 bits)	dst* ¹ ,* ³	2 (Word)* ¹	0 (Signed)* ¹	0 (Read)	01101 _B
SWITCH	2 (16 bits)	0* ¹	1 (Half-word)* ¹	0 (Signed)* ¹	0 (Read)	10000 _B
CALLT	2 (16 bits)	0* ¹	1 (Half-word)* ¹	1 (Unsigned)* ¹	0 (Read)	10001 _B
SYSCALL	4 (32 bits)	0* ¹	2 (Word)* ¹	0 (Signed)* ¹	0 (Read)	10010 _B
CACHE	4 (32 bits)	0* ¹	0 (Byte)* ¹	0 (Signed)* ¹	0 (Read)	10100 _B
Interrupt (table reference method)* ⁴	0 (Non-instruction)	0* ¹	2 (Word)* ¹	0 (Signed)* ¹	0 (Read)	10101 _B
Save onto register bank	0 (Non-instruction)	0* ¹	2 (Word)* ¹	0 (Signed)* ¹	1 (Write)	10110 _B
RESBANK	4 (32 bits)	0* ¹	2 (Word)* ¹	0 (Signed)* ¹	0 (Read)	10110 _B
LDV.W (disp16)	6 (48 bits)	dst	2 (Word)	0 (Signed)	0 (Read)	11101 _B
LDV.DW (disp16)	6 (48 bits)	dst	3 (Double-word)	0 (Signed)	0 (Read)	11101 _B
LDV.QW (disp16)	6 (48 bits)	dst	4 (Quad-word)	0 (Signed)	0 (Read)	11101 _B
STV.W (disp16)	6 (48 bits)	src	2 (Word)	0 (Signed)	1 (Write)	11101 _B
STV.DW (disp16)	6 (48 bits)	src	3 (Double-word)	0 (Signed)	1 (Write)	11101 _B
STV.QW (disp16)	6 (48 bits)	src	4 (Quad-word)	0 (Signed)	1 (Write)	11101 _B
LDVZ.H4 (disp16)	6 (48 bits)	dst	3 (Double-word)	0 (Signed)	0 (Read)	11111 _B
STVZ.H4 (disp16)	6 (48 bits)	src	3 (Double-word)	0 (Signed)	1 (Write)	11111 _B

Remarks: dst: Destination register number; src: Source register number

Note 1. Specific to this CPU.

Note 2. This exception occurs when the instruction executes a read access.

Note 3. When the destination is r3, 0 is stored.

Note 4. When an exception occurs during a table reference that is triggered by an interrupt for which the table reference method is selected,

Note 5. In instruction format (1), "4 (32 bits)" is set.

Note 6. In instruction format (2), the value is selected as follows according to the value of the ff field in the instruction code:

ff = 00_B: 4 (32 bits)

ff = 01_B: 6 (48 bits)

ff = 10_B: 6 (48 bits)

ff = 11_B: 8 (64 bits)

(28) RBCR0 — Register Bank Control 0

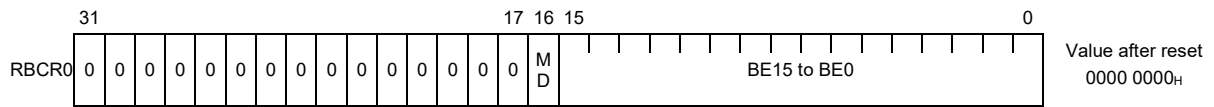


Table 3.38 RBCR0 Register Contents

Bit	Name	Description	R/W	Value after Reset																											
31 to 17	—	(Reserved for future expansion. Be sure to set to 0.)	R	0																											
16	MD	This bit specifies the save mode of the register bank. 0: Save mode 0 1: Save mode 1	R/W	0																											
List of Registers to be Saved																															
<table border="1"> <thead> <tr> <th>Target Register</th> <th>Save Mode 0</th> <th>Save Mode 1</th> </tr> </thead> <tbody> <tr><td>PC</td><td>✓</td><td>✓</td></tr> <tr><td>PSW</td><td>✓</td><td>✓</td></tr> <tr><td>EIIC</td><td>✓</td><td>✓</td></tr> <tr><td>FPSR</td><td>✓</td><td>✓</td></tr> <tr><td>r1-r19</td><td>✓</td><td>✓</td></tr> <tr><td>r20-r29</td><td>—</td><td>✓</td></tr> <tr><td>r30</td><td>✓</td><td>✓</td></tr> <tr><td>r31</td><td>—</td><td>✓</td></tr> </tbody> </table>					Target Register	Save Mode 0	Save Mode 1	PC	✓	✓	PSW	✓	✓	EIIC	✓	✓	FPSR	✓	✓	r1-r19	✓	✓	r20-r29	—	✓	r30	✓	✓	r31	—	✓
Target Register	Save Mode 0	Save Mode 1																													
PC	✓	✓																													
PSW	✓	✓																													
EIIC	✓	✓																													
FPSR	✓	✓																													
r1-r19	✓	✓																													
r20-r29	—	✓																													
r30	✓	✓																													
r31	—	✓																													
✓: Saved —: Not saved																															
15 to 0	BE15 to BE0	These bits specify the priority level of the interrupt on which the register bank is to be used. 0: Do not use the register bank on an interrupt of the level corresponding to this bit. 1: Use the register bank on an interrupt of the level corresponding to this bit.	R/W	0																											
The bit positions correspond to the priority levels of the interrupts as follows:																															
<table border="1"> <thead> <tr> <th>Bit</th> <th>Priority Level</th> </tr> </thead> <tbody> <tr><td>0</td><td>Level 0 (highest priority)</td></tr> <tr><td>1</td><td>Level 1</td></tr> <tr><td colspan="2" style="text-align: center;">⋮</td></tr> <tr><td>14</td><td>Level 14</td></tr> <tr><td>15</td><td>Level 15 (lowest priority)</td></tr> </tbody> </table>					Bit	Priority Level	0	Level 0 (highest priority)	1	Level 1	⋮		14	Level 14	15	Level 15 (lowest priority)															
Bit	Priority Level																														
0	Level 0 (highest priority)																														
1	Level 1																														
⋮																															
14	Level 14																														
15	Level 15 (lowest priority)																														
The register bank is used when an interrupt (EIINT _n) of a level is accepted while the bit corresponding to that level is set (1).																															

(29) RBCR1 — Register Bank Control 1

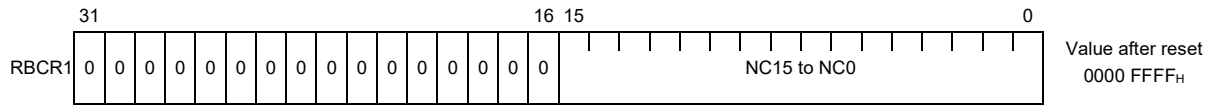


Table 3.39 RBCR1 Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 16	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
15 to 0	NC15 to NC0	These bits specify the values to be set to the PSW.ID bit when an interrupt (EIINT _n) of the priority level corresponding to the bit position is accepted. 0: PSW.ID is set to 0 when an interrupt of the priority level corresponding to the bit position is accepted. 1: PSW.ID is set to 1 when an interrupt of the priority level corresponding to the bit position is accepted (initial value).	R/W	FFFF _H

This setting, however, is enabled only when the interrupt handler address selection method is set to the table reference method and the use of the register banks is enabled.

The bit positions correspond to the priority levels of the interrupts as follows:

Bit	Priority Level
0	Level 0 (highest priority)
1	Level 1
⋮	
14	Level 14
15	Level 15 (Lowest priority)

(30) RBNR — Register Bank Number

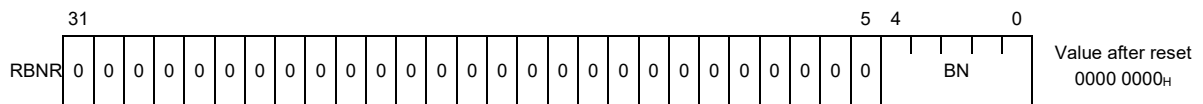


Table 3.40 RBNR Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	BN	These bits indicate the number of the register bank to be used next. The value in these bits is incremented by 1 when an interrupt (EIINT _n) that uses a register bank is accepted. If an interrupt that uses a register bank occurs when the value of BN is greater than 15, the interrupt is not accepted and held pending, and a SYSERR exception is generated. The value of BN is decremented by 1 when a RESBANK instruction is executed. If the RESBANK instruction is executed when the value of BN is 0, the instruction is not executed and a SYSERR exception is generated. In this case, BN is not updated and its value remains the same. If an attempt is made to write a value of 16 or greater to these bits, the value of BN becomes 16.	R/W	0

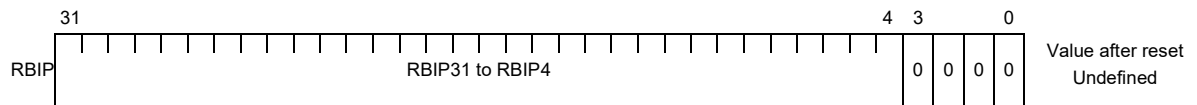
(31) RBIP — Register Bank Initial Pointer

Table 3.41 RBIP Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 4	RBIP31 to RBIP4	<p>These bits indicate the initial pointer to the register bank.</p> <p>The values of RBCR0.MD and RBNR.BN together with this value determine the memory address to which the next register bank to be used is allocated as follows.</p> <ul style="list-style-type: none"> • Save mode 0 (RBCR0.MD = 0): $RBIP - RBNR.BN \times 60_H$ • Save mode 1 (RBCR0.MD = 1): $RBIP - RBNR.BN \times 90_H$ 	R/W	Undefined
3 to 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

3.2.3.3 Interrupt Function Registers

(1) Interrupt Function System Registers

Interrupt function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.42 Interrupt Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR10, 2	ISPR	Priority of interrupt being serviced	SV
SR12, 2	ICSR	Interrupt control status	SV
SR13, 2	INTCFG	Interrupt function setting	SV
SR14, 2	PLMR	Interrupt priority masking	SV

(a) ISPR — Priority of Interrupt being Serviced

The ISPR register holds the priority levels of the EIINT n interrupts that are being processed by the CPU so that priority sealing can be performed based on interrupt priorities when multiple interrupts occurred.

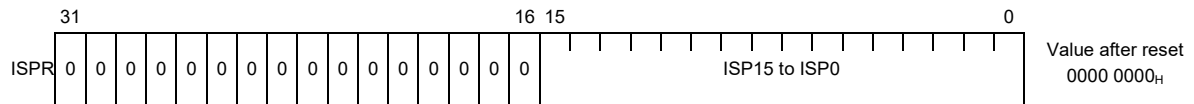


Table 3.43 ISPR Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 16	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
15 to 0	ISP15 to ISP0	These bits indicate the acknowledgment status of an EIINT n interrupt with a priority*1 that corresponds to the relevant bit position. 0: An interrupt request for an interrupt whose priority corresponds to the relevant bit position has not been acknowledged. 1: An interrupt request for an interrupt whose priority corresponds to the relevant position is being serviced by the CPU core.	R*3	0

The bit positions correspond to the priority levels of the interrupts as follows:

Bit	Priority
0	Priority 0 (highest)
1	Priority 1
:	
14	Priority 14
15	Priority 15 (lowest)

When an interrupt request (EIINT n) is acknowledged, the bit corresponding to the acknowledged interrupt request is automatically set to 1. If PSW.EP is 0 when the EIRET instruction is executed, the bit with the highest priority among the ISP15 to ISP0 bits that are set (0 is the highest priority) is cleared to 0*2. While a bit in this register is set to 1, same or lower priority interrupts (EIINT n) are masked. Priority level judgment is therefore not performed when the system is determining whether to acknowledge an exception, meaning that exceptions will not be acknowledged.

For details, see **Section 3.2.4.1(5), Interrupt Exception Priority and Priority Masking.**

When performing software-based priority control using the PLMR register, be sure to clear this register by using the INTCFG.ISPC bit.

Note 1. For details, see **Section 3.2.4.1(5), Interrupt Exception Priority and Priority Masking.**

Note 2. Interrupt acknowledgment and auto-updating of values when the EIRET instruction is executed are disabled by setting (1) the INTCFG.ISPC bit. It is recommended to enable auto-updating of values, so in normal cases, the INTCFG.ISPC bit should be cleared to 0.

Note 3. This is R or R/W, depending on the setting of the INTCFG.ISPC bit. It is recommended to use this register as a read-only (R) register.

(b) ICSR — Interrupt Control Status

This register indicates the interrupt control status in the CPU.

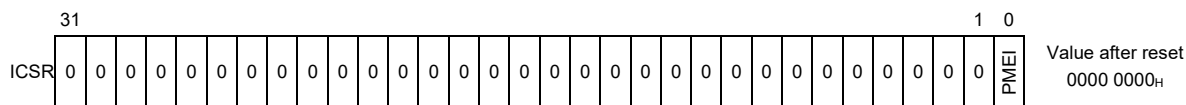


Table 3.44 ICSR Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	PMEI	This bit indicates that an interrupt (EIINTn) with the priority level masked by the PLMR register exists. For details, see Section 3.2.4.1(5), Interrupt Exception Priority and Priority Masking .	R	0

(c) INTCFG — Interrupt Function Setting

This register is used to specify settings related to the CPU’s internal interrupt function.

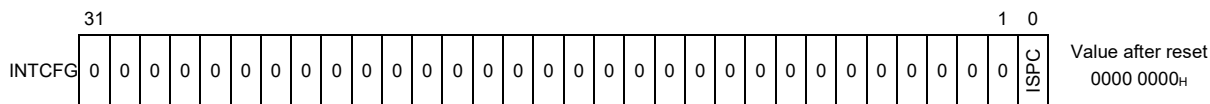


Table 3.45 INTCFG Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	ISPC	This bit changes how the ISPR register is written. 0: The ISPR register is automatically updated. Updates triggered by the program (via execution of LDSR instruction) are ignored. 1: The ISPR register is not automatically updated. Updates triggered by the program (via execution of LDSR instruction) are performed. If this bit is cleared to 0, the bits of the ISPR register are automatically set to 1 when an interrupt (EIINTn) is acknowledged, and cleared to 0 when the EIRET instruction is executed. In this case, the bits are not updated by an LDSR instruction executed by the program. If this bit is set to 1, the bits of the ISPR register are not updated by the acknowledgement of an interrupt (EIINTn) or by execution of the EIRET instruction. In this case, the bits can be updated by an LDSR instruction executed by the program. In normal cases, the ISPC bit should be cleared. When performing software-based priority control, however, set this bit (1). For details, see Section 3.2.4.1(5)(b), Interrupt Priority Mask .	R/W	0

(d) PLMR — Interrupt Priority Level Mask

This register masks the interrupts (EIINT n) whose priority level is not higher than the level specified by these bits.

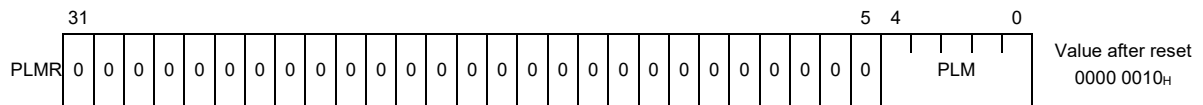


Table 3.46 PLMR Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	PLM	These bits are used to mask the interrupts (EIINT n) whose priority level is not higher than the level specified by these bits. When an interrupt (EIINT n) is masked by this register, it is not accepted.	R/W	10 _H

The correspondence between the value of the PLM bit and the highest priority of interrupts to be masked is shown below.

Value of PLM Bit	Highest Priority of Interrupts to be Masked
0	Priority 0 (highest priority)
1	Priority 1
:	
14	Priority 14
15	Priority 15 (lowest priority)
16 or greater	No mask

Since the highest priority level of an interrupt that is defined for this CPU is 0, if 0 is specified in the PLM bit, all interrupts (EIINT n) are masked by this register. Since the lowest priority level that is defined for this CPU is 15, no interrupts are masked by this register if 16 or greater value is specified in the PLM bit.

If the value of PLMR is altered by the LDSR instruction, the new PLMR value is reflected in the instructions following that LDSR instruction.

3.2.3.4 FPU Function Registers

(1) Floating-Point Registers

The FPU uses the CPU general-purpose registers (r0 to r31). There are no register files used only for floating-point operations.

- Single-precision floating-point instruction:
Thirty-two 32-bit registers can be specified. These general-purpose registers correspond to r0 to r31.
- Double-precision floating-point instruction:
Sixteen 64-bit registers can be specified. Paired general-purpose registers are used as register pairs ($\{r1, r0\}$, $\{r3, r2\}$... $\{r31, r30\}$). Each register pair is specified in the instruction format with an even numbered register. Because r0 is a zero register (always holds 0), in principle $\{r1, r0\}$ cannot be used by a double-precision floating-point instruction.

(2) Floating-Point Function System Registers

The FPU can use the following system registers to control floating-point operations. Floating-point function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

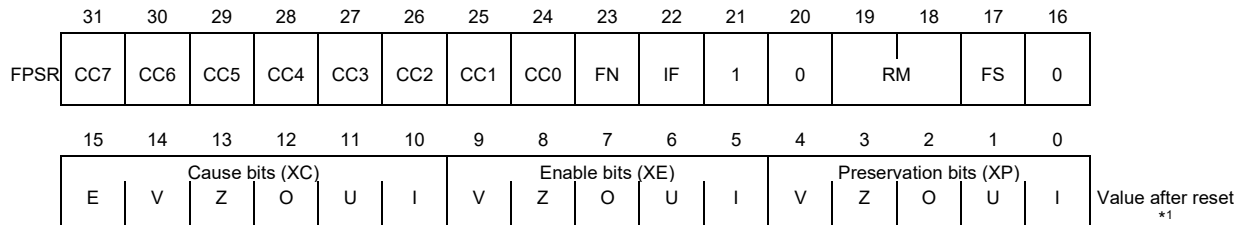
- FPSR: This register is used to control and monitor exceptions. It also holds the result of compare operations, and sets the FPU operation mode. Its bits are used to set condition code, subnormal number flush enable, rounding mode control, cause, exception enable, and preservation.
- FPEPC: This register stores the program counter value for the instruction where a floating-point operation exception has occurred.
- FPST: This register reflects the contents of the FPSR register bits related to the operation status.
- FPCC: This register reflects the contents of the FPSR.CC (7:0) bits.
- FPCFG: This register reflects the contents of the FPSR register bits related to the operation settings.

Table 3.47 FPU System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR6, 0	FPSR	Floating-point operation configuration/status	CU0 and SV
SR7, 0	FPEPC	Floating-point operation exception program counter	CU0 and SV
SR8, 0	FPST	Floating point operation status	CU0
SR9, 0	FPCC	Floating-point operation comparison result	CU0
SR10, 0	FPCFG	Floating-point operation configuration	CU0

(a) FPSR — Floating-point Configuration/Status

This register indicates the execution status of floating-point operations and any exceptions that occur. For details about exception, see **Section 3.2.6.1(5), Floating-Point Operation Exceptions**.



Note 1. See the descriptions of each bit.

Table 3.48 FPSR Register Contents (1/2)

Bit	Name	Description	R/W	Value after Reset																						
31 to 24	CC7 to CC0	These are the CC (condition) bits. They store the results of floating-point comparison instructions. The CC7 to CC0 bits are not affected by any instructions except the comparison instruction and LDSR instruction. 0: Comparison result is false 1: Comparison result is true	R/W	Undefined																						
23	FN	This bit enables flush-to-nearest mode. When the FN bit is set to 1, if the rounding mode is RN and the operation result is a subnormal number, the number is flushed to the nearest number. For details, see Section 3.2.6.1(9), Flush to Nearest .	R/W	0																						
22	IF	This bit accumulates and indicates information about the flushing of input operands. For details about flushing subnormal numbers, see Section 3.2.6.1(8), Flushing Subnormal Numbers .	R/W	0																						
21	—	(Reserved for future expansion. Be sure to set to 1.)	R	1																						
20	—	(Reserved for future expansion. Be sure to set to 0.)	R	0																						
19, 18	RM	These are the rounding mode control bits. The RM bits define the rounding mode that the FPU uses for all floating-point instructions.	R/W	00																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e0e0e0;"> <th colspan="2">RM Bits</th> <th rowspan="2">Mnemonic</th> <th rowspan="2">Description</th> </tr> <tr style="background-color: #e0e0e0;"> <th>19</th> <th>18</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>RN</td> <td>Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.</td> </tr> <tr> <td>0</td> <td>1</td> <td>RZ</td> <td>Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.</td> </tr> <tr> <td>1</td> <td>0</td> <td>RP</td> <td>Rounds the result toward $+\infty$. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.</td> </tr> <tr> <td>1</td> <td>1</td> <td>RM</td> <td>Rounds the result toward $-\infty$. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.</td> </tr> </tbody> </table>					RM Bits		Mnemonic	Description	19	18	0	0	RN	Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.	0	1	RZ	Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.	1	0	RP	Rounds the result toward $+\infty$. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.	1	1	RM	Rounds the result toward $-\infty$. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.
RM Bits		Mnemonic	Description																							
19	18																									
0	0	RN	Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.																							
0	1	RZ	Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.																							
1	0	RP	Rounds the result toward $+\infty$. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.																							
1	1	RM	Rounds the result toward $-\infty$. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.																							

Table 3.48 FPSR Register Contents (2/2)

Bit	Name	Description	R/W	Value after Reset																			
17	FS	This bit enables values that could not be normalized (subnormal numbers) to be flushed. If the FS bit is set, input operands and operation results that are subnormal numbers are flushed without causing an unimplemented operation exception (E). An input operand that is a subnormal number is flushed to 0 with the same sign. Operation results that are subnormal numbers either become 0 or the minimum normalized number, depending on the rounding mode.	R/W	1																			
		<table border="1"> <thead> <tr> <th rowspan="2">Operation Result that is a Subnormal Number</th> <th colspan="4">Rounding Mode and Value after Flushing</th> </tr> <tr> <th>RN*1</th> <th>RZ</th> <th>RP</th> <th>RM</th> </tr> </thead> <tbody> <tr> <td>Positive</td> <td>+0</td> <td>+0</td> <td>+2^{E_{min}}</td> <td>+0</td> </tr> <tr> <td>Negative</td> <td>-0</td> <td>-0</td> <td>-0</td> <td>-2^{E_{min}}</td> </tr> </tbody> </table> <p>Note 1. If the rounding mode is RN and the FPSR.FN bit is set, flushing will occur in the direction of higher accuracy. For details, see Section 3.2.6.1(9), Flush to Nearest.</p>	Operation Result that is a Subnormal Number	Rounding Mode and Value after Flushing				RN*1	RZ	RP	RM	Positive	+0	+0	+2 ^{E_{min}}	+0	Negative	-0	-0	-0	-2 ^{E_{min}}		
Operation Result that is a Subnormal Number	Rounding Mode and Value after Flushing																						
	RN*1	RZ	RP	RM																			
Positive	+0	+0	+2 ^{E_{min}}	+0																			
Negative	-0	-0	-0	-2 ^{E_{min}}																			
16	—	(Reserved for future expansion. Be sure to set to 0.)	R	0																			
15 to 10	XC (E,V,Z,O,U,I)	These are the cause bits. For details, see Section 3.2.3.4.2.a.1, Cause Bits (XC) .	R/W	Undefined																			
9 to 5	XE (V,Z,O,U,I)	These are the enable bits. For details, see Section 3.2.3.4.2.a.2, Enable Bits (XE) .	R/W	0																			
4 to 0	XP (V,Z,O,U,I)	These are the preservation bits. For details, see Section 3.2.3.4.2.a.3, Preservation Bits (XP) .	R/W	Undefined																			

3.2.3.4.2.a.1 Cause Bits (XC)

Bits 15 to 10 in the FPSR register are cause bits, which indicate the occurrence and cause of a floating-point operation exception. If an exception defined by IEEE754 is generated, when an enable bit is set to 1 corresponding to the exception, a cause bit is set, and the exception then occurs. When two or more exceptions occur during a single instruction, each corresponding bit is set to 1.

If two or more exceptions are detected, as long as the enable bit corresponding to one of the exceptions is set to 1, the exception occurs. In this case, the cause bits of all the detected exceptions, including exceptions whose enable bits are cleared to 0, are set to 1.

The cause bits are rewritten by a floating-point instruction (except the TRFSR instruction) where the floating-point operation exception occurred. The E bit is set to 1 when software emulation is required, otherwise it is cleared to 0. Other bits are set to 1 or cleared to 0 depending on whether or not an IEEE754-defined exception has occurred.

When a floating-point operation exception has occurred, the operation result is not stored, and only the cause bits are affected.

When the cause bits are set to 1 by an LDSR instruction, a floating-point operation exception does not occur.

3.2.3.4.2.a.2 Enable Bits (XE)

Bits 9 to 5 in the FPSR register are the enable bits, which enable floating-point operation exceptions. When an IEEE754-defined exception occurs, a floating-point operation exception occurs if the enable bit corresponding to the exception has been set to 1.

There are no enable bits corresponding to an unimplemented operation exception (E). An unimplemented operation exception (E) always occurs as a floating-point operation exception.

If the corresponding enable bit has not been set to 1, no exception occurs and the default result defined by IEEE754 is stored.

3.2.3.4.2.a.3 Preservation Bits (XP)

Bits 4 to 0 in the FPSR register are preservation bits. These bits store and indicate the detected exception after reset. An exception defined by IEEE754 occurs, and if a floating-point operation exception is not generated, the preservation bit is set to 1, otherwise it does not change. The preservation bits are not cleared to 0 by the floating-point operation. However, these bits can be set and cleared by software when an LDSR instruction is used to write a new value to the FPSR register.

There are no preservation bits corresponding to unimplemented operation exceptions (E). An unimplemented operation exception (E) always occurs as a floating-point operation exception.

NOTE

For details about the exception types and how they relate to particular bits, see **Figure 3.37, Cause, Enable, and Preservation Bits of FPSR Register**.

(b) FPEPC — Floating-point Exception Program Counter

When an exception that is enabled by an enable bit occurs, the program counter (PC) of the instruction that caused the exception is stored.

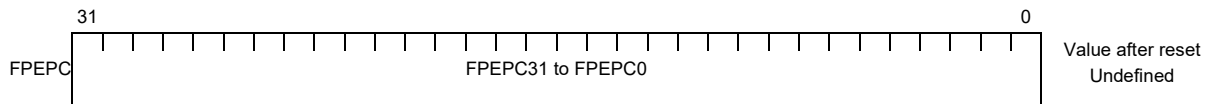


Table 3.49 FPEPC Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 1	FPEPC31 to FPEPC1	These bits store the program counter (PC) of the floating-point instruction that caused the exception when a floating-point operation exception that is enabled by an enable bit occurs.	R/W	Undefined
0	FPEPC0	This bit stores the program counter (PC) of the floating-point instruction that caused the exception when a floating-point operation exception that is enabled by an enable bit occurs.	R	0

(c) FPST — Floating-point Operation Status

This register reflects the contents of the FPSR register bits related to the operation status.

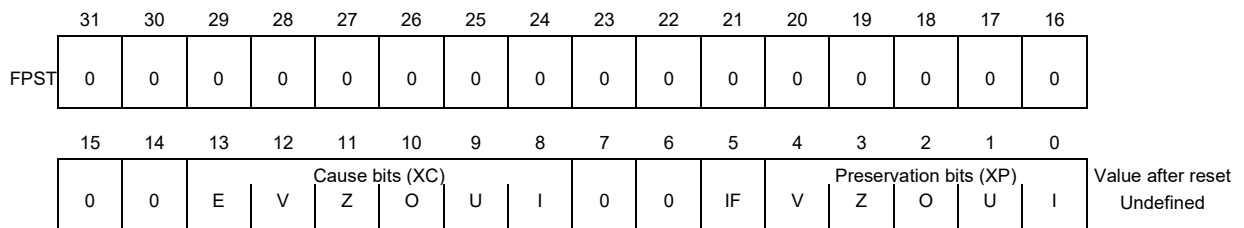


Table 3.50 FPST Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 14	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
13 to 8	XC (E, V, Z, O, U, I)	These are cause bits. For details, see Section 3.2.3.4.2.a.1, Cause Bits (XC) . Values written to these bits are reflected in FPSR.XC bits.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5	IF	This bit accumulates and indicates information about the flushing of input operands. For details about flushing subnormal numbers, see Section 3.2.6.1(8), Flushing Subnormal Numbers . Value written to this bit is reflected in FPSR.IF bit.	R/W	0
4 to 0	XP (V, Z, O, U, I)	These are preservation bits. For details, see Section 3.2.3.4.2.a.3, Preservation Bits (XP) . Values written to these bits are reflected in FPSR.XP bits.	R/W	Undefined

(d) FPCC — Floating-point Operation Comparison Result

This register reflects the contents of the FPSR.CC7 to FPSR.CC0 bits.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
FPC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	0	0	0	0	0	0	0	0	CC7	CC6	CC5	CC4	CC3	CC2	CC1	CC0	Value after reset Undefined

Table 3.51 FPCC Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7 to 0	CC7 to CC0	These are CC (condition) bits. They store the result of a floating-point comparison instruction. The CC7 to CC0 bits are not affected by any instructions except the comparison instruction and LDSR instruction. Values written to these bits are reflected in the CC7 to CC0 bits of FPSR. 0: Comparison result is false 1: Comparison result is true	R/W	Undefined

(e) FPCFG — Floating-point Operation Configuration

This register reflects the contents of the FPSR register bits related to the operation settings.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
FPCFG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	0	0	0	0	0	0	RM	0	0	0	V	Z	O	U	I	Value after reset 0000 0000 _H	

Table 3.52 FPCFG Register Contents

Bit	Name	Description	R/W	Value after Reset																						
31 to 10	—	(Reserved for future expansion. Be sure to set to 0.)	R	0																						
9, 8	RM	These are rounding mode control bits. The RM bits define the rounding mode that the FPU uses for all floating-point instructions. Values written to these bits are reflected in RM bits of FPSR.	R/W	0																						
		<table border="1"> <thead> <tr> <th colspan="2">RM Bits</th> <th rowspan="2">Mnemonic</th> <th rowspan="2">Description</th> </tr> <tr> <th>9</th> <th>8</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>RN</td> <td>Rounds the result to the nearest representable value. If the value is exactly in-between the two representable values, the result is rounded toward the value whose least significant bit is 0.</td> </tr> <tr> <td>0</td> <td>1</td> <td>RZ</td> <td>Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.</td> </tr> <tr> <td>1</td> <td>0</td> <td>RP</td> <td>Rounds the result toward $+\infty$. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.</td> </tr> <tr> <td>1</td> <td>1</td> <td>RM</td> <td>Rounds the result toward $-\infty$. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.</td> </tr> </tbody> </table>	RM Bits		Mnemonic	Description	9	8	0	0	RN	Rounds the result to the nearest representable value. If the value is exactly in-between the two representable values, the result is rounded toward the value whose least significant bit is 0.	0	1	RZ	Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.	1	0	RP	Rounds the result toward $+\infty$. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.	1	1	RM	Rounds the result toward $-\infty$. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.		
RM Bits		Mnemonic	Description																							
9	8																									
0	0	RN	Rounds the result to the nearest representable value. If the value is exactly in-between the two representable values, the result is rounded toward the value whose least significant bit is 0.																							
0	1	RZ	Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.																							
1	0	RP	Rounds the result toward $+\infty$. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.																							
1	1	RM	Rounds the result toward $-\infty$. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.																							
7 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0																						
4 to 0	XE (V,Z,O,U,I)	These are the enable bits. For details, see Section 3.2.3.4.2.a.2, Enable Bits (XE) . Values written to these bits are reflected in the FPSR.XE bits.	R/W	0																						

3.2.3.5 FXU Function Registers

(1) Vector Registers

32 vector registers (wr0-wr31) are provided as data registers dedicated to the extended floating-point operation unit (FXU). All of these registers can be used to store data variables. The data width of these registers is 128 bits. The values of the vector registers wr0 to wr31 after reset are undefined.

As shown in **Figure 3.15** and **Table 3.53**, one vector register consists of 4 arithmetic ways. One single-precision floating-point data is stored in one arithmetic way. This allows four single-precision floating-point operations to be performed with one extended floating-point operation instruction.

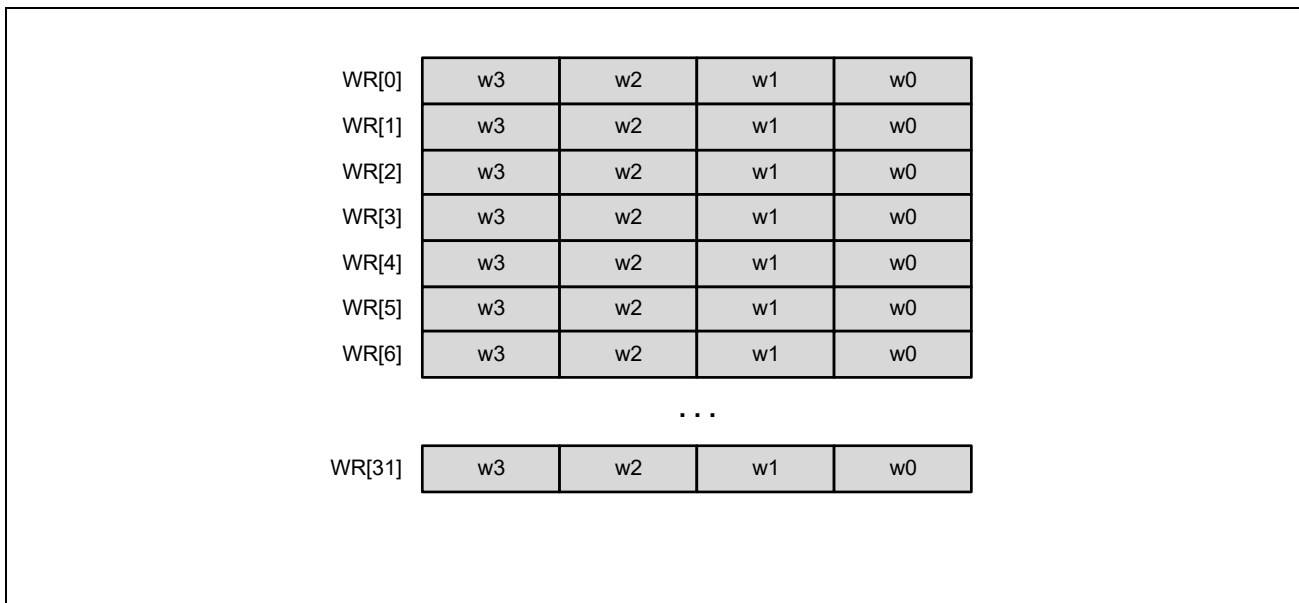


Figure 3.15 Vector Register Configuration

Table 3.53 Way Configuration of Vector Registers

Name	Bits	Function
W0	[31:0]	Way 0 data
W1	[63:32]	Way 1 data
W2	[95:64]	Way 2 data
W3	[127:96]	Way 3 data

(2) Extended Floating-point Function System Registers

The FXU can use the following system registers to control arithmetic operations. Extended floating-point function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

- **FXSR:** This register is used to control and monitor the extended floating-point operation exceptions. It is also used to set the operation mode of the FXU. It has bits for subnormal flush enabling, rounding mode control, exception cause, exception enabling and preservation.
- **FXST:** This register reflects the contents of the FXSR register bits related to the operation status.
- **FXINFO:** This register holds the information about the configuration of the FXU.
- **FXCFG:** This register reflects the contents of the FXSR register bits related to the operation settings.
- **FXXC:** This register holds the XC (cause) bits of extended floating-point operation exceptions for each arithmetic way.
- **FXXP:** This register holds the XP (preservation) bits of extended floating-point operation exceptions for each arithmetic way.

Table 3.54 List of Extended Floating-point Function System Registers

Register Number (regID, selID)	Name	Function	Access Permission
SR6, 10	FXSR	Extended floating-point operation configuration/status	CU1 and SV
SR8, 10	FXST	Extended floating-point operation status	CU1
SR9, 10	FXINFO	FXU configuration information	CU1
SR10, 10	FXCFG	Extended floating-point operation configuration	CU1
SR12, 10	FXXC	XC (cause) bits for each arithmetic way	CU1
SR13, 10	FXXP	XP (preservation) bits for each arithmetic way	CU1

(a) FXSR — Extended Floating-point Operation Configuration/Status

The FXSR register indicates the execution status of extended floating-point operations and occurrence of exceptions.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FXSR	0	0	0	0	0	0	0	0	FN	IF	1	0	RM		FS	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Cause bits (XC)					Enable (XE)					Preservation bits (XP)					Value after reset *1
	E	V	Z	O	U	I	V	Z	O	U	I	V	Z	O	U	I

Note 1. See the descriptions of each bit.

Table 3.55 FXSR Register Contents (1/2)

Bit	Name	Description	R/W	Value after Reset																						
31 to 24	—	(Reserved for future expansion. Be sure to set to 0.)	R	0																						
23	FN	This bit enables flush-to-nearest mode. When the FN bit is set to 1, if the rounding mode is RN and the operation result is a subnormal number, the number is flushed to the nearest number. For details, see Section 3.2.6.2(9), Flush to Nearest .	R/W	0																						
22	IF	This bit accumulates and indicates information about the flushing of input operands. For details about flushing subnormal numbers, see Section 3.2.6.2(8), Flushing Subnormal Numbers .	R/W	0																						
21	—	(Reserved for future expansion. Be sure to set to 1.)	R	1																						
20	—	(Reserved for future expansion. Be sure to set to 0.)	R	0																						
19, 18	RM	These are the rounding mode control bits. The RM bits define the rounding mode that the FXU uses for all extended floating-point instructions.	R/W	00																						
		<table border="1"> <thead> <tr> <th colspan="2">RM Bits</th> <th rowspan="2">Mnemonic</th> <th rowspan="2">Description</th> </tr> <tr> <th>19</th> <th>18</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>RN</td> <td>Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.</td> </tr> <tr> <td>0</td> <td>1</td> <td>RZ</td> <td>Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.</td> </tr> <tr> <td>1</td> <td>0</td> <td>RP</td> <td>Rounds the result toward $+\infty$. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.</td> </tr> <tr> <td>1</td> <td>1</td> <td>RM</td> <td>Rounds the result toward $-\infty$. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.</td> </tr> </tbody> </table>	RM Bits		Mnemonic	Description	19	18	0	0	RN	Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.	0	1	RZ	Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.	1	0	RP	Rounds the result toward $+\infty$. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.	1	1	RM	Rounds the result toward $-\infty$. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.		
RM Bits		Mnemonic	Description																							
19	18																									
0	0	RN	Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.																							
0	1	RZ	Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.																							
1	0	RP	Rounds the result toward $+\infty$. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.																							
1	1	RM	Rounds the result toward $-\infty$. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.																							
17	FS	This bit enables values that could not be normalized (subnormal numbers) to be flushed. If the FS bit is set, input operands and operation results that are subnormal numbers are flushed without causing an unimplemented operation exception (E). An input operand that is a subnormal number is flushed to 0 with the same sign. Operation results that are subnormal numbers either become 0 or the minimum normalized number, depending on the rounding mode.	R/W	1																						
		<table border="1"> <thead> <tr> <th rowspan="2">Operation Result that is a Subnormal Number</th> <th colspan="4">Rounding Mode and Value after Flushing</th> </tr> <tr> <th>RN*1</th> <th>RZ</th> <th>RP</th> <th>RM</th> </tr> </thead> <tbody> <tr> <td>Positive</td> <td>+0</td> <td>+0</td> <td>$+2^{E_{min}}$</td> <td>+0</td> </tr> <tr> <td>Negative</td> <td>-0</td> <td>-0</td> <td>-0</td> <td>$-2^{E_{min}}$</td> </tr> </tbody> </table>	Operation Result that is a Subnormal Number	Rounding Mode and Value after Flushing				RN*1	RZ	RP	RM	Positive	+0	+0	$+2^{E_{min}}$	+0	Negative	-0	-0	-0	$-2^{E_{min}}$					
Operation Result that is a Subnormal Number	Rounding Mode and Value after Flushing																									
	RN*1	RZ	RP	RM																						
Positive	+0	+0	$+2^{E_{min}}$	+0																						
Negative	-0	-0	-0	$-2^{E_{min}}$																						
<p>Note 1. If the rounding mode is RN and the FXSR.FN bit is set, flushing will occur in the direction of higher accuracy. For details, see Section 3.2.6.2(9), Flush to Nearest.</p>																										

Table 3.55 FXSR Register Contents (2/2)

Bit	Name	Description	R/W	Value after Reset
16	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
15 to 10	XC (E,V,Z,O,U,I)	These are the cause bits. For details, see Section 3.2.3.5.2.a.1, Cause Bits (XC) .	R/W	Undefined
9 to 5	XE (V,Z,O,U,I)	These are the enable bits. For details, see Section 3.2.3.5.2.a.2, Enable Bits (XE) .	R/W	0
4 to 0	XP (V,Z,O,U,I)	These are the preservation bits. For details, see Section 3.2.3.5.2.a.3, Preservation Bits (XP) .	R/W	Undefined

3.2.3.5.2.a.1 Cause Bits (XC)

Bits 15 to 10 in the FXSR register are cause bits, which indicate the occurrence and cause of an extended floating-point operation exception. The FXSR.XC bits hold the OR of the detection results of extended floating-point operation exceptions that occurred in the arithmetic ways. If an exception defined by IEEE754 is generated, when an enable bit is set to 1 corresponding to the exception, a cause bit is set, and the exception then occurs. When two or more exceptions occur during a single instruction, each corresponding bit is set to 1.

If two or more exceptions are detected, as long as the enable bit corresponding to one of the exceptions is set to 1, the exception occurs. In this case, the cause bits of all the detected exceptions, including exceptions whose enable bits are cleared to 0, are set to 1.

The cause bits are rewritten by the extended floating-point operation instruction that caused the extended floating-point operation exception. The E bit is set to 1 when software emulation is required, otherwise it is cleared to 0. Other bits are set to 1 or cleared to 0 depending on whether or not an IEEE754-defined exception has occurred.

When an extended floating-point operation exception has occurred, the operation result is not stored, and only the cause bits are affected.

When the cause bits are set to 1 by an LDSR instruction, no extended floating-point operation exception occurs.

3.2.3.5.2.a.2 Enable Bits (XE)

Bits 9 to 5 in the FXSR register are the enable bits, which enable extended floating-point operation exceptions. When an IEEE754-defined exception occurs, an extended floating-point operation exception occurs if the enable bit corresponding to the exception has been set to 1.

There are no enable bits corresponding to an unimplemented operation exception (E). An unimplemented operation exception (E) always occurs as an extended floating-point operation exception.

The FXSR.XE bits cannot be set on an arithmetic way basis. The same setting is used for the all arithmetic ways.

If the corresponding enable bit has not been set to 1, no exception occurs and the default result defined by IEEE754 is stored.

3.2.3.5.2.a.3 Preservation Bits (XP)

Bits 4 to 0 in the FXSR register are preservation bits and accumulate and indicate the exceptions that have been detected since a reset. An exception defined by IEEE754 occurs, and if an extended floating-point operation exception is not generated, the preservation bit is set to 1, otherwise it does not change. The preservation bits are not cleared to 0 by the extended floating-point operation. However, these bits can be set and cleared by software when an LDSR instruction is used to write a new value to the FXSR register.

There are no preservation bits corresponding to unimplemented operation exceptions (E). An unimplemented operation exception (E) always occurs as an extended floating-point operation exception.

NOTE

For details about the exception types and how they relate to particular bits, see **Figure 3.38, Relationship among the Cause and Enable Bits of the FXSR and FXXC Registers** and **Figure 3.39, Relationship among the Preservation and Enable Bits of the FXSR and FXXP Registers**.

(b) FXST — Extended Floating-point Operation Status

This register reflects the contents of the FXSR register bits related to the operation status.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
FXST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	0	0	E	V	Z	O	U	I	0	0	IF	V	Z	O	U	I	Value after reset Undefined

Table 3.56 FXST Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 14	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
13 to 8	XC (E,V,Z,O,U,I)	These are cause bits. For details, see Section 3.2.3.5.2.a.1, Cause Bits (XC) . Values written to these bits are reflected in FXSR.XC bits.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5	IF	This bit accumulates and indicates information about the flushing of input operands. For details about flushing subnormal numbers, see Section 3.2.6.2 (8), Flushing Subnormal Numbers . Value written to this bit is reflected in FXSR.IF bit.	R/W	0
4 to 0	XP (V,Z,O,U,I)	These are preservation bits. For details, see Section 3.2.3.5.2.a.3, Preservation Bits (XP) . Values written to these bits are reflected in FXSR.XP bits.	R/W	Undefined

(c) FXINFO — FXU Configuration Information

This register holds the information about the configuration of the FXU.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
FXINFO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NFPU	RSIZE	Value after reset 0000 0003 _H

Table 3.57 FXINFO Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	NFPU	This bit indicates that 4-way SIMD arithmetic unit is provided.	R	1
0	RSIZE	This bit indicates that 32 128-bit vector registers are provided.	R	1

(d) FXCFG — Extended Floating-point Operation Configuration

This register reflects the contents of the FXSR register bits related to the operation settings.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FXCFG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	RM	0	0	0	V	Z	O	U	I	Value after reset 0000 0000 _H

Table 3.58 FXCFG Register Contents

Bit	Name	Description	R/W	Value after Reset																						
31 to 10	—	(Reserved for future expansion. Be sure to set to 0.)	R	0																						
9, 8	RM	These are the rounding mode control bits. The RM bits define the rounding mode that the FXU uses for all extended floating-point instructions. Values written to these bits are reflected in the RM bits of the FXSR register.	R/W	0																						
		<table border="1"> <thead> <tr> <th colspan="2">RM Bits</th> <th rowspan="2">Mnemonic</th> <th rowspan="2">Description</th> </tr> <tr> <th>9</th> <th>8</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>RN</td> <td>Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.</td> </tr> <tr> <td>0</td> <td>1</td> <td>RZ</td> <td>Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.</td> </tr> <tr> <td>1</td> <td>0</td> <td>RP</td> <td>Rounds the result toward $+\infty$. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.</td> </tr> <tr> <td>1</td> <td>1</td> <td>RM</td> <td>Rounds the result toward $-\infty$. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.</td> </tr> </tbody> </table>	RM Bits		Mnemonic	Description	9	8	0	0	RN	Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.	0	1	RZ	Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.	1	0	RP	Rounds the result toward $+\infty$. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.	1	1	RM	Rounds the result toward $-\infty$. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.		
RM Bits		Mnemonic	Description																							
9	8																									
0	0	RN	Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.																							
0	1	RZ	Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.																							
1	0	RP	Rounds the result toward $+\infty$. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.																							
1	1	RM	Rounds the result toward $-\infty$. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.																							
7 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0																						
4 to 0	XE (V,Z,O,U,I)	These are the enable bits. For details, see Section 3.2.3.5.2.a.2, Enable Bits (XE) . Values written to these bits are reflected in FXSR.XE bits.	R/W	0																						

(e) FXXC — XC (Cause) Bits for Each Arithmetic Way

This register holds the XC (cause) bits of extended floating-point operation exceptions for each arithmetic way. Any attempts to write this register will not affect the value of the XC bits of the FXSR register.

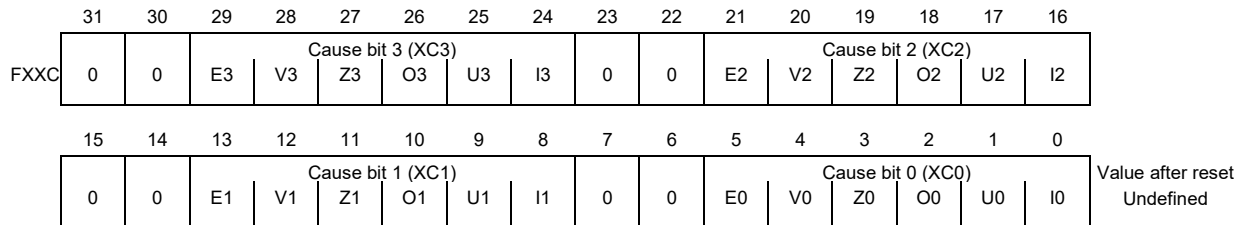


Table 3.59 FXXC Register Contents

Bit	Name	Description	R/W	Value after Reset
31, 30	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
29 to 24	XC3 (E3,V3,Z3,O3,U3,I3)	These bits hold the cause bits associated with arithmetic way 3.	R/W	Undefined
23, 22	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
21 to 16	XC2 (E2,V2,Z2,O2,U2,I2)	These bits hold the cause bits associated with arithmetic way 2.	R/W	Undefined
15, 14	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
13 to 8	XC1 (E1,V1,Z1,O1,U1,I1)	These bits hold the cause bits associated with arithmetic way 1.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5 to 0	XC0 (E0,V0,Z0,O0,U0,I0)	These bits hold the cause bits associated with arithmetic way 0.	R/W	Undefined

(f) FXXP — XP (Preservation) Bits for Each Arithmetic Way

This register holds the XP (preservation) bits of extended floating-point operation exceptions for each arithmetic way. Any attempts to write this register will not affect the value of the XP bits of the FXSR register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
FXXP	0	0	0	V3	Preservation bit 3 (XP3)			I3	0	0	0	V2	Preservation bit 2 (XP2)			I2	
					Z3	O3	U3					Z2	O2	U2			
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	0	0	0	V1	Preservation bit 1 (XP1)			I1	0	0	0	V0	Preservation bit 0 (XP0)			I0	Value after reset Undefined
					Z1	O1	U1					Z0	O0	U0			

Table 3.60 FXXP Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 29	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
28 to 24	XP3 (V3,Z3,O3,U3,I3)	These bits hold the preservation bits associated with arithmetic way 3.	R/W	Undefined
23 to 21	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
20 to 16	XP2 (V2,Z2,O2,U2,I2)	These bits hold the preservation bits associated with arithmetic way 2.	R/W	Undefined
15 to 13	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
12 to 8	XP1 (V1,Z1,O1,U1,I1)	These bits hold the preservation bits associated with arithmetic way 1.	R/W	Undefined
7 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	XP0 (V0,Z0,O0,U0,I0)	These bits hold the preservation bits associated with arithmetic way 0.	R/W	Undefined

3.2.3.6 MPU Function Registers

(1) MPU Function System Registers

MPU function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.61 MPU Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR0, 5	MPM	Memory protection operation mode setting	SV
SR2, 5	MPCFG	MPU configuration	SV
SR8, 5	MCA	Memory protection setting check address	SV
SR9, 5	MCS	Memory protection setting check size	SV
SR10, 5	MCC	Memory protection setting check command	SV
SR11, 5	MCR	Memory protection setting check result	SV
SR12, 5	MCI	Memory protection setting check SPID	SV
SR16, 5	MPIDX	Index of memory protection setting registers to be accessed	SV
SR20, 5	MPLA	Protection area minimum address	SV
SR21, 5	MPUA	Protection area maximum address	SV
SR22, 5	MPAT	Protection area attribute	SV
SR24, 5	MPID0	SPID which can access protection area	SV
SR25, 5	MPID1	SPID which can access protection area	SV
SR26, 5	MPID2	SPID which can access protection area	SV
SR27, 5	MPID3	SPID which can access protection area	SV
SR28, 5	MPID4	SPID which can access protection area	SV
SR29, 5	MPID5	SPID which can access protection area	SV
SR30, 5	MPID6	SPID which can access protection area	SV
SR31, 5	MPID7	SPID which can access protection area	SV

(a) MPM — Memory Protection Operation Mode

The memory protection operation mode register is used to define the basic operation mode of the memory protection function.

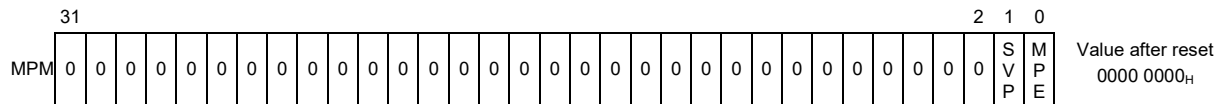


Table 3.62 MPM Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	SVP	In SV mode (when PSW.UM = 0), this bit is used to specify whether to restrict access according to the SX, SW, and SR bits of the MPAT register for each protection area.*1 0: Enable all accesses in SV mode. 1: Restrict access according to the SX, SW, and SR bits in SV mode.*2	R/W	0
0	MPE	This bit is used to specify whether to enable or disable MPU function. 0: Disable 1: Enable	R/W	0

Note 1. When the SVP bit is set to 1 by an LDSR instruction, the setting is effective from the operand access by the subsequent instruction. On the other hand, since the instruction fetch access is executed independently from the setting of the SVP bit, it is not possible to identify the timing when the new setting is applied to the instruction fetch access. In this case, make settings of the protection area in advance so that the program which sets the SVP bit itself will not be blocked.

Note 2. If the SVP bit is set to 1, the MDP or MIP exception handler cannot be executed if it is not permitted to access to the necessary memory areas. Be sure to make settings of the protection area in advance so that the necessary memory areas can be accessed by the exception handler.

(b) MPCFG — MPU Configuration

This register holds the information about the configuration of the MPU.

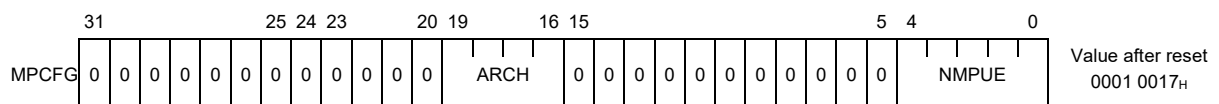


Table 3.63 MPCFG Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 20	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
19 to 16	ARCH	These bits indicate the version of the MPU architecture specifications. A value of 1 is read for this CPU.	R	1
15 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	NMPUE	These bits indicate the number of MPU entries implemented in this CPU minus 1. A value of 23 is read since this CPU incorporates 24 MPU entries.	R	23

(c) MCA — Memory Protection Setting Check Address

This register is used to specify the base address of the area for which a memory protection setting check is to be performed.

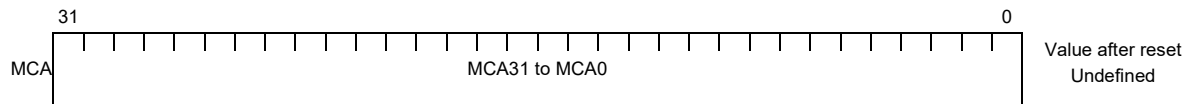


Table 3.64 MCA Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 0	MCA31 to MCA0	These bits are used to specify the starting address of the memory area which subjects to a memory protection setting check in bytes.	R/W	Undefined

(d) MCS — Memory Protection Setting Check Size

This register is used to specify the size of the area for which a memory protection setting check is to be performed.

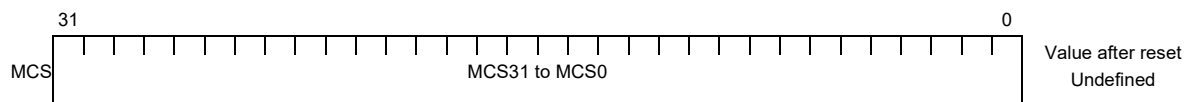


Table 3.65 MCS Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 0	MCS31 to MCS0	These bits are used to specify the size of the memory area in bytes which subjects to a memory protection setting check. Because the specified size is assumed to represent an unsigned integer, it is not possible to check an area in the direction in which the address value decreases relative to the MCA register value.	R/W	Undefined

(e) MCC — Memory Protection Setting Check Command

This command register is used to start a memory protection setting check.

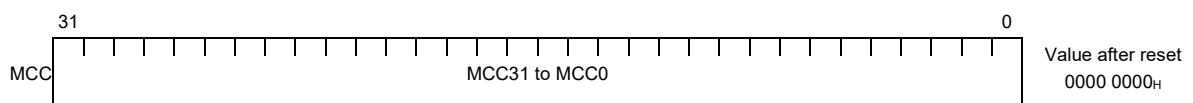


Table 3.66 MCC Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 0	MCC31 to MCC0	When any value is written to the MCC register, a memory protection setting check starts. By setting up the MCA / MCS register and then writing to the MCC register, results are stored in MCR. Because the check is started by any written value, a check can be started by using r0 as the source register without using any unnecessary registers. Note that, for the check, the results are applied according to each area setting regardless of the state of the PSW.UM bit. When the MCC register is read, value 0000 0000 _H is always returned.	R/W	0

(f) MCR — Memory Protection Setting Check Result

This register is used to store the results of a memory protection setting check.

When the MCC register is written, the value of the MPIDn registers (n = 0 to 7), the WMPIDn bits (n = 0 to 7) (for write permission) and the RMPIDn bits (n = 0 to 7) (for execution and read permission) of each MPAT register are checked to verify that the SPID specified by the MCI register matches the SPID for which the access is permitted.

However, if the WG bit or RG bit of the MPAT register is set to 1, it is assumed that the SPID specified by the MCI register matches the SPID for which write or execution and read access is permitted regardless of the SPID matching result above.

If an SPID match is found, protection attributes of the protection area which includes the address area specified by the MCA and MCS registers are stored in the corresponding attribute bits of the MCR register. If multiple protection areas include the specified address area, and if an access is permitted by any of the protection areas, 1, which indicates the access is permitted, is stored in the corresponding attribute bits of the MCR register.

If no SPID match is found or no protection area including the specified address area exists, 0, which indicates the access is not permitted, is stored in each attribute bit of the MCR register.

CAUTION

If the specified area to be checked crosses 0000 0000_H or 7FFF FFFF_H, it is judged as an area setting error, and the MCR.OV bit is set to 1. This means that the MCR.OV bit must be read to confirm that the result is not invalid (OV = 0) before referencing the check result of each attribute bit. When the MCR.OV bit is set to 1, all attribute bits are cleared to 0 unless the MPM.SVP bit is 0. If the MPM.SVP bit is 0, the MCR.SXE, SWE and SRE bits are set to 1, even if the MCR.OV bit is set to 1. Even when no SPID match is found, the MCR.OV bit is set to 1 if the area setting is wrong.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	OV	0	0	SXE	SWE	SRE	UXE	UWE	URE

Value after reset
Undefined

Table 3.67 MCR Register Contents (1/2)

Bit	Name	Description	R/W	Value after Reset
31 to 9	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
8	OV	If the specified area includes 0000 0000 _H or 7FFF FFFF _H , 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5	SXE	If the specified area is contained within one protection area and execution is permitted for that area in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
4	SWE	If the specified area is contained within one protection area and writing to that area is permitted in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
3	SRE	If the specified area is contained within one protection area and reading from that area is permitted in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
2	UXE	If the specified area is contained within one protection area and execution is permitted for that area in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined

Table 3.67 MCR Register Contents (2/2)

Bit	Name	Description	R/W	Value after Reset
1	UWE	If the specified area is contained within one protection area and reading from that area is permitted in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
0	URE	If the specified area is contained within one protection area and reading from that area is permitted in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined

(g) MCI — Memory Protection Setting Check SPID

This register is used to specify the SPID for which a memory protection settings check is to be performed.

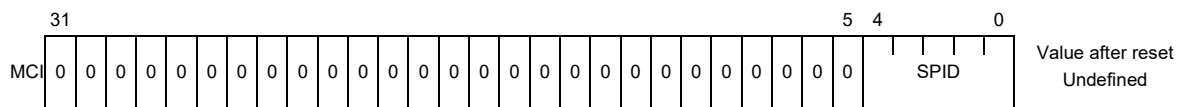


Table 3.68 MCI Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	SPID	These bits specify the SPID for which a memory protection settings check is to be performed.	R/W	Undefined

(h) MPIDX — Index of Memory Protection Setting Registers to be Accessed

This register is used to specify the index of memory protection setting registers to be accessed.

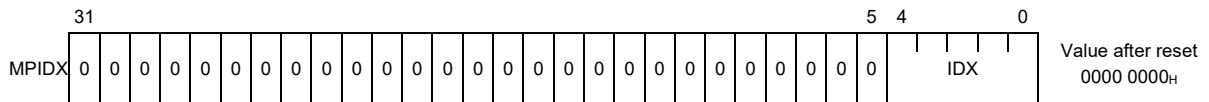


Table 3.69 MPIDX Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	IDX	These bits specify the index of the MPLA, MPUA, and MPAT registers to be accessed by the LDSR or STSR instruction. The result of changing the value of this register is reflected in the instruction immediately following. This CPU provides 24 channels of protection areas, so that IDX may be set to a value from 0 to 23. If a value from 24 to 31 is set to IDX and the MPLA, MPUA, or MPAT register is accessed by the LDSR or STSR instruction, it is handled as an undefined register.	R/W	0

(i) MPLA — Protection Area Minimum Address

This register indicates the minimum address of a protection area. The value written into this register with the LDSR instruction is set as the minimum address of area n ($n = 0$ to 23) specified in the MPIDX register. The value read from this register with the STSR instruction is the minimum address of area n specified in the MPIDX register. It is impossible to manipulate the minimum address of area n without using the MPIDX register and this register.

This CPU provides 24 channels of protection areas, so that the MPLA register can be used specifying a value from 0 to 23 in the MPIDX register. If a value from 24 to 31 is specified in the MPIDX register, the register is handled as an undefined register.



Table 3.70 MPLA Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 2	MPLA31 to MPLA2	These bits specify the minimum address of area n ($n = 0$ to 23) specified by the MPIDX register. The specified minimum address is included in the range of area matching.	R/W	Undefined
1, 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(j) MPUA — Protection Area Maximum Address

This register indicates the maximum address of a protection area. The value written into this register with the LDSR instruction is set as the maximum address of area n ($n = 0$ to 23) specified in the MPIDX register. The value read from this register with the STSR instruction is the maximum address of area n specified in the MPIDX register. It is impossible to manipulate the maximum address of area n without using the MPIDX register and this register.

This CPU provides 24 channels of protection areas, so that the MPUA register can be used specifying a value from 0 to 23 in the MPIDX register. If a value from 24 to 31 is specified in the MPIDX register, the register is handled as an undefined register.

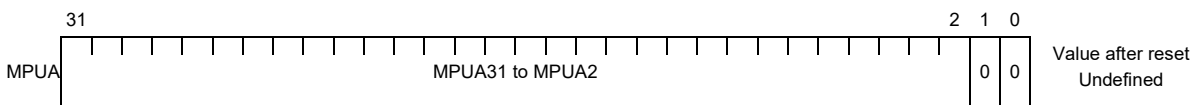


Table 3.71 MPUA Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 2	MPUA31 to MPUA2	These bits specify the maximum address of area n ($n = 0$ to 23) specified by the MPIDX register. The specified maximum address is included in the range of area matching. Bits 1 and 0 of the maximum address are handled as 1.	R/W	Undefined
1, 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(k) MPAT — Protection Area Attribute

This register indicates the attribute of a protection area. The value written into this register with the LDSR instruction is set as the protection attribute of area n (n = 0 to 23) specified in the MPIDX register. The value read from this register with the STSR instruction is the protection attribute of area n specified by the MPIDX register. It is impossible to manipulate the attribute of area n without using the MPIDX register and this register.

This CPU provides 24 channels of protection areas, so that the MPAT register can be used specifying a value from 0 to 23 in the MPIDX register. When a value from 24 to 31 is specified in the MPIDX register, the register is handled as an undefined register.

For details on how to set these bits, see **Section 3.2.5.1(2), Protection Area Settings**.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13		8	7	6	5	4	3	2	1	0					
MPAT	WMPID7	WMPID6	WMPID5	WMPID4	WMPID3	WMPID2	WMPID1	WMPID0	RMPID7	RMPID6	RMPID5	RMPID4	RMPID3	RMPID2	RMPID1	RMPID0	WG	RG	0	0	0	0	0	0	0	E	0	SX	SW	SR	UX	UW	UR	Value after reset Undefined

Table 3.72 MPAT Register Contents (1/2)

Bit	Name	Description	R/W	Value after Reset
31 to 24	WMPID7 to WMPID0	This bit indicates the write permission for the SPID specified in the MPIDn (n = 0 to 7) register. 0: Writing by the SPID specified by the MPIDn register is disabled. 1: Writing by the SPID specified by the MPIDn register is enabled.	R/W	Undefined
23 to 16	RMPID7 to RMPID0	This bit indicates the execution and read permissions for the SPID specified in the MPIDn (n = 0 to 7) register. 0: Execution and reading by the SPID specified by the MPIDn register is disabled. 1: Execution and reading by the SPID specified by the MPIDn register is enabled.	R/W	Undefined
15	WG	This bit indicates the write permission for any SPID. 0: Writing is enabled by the settings of the MPIDn register and WMPIDn bit (n = 0 to 7). 1: Writing by any SPID is enabled regardless of the settings of the MPIDn register and WMPIDn bit (n = 0 to 7).	R/W	Undefined
14	RG	This bit indicates the execution and read permissions for any SPID. 0: Execution and reading is enabled by the settings of the MPIDn register and RMPIDn bit (n = 0 to 7). 1: Execution and reading by any SPID is enabled regardless of the settings of the MPIDn register and RMPIDn bit (n = 0 to 7).	R/W	Undefined
13 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	E	This bit indicates whether area n (n = 0 to 23) specified by the MPIDX register is enabled or disabled. 0: Area n is disabled. 1: Area n is enabled.	R/W	0
6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5	SX	This bit indicates the execution permission in the supervisor mode for area n (n = 0 to 23) specified by the MPIDX register.*1 0: Execution is disabled. 1: Execution is enabled.	R/W	Undefined
4	SW	This bit indicates the write permission in the supervisor mode for area n (n = 0 to 23) specified by the MPIDX register.*1 0: Writing is disabled. 1: Writing is enabled.	R/W	Undefined

Table 3.72 MPAT Register Contents (2/2)

Bit	Name	Description	R/W	Value after Reset
3	SR	This bit indicates the read permission in the supervisor mode for area n (n = 0 to 23) specified by the MPIDX register.*1,*2 0: Reading is disabled. 1: Reading is enabled.	R/W	Undefined
2	UX	This bit indicates the execution permission in the user mode for area n (n = 0 to 23) specified by the MPIDX register. 0: Execution is disabled. 1: Execution is enabled.	R/W	Undefined
1	UW	This bit indicates the write permission in the user mode for area n (n = 0 to 23) specified by the MPIDX register. 0: Writing is disabled. 1: Writing is enabled.	R/W	Undefined
0	UR	This bit indicates the read permission in the user mode for area n (n = 0 to 23) specified by the MPIDX register.*2 0: Reading is disabled. 1: Reading is enabled.	R/W	Undefined

Note 1. If access is restricted in SV mode, the MDP or MIP exception handler might not be executed depending on the settings. Be sure to make settings of the protection area in advance so that the necessary memory areas can be accessed by the exception handler.

Note 2. For the CALLT, SWITCH, and SYSCALL instructions and interrupts in table reference method, an MDP exception occurs if reading the address where the table is stored is not permitted. Permission for the SYSCALL instruction and interrupts in table reference method are judged after the operation mode of the CPU is transitioned to the supervisor mode. Therefore, when the MPM.SVP bit is set (1), the SR bit need to be set (1) as well.

(I) MPIDn —SPID which can Access Protection Area

This register specifies the SPID which can access protection area. The accessibility for the specified SPID is determined in conjunction with the WMPIDn and RMPIDn bits (n = 0 to 7) which are specified via the MPAT register.

This CPU has 8 MPID registers.

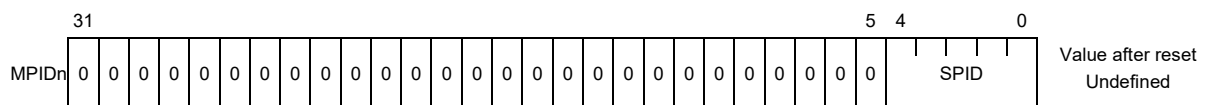


Table 3.73 MPIDn Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	SPID	These bits specify the SPID which can access protection area.	R/W	Undefined

3.2.3.7 Cache Operation Function Registers

(1) Cache Control Function System Registers

Cache control function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.74 Cache Control Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR16, 4	ICTAGL	Instruction cache tag Lo access	SV
SR17, 4	ICTAGH	Instruction cache tag Hi access	SV
SR18, 4	ICDATL	Instruction cache data Lo access	SV
SR19, 4	ICDATH	Instruction cache data Hi access	SV
SR24, 4	ICCTRL	Instruction cache control	SV
SR26, 4	ICCFG	Instruction cache configuration	SV
SR28, 4	ICERR	Instruction cache error	SV

(a) ICTAGL — Instruction Cache Tag Lo Access

This register is used by the CIST/CILD instruction in relation to the instruction cache. During execution of the CIST instruction, values that are stored to the tag RAM for the instruction cache are stored. During execution of the CILD instruction, values read from the tag RAM for the instruction cache are stored.

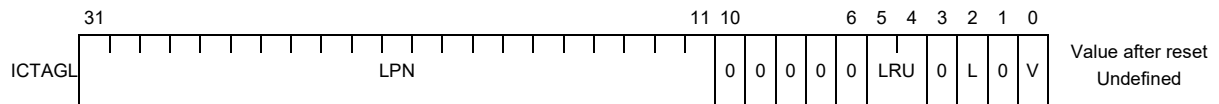


Table 3.75 ICTAGL Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 11	LPN	These bits hold address bits 27 to 11. Bits 31 to 28 must always be set to 0. Bits 27 to 11 are held if the cache size is 8 Kbytes. Bits 27 to 12 are held and bit 11 is always set to 0 if the cache size is 16 Kbytes.*1	R/W	Undefined
10 to 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5, 4	LRU	These bits indicate LRU information of specified cache line. LRU information cannot be freely changed to any value by the CIST instruction.	R/W	Undefined
3	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
2	L	This bit holds the lock information.*2 0: The cache line is not locked. 1: The cache line is locked.	R/W	Undefined
1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	V	This bit holds valid/invalid information of specified cache line.*2 0: The cache line is disabled. 1: The cache line is enabled.	R/W	Undefined

Note 1. Bit 11 cannot be written with any value by an LDSR instruction if the cache size is 16 Kbytes. It is always set to 0.

Note 2. If a tag RAM is set by a CIST instruction, the settings of other bits are valid only when the V bit is enabled. When the V bit is disabled, the cache line is always judged as a cache miss regardless of the value in the LPN bit. Also, when the V bit is disabled, even if the cache line is locked by the setting of the L bit, the cache line becomes the target to be replaced.

(b) ICTAGH — Instruction Cache Tag Hi Access

This register is used by the CIST/CILD instruction in relation to the instruction cache. During execution of the CIST instruction, values that are stored to the tag RAM for the instruction cache are stored. During execution of the CILD instruction, values read from the tag RAM for the instruction cache are stored.

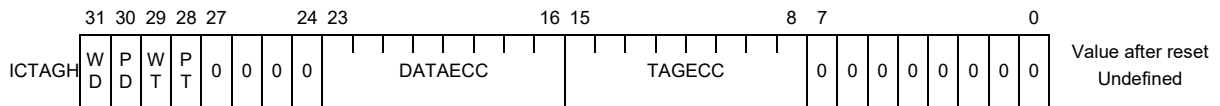


Table 3.76 ICTAGH Register Contents

Bit	Name	Description	R/W	Value after Reset
31	WD	This bit specifies whether the data RAM of the cache is updated during execution of the CIST instruction. 0: Data RAM of the cache is not updated. 1: Data RAM of the cache is updated.	R/W	0
30	PD	This bit specifies the data to be written to the ECC of the data RAM when the WD bit is set to 1 and the CIST instruction is executed. 0: ECC automatically generated from the write data is written to the ECC of the data RAM. 1: Values in the DATAECC field are written to the ECC of the data RAM.	R/W	0
29	WT	This bit specifies whether the tag RAM of the cache is updated during execution of the CIST instruction. When this bit is set to 1, the V bit and L bit of the cache line are also updated. 0: Tag RAM of the cache is not updated. 1: Tag RAM of the cache is updated.	R/W	0
28	PT	This bit specifies the data to be written to the ECC of the tag RAM when the WT bit is set to 1 and the CIST instruction is executed. 0: ECC automatically generated from the write data is written to the ECC of the tag RAM. 1: Values in the TAGECC field are written to the ECC of the tag RAM.	R/W	0
27 to 24	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
23 to 16	DATAECC	These bits hold ECC for data RAM.	R/W	Undefined
15 to 8	TAGECC	These bits hold ECC for tag RAM. Bit 15 is fixed to 0.	R/W	Undefined
7 to 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(c) ICDATL — Instruction Cache Data Lo Access

This register is used by the CIST/CILD instruction in relation to the instruction cache. During execution of the CIST instruction, values that are stored to the data RAM for the instruction cache are stored. During execution of the CILD instruction, values read from the data RAM for the instruction cache are stored.

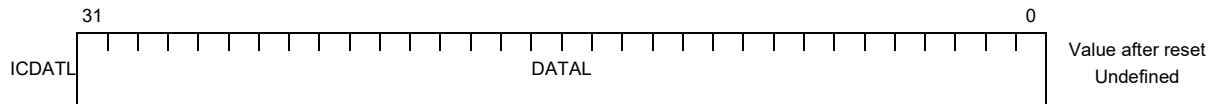


Table 3.77 ICDATL Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 0	DATAL	These bits hold the lower-order 32 bits of the 64-bit instruction data* ¹ in the block in the specified cache line. The bits to be held is specified by offset of the index.* ² Offset of index = 00: Bits 31 to 0* ³ Offset of index = 01: Bits 95 to 64* ³ Offset of index = 10: Bits 159 to 128* ³ Offset of index = 11: Bits 223 to 192* ³	R/W	Undefined

Note 1. Data alignment of the instruction data in the block in the cache line is different from the data alignment of memory. For details, see **Section 3.2.5.2(10), Configuration of Instruction Cache**.

Note 2. For details on how to specify Offset, see **Section 3.2.5.2(6), Cache Index Specification Method**.

Note 3. For details on the bit position, see **Figure 3.31** in **Section 3.2.5.2(10), Configuration of Instruction Cache**.

(d) ICDATH — Instruction Cache Data Hi Access

This register is used by the CIST/CILD instruction in relation to the instruction cache. During execution of the CIST instruction, values that are stored to the data RAM for the instruction cache are stored. During execution of the CILD instruction, values read from the data RAM for the instruction cache are stored.

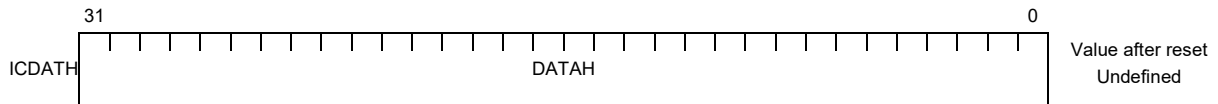


Table 3.78 ICDATH Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 0	DATAH	These bits hold the upper-order 32 bits of the 64-bit instruction data* ¹ in the block in the specified cache line. The bits to be held is specified by offset of the index.* ² Offset of index = 00: Bits 63 to 32* ³ Offset of index = 01: Bits 127 to 96* ³ Offset of index = 10: Bits 191 to 160* ³ Offset of index = 11: Bits 255 to 224* ³	R/W	Undefined

Note 1. Data alignment of the instruction data in the block in the cache line is different from the data alignment of memory. For details, see **Section 3.2.5.2(10), Configuration of Instruction Cache**.

Note 2. For details on how to specify Offset, see **Section 3.2.5.2(6), Cache Index Specification Method**.

Note 3. For details on the bit position, see **Figure 3.31** in **Section 3.2.5.2(10), Configuration of Instruction Cache**.

(e) ICCTRL — Instruction Cache Control

This register is used to control the instruction cache.

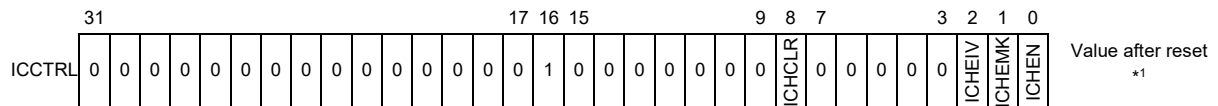


Table 3.79 ICCTRL Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 17	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
16	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
15 to 9	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
8	ICHCLR	When this bit is set to 1, the entire instruction cache is cleared. This clears the V and L bits (to 0) and initializes the LRU information. This bit is always read as 0.	R/W	0
7 to 3	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
2	ICHEIV	When this bit is set to 1, the instruction cache is automatically set as invalid (the ICHEN bit is cleared to 0) whenever a cache error* ² occurs.	R/W	0
1	ICHEMK	When this bit is set to 1, it masks notification of cache error exceptions for the CPU after a cache error* ² has occurred.	R/W	1
0	ICHEN	This bit indicates valid/invalid status of instruction cache. 0: Instruction cache is invalid 1: Instruction cache is valid	R/W	*1

Note 1. The value after reset is determined by the hardware specifications. For details, see **Section 3.8.7, Product Information of Initial Value for G4MH Register**.

Note 2. For details on the cache error, see **Section 3.2.5.2(10), Configuration of Instruction Cache**.

(f) ICCFG — Instruction Cache Configuration

This register indicates the instruction cache configuration.

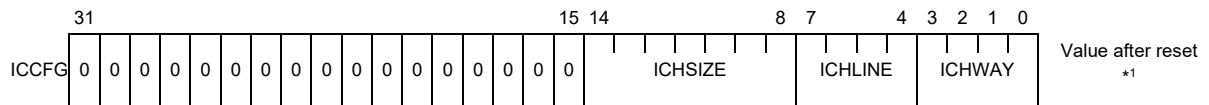


Table 3.80 ICCFG Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 15	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
14 to 8	ICHSIZE	These bits indicate the size (in Kbytes) of the instruction cache. 000 1000: 8 Kbytes 001 0000: 16 Kbytes Other than above: Setting prohibited	R	*1
7 to 4	ICHLINE	These bits indicate the number of lines for each way in the instruction cache. 0010: 64 lines 0100: 128 lines Other than above: Setting prohibited	R	*1
3 to 0	ICHWAY	These bits indicate the number of ways in the instruction cache. 0100: 4 ways Other than above: Setting prohibited	R	*1

Note 1. The value after reset is determined by the hardware specifications. For details, see **Section 3.8.7, Product Information of Initial Value for G4MH Register**.

(g) ICERR — Instruction Cache Error

This register is used to store cache error information for the instruction cache.

After the ICHERR bit is set to 1, any subsequent cache error information that is generated is not stored until this setting is explicitly cleared to 0. However, information on error status (ESAFE, ESMH, ESPBSE, ESTE1, ESTE2, ESDC, and ESDE) is accumulated.

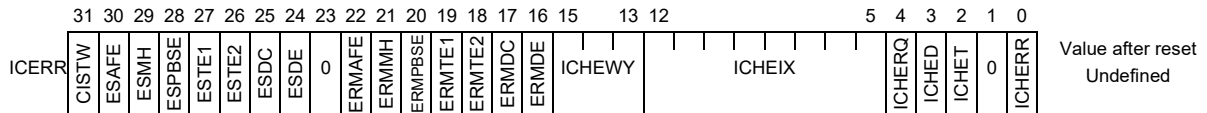


Table 3.81 ICERR Register Contents

Bit	Name	Description	R/W	Value after Reset
31	CISTW	This bit is set to indicate that the destination way specified for a CISTI instruction was in error. Although the entry information is overwritten so that writing is completed, the V bit will be cleared the next time the cache line is read (i.e. reading will be judged to have missed the cache). However, setting of this bit is not accompanied by an exception for the CPU.	R/W	Undefined
30	ESAFE	Error status: Address feedback error	R/W	Undefined
29	ESMH	Error status: Multi hit	R/W	Undefined
28	ESPBSE	Error status: WAY error	R/W	Undefined
27	ESTE1	Error status: Tag RAM 1-bit error	R/W	Undefined
26	ESTE2	Error status: Tag RAM 2-bits error	R/W	Undefined
25	ESDC	Error status: Data RAM 1-bit error	R/W	Undefined
24	ESDE	Error status: Data RAM 2-bits error	R/W	Undefined
23	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
22	ERMAFE	Error exception notification mask: Address feedback error ^{*3}	R/W	0
21	ERM MH	Error exception notification mask: Multi hit ^{*3}	R/W	0
20	ERMPBSE	Error exception notification mask: WAY error ^{*3}	R/W	0
19	ERMTE1	Error exception notification mask: Tag RAM 1-bit error ^{*3}	R/W	0
18	ERMTE2	Error exception notification mask: Tag RAM 2-bits error ^{*3}	R/W	0
17	ERMDC	Error exception notification mask: Data RAM 1-bit error ^{*3}	R/W	0
16	ERMDE	Error exception notification mask: Data RAM 2-bits error ^{*3}	R/W	0
15 to 13	ICHEWY	These bits hold the way number where a cache error occurred. Bit 15 is always set to 0.	R/W	Undefined
12 to 5	ICHEIX	These bits hold the cache index where a cache error occurred. Bits 10 to 5 are held and bits 12 and 11 are fixed to 0 for a cache size of 8 Kbytes.*1 Bits 11 to 5 are held and bit 12 is fixed to 0 for a cache size of 16 Kbytes.*2	R/W	Undefined
4	ICHERQ	When this bit is set to 1, this bit indicates that cache error exception notification is in progress. However, if cache error exception notification has been masked, the CPU is not notified even when 1 has been set to this bit.	R/W	0
3	ICHED	This bit indicates that an error has occurred in data RAM.	R/W	0
2	ICHET	This bit indicates that an error has occurred in tag RAM.	R/W	0
1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	ICHERR	This bit is set to 1 when a cache error has occurred.	R/W	0

Note 1. Bits 12 and 11 cannot be written with any value by an LDSR instruction if the cache size is 8 Kbytes. They are always set to 0.

Note 2. Bit 12 cannot be written with any value by an LDSR instruction if the cache size is 16 Kbytes. It is always set to 0.

- Note 3. While the error exception notification mask bit for each error cause is cleared (to 0), if the corresponding error occurs, the ICHERR, ICHERQ, ICHED/ICHET, ICHEIX, and ICHEWY bits are updated and the corresponding error status bit is set (to 1). While the error exception notification mask bit is set (to 1), if the corresponding error occurs, only the corresponding error status bit is set (to 1).

3.2.3.8 Count Function Registers

(1) Count Function System Registers

Count function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.82 Count Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR0, 11	TSCOUNTL	Timestamp count L register	UM*1
SR1, 11	TSCOUNTH	Timestamp count H register	UM*1
SR2, 11	TSCTRL	Timestamp count control register	SV
SR8, 11	PMUMCTRL	Performance counter user mode control register	SV
SR16, 11	PMCOUNT0	Performance count 0 register	UM*2
SR17, 11	PMCTRL0	Performance count control 0 register	UM*2
SR18, 11	PMCOUNT1	Performance count 1 register	UM*2
SR19, 11	PMCTRL1	Performance count control 1 register	UM*2
SR20, 11	PMCOUNT2	Performance count 2 register	UM*2
SR21, 11	PMCTRL2	Performance count control 2 register	UM*2
SR22, 11	PMCOUNT3	Performance count 3 register	UM*2
SR23, 11	PMCTRL3	Performance count control 3 register	UM*2

Note 1. Only reads are permitted in user mode. A PIE exception will occur if an attempt is made to write in user mode. Reads and writes are permitted in supervisor mode.

Note 2. Accessing in user mode becomes possible by configuring the PMUMCTRL register.

(a) TSCOUNTL — Timestamp Count L

This register, together with the timestamp count H register, implements a 64-bit counter. This register serves as the lower-order 32 bits of the 64-bit counter.

This register can be read only in user mode. A PIE exception will occur if an attempt is made to write this register in user mode. This register can be read and written in supervisor mode.

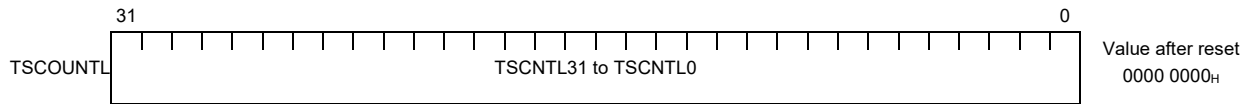


Table 3.83 TSCOUNTL Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 0	TSCNTL31 to TSCNTL0	<p>These bits account for the lower-order 32 bits of the 64-bit counter.</p> <p>If counting is enabled by the TSCTRL register, the value of this register is incremented by 1 on every clock cycle.</p> <p>When this register counts up from FFFF FFFF_H, the TSCOUNTH register is incremented by 1 as a carryover and this register wraps around to 0000 0000_H.</p> <p>This register is accessible at an arbitrary timing regardless of whether counting is enabled or disabled. In addition, it is possible to start counting at an arbitrary count by loading that value in this register before starting to count.</p>	R/W*1	0

Note 1. To use this register for making up an accurate 64-bit counter, see **Section 3.2.2.12, Timestamp Counter**.

(b) TSCOUNTH — Timestamp Count H

This register, together with the timestamp count L register, implements a 64-bit counter. This register serves as the higher-order 32 bits of the 64-bit counter.

This register can be read only in user mode. A PIE exception will occur if an attempt is made to write this register in user mode. This register can be read and written in supervisor mode.

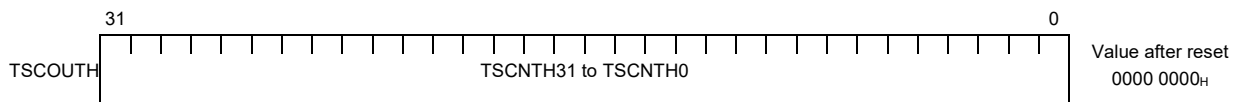


Table 3.84 TSCOUNTH Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 0	TSCNTH31 to TSCNTH0	<p>These bits account for the higher-order 32 bits of the 64-bit counter.</p> <p>If counting is enabled by the TSCTRL register, the value of this register is incremented by 1 when the TSCOUNTL register counts up and generates a carryover.</p> <p>If a carryover occurs in the TSCOUNTL register when this register holds a value of FFFF FFFF_H, the TSCTRL.OVF bit is set to 1 and this register wraps around to 0000 0000_H.</p> <p>This register is accessible at an arbitrary timing regardless of whether counting is enabled or disabled. In addition, it is possible to start counting at an arbitrary count by loading that value in this register before starting to count.</p>	R/W*1	0

Note 1. To use this register for making up an accurate 64-bit counter, see **Section 3.2.2.12, Timestamp Counter**.

(c) TSCTRL — Timestamp Count Control

This register is used to control the 64-bit timestamp counter which is implemented by combining the TSCOUNTH and TSCOUNTL registers.

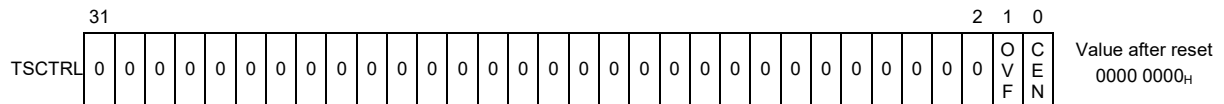


Table 3.85 TSCTRL Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	OVF	This bit serves as the overflow flag. This bit is set to 1 when an overflow occurs in the TSCOUNTH register as the result of its count operation. This bit is not automatically cleared to 0. To reset the overflow state, write a 0 into this bit. Since this bit can be written with a 1, it can be an overflow condition regardless of the count operation. Although it does not affect the counter operation, care must be taken not to take that condition for an overflow.	R/W	0
0	CEN	This bit enables or disables the count operation of the 64-bit timestamp counter which is implemented by combining the TSCOUNTH and TSCOUNTL registers. 0: Disables count operation. 1: Enables count operation. If this bit is written with a 1 when it is set to 0, the counter starts counting immediately. If this bit is written with a 0 when it is set to 1, the counter stops counting immediately. If this bit is 0, the values of the TSCOUNTH and TSCOUNTL registers are preserved. There is no factor that will automatically change the value of this bit.	R/W	0

(d) PMUMCTRL — Performance Counter User Mode Control

This register specifies the accessibility to system registers of the performance measurement function in user mode. The accessibility of the system registers is specified for each of the corresponding channels of the performance measurement function.

When the access in user mode is disabled, any attempt to access one of these system registers generates a PIE exception. When the access in user mode is enabled, the system registers can be read and written.

The performance measurement function itself runs in user mode regardless of the settings of this register. Even when all channels are disabled by this register for accesses in user mode, it is possible to measure the performance in user mode if the settings are done in supervisor mode.

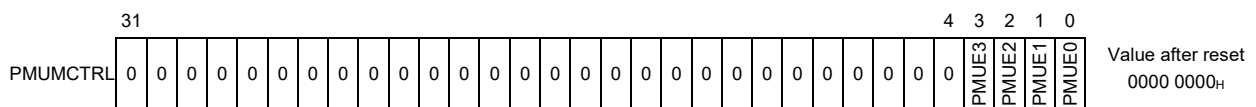


Table 3.86 PMUMCTRL Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 4	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
3	PMUE3	This bit specifies the accessibility of the PMCOUNT3 and PMCTRL3 registers in user mode. 0: Disables accesses to PMCOUNT3 and PMCTRL3 in user mode. 1: Enables accesses to PMCOUNT3 and PMCTRL3 in user mode.	R/W	0
2	PMUE2	This bit specifies the accessibility of the PMCOUNT2 and PMCTRL2 registers in user mode. 0: Disables accesses to PMCOUNT2 and PMCTRL2 in user mode. 1: Enables accesses to PMCOUNT2 and PMCTRL2 in user mode.	R/W	0
1	PMUE1	This bit specifies the accessibility of the PMCOUNT1 and PMCTRL1 registers in user mode. 0: Disables accesses to PMCOUNT1 and PMCTRL1 in user mode. 1: Enables accesses to PMCOUNT1 and PMCTRL1 in user mode.	R/W	0
0	PMUE0	This bit specifies the accessibility of the PMCOUNT0 and PMCTRL0 registers in user mode. 0: Disables accesses to PMCOUNT0 and PMCTRL0 in user mode. 1: Enables accesses to PMCOUNT0 and PMCTRL0 in user mode.	R/W	0

(e) PMCOUNTn — Performance Count

This register counts the number of occurrences of various events, which are specified by the PMCTRLn register. This CPU has 4 channels (n = 0 to 3) of performance count registers.

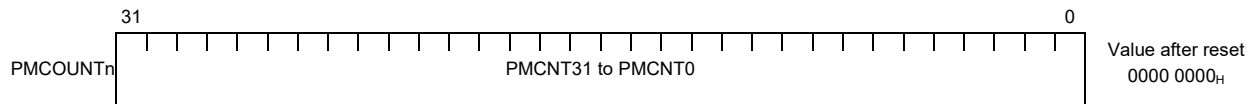


Table 3.87 PMCOUNTn Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 0	PMCNT31 to PMCNT0	<p>These bits form a 32-bit counter register.</p> <p>The counter counts the number of occurrences of the event that is specified by the PMCTRLn.CND bit.</p> <p>When this register counts up from FFFF FFFF_H, the PMCTRLn.OVF bit is set to 1 and the register wraps around to 0000 0000_H.</p> <p>This register is accessible at an arbitrary timing regardless of whether counting is enabled or disabled. In addition, it is possible to start counting at an arbitrary count by loading that value in this register before starting to count.</p>	R/W	0

(f) PMCTRLn — Performance Count Control

This register controls the counting operation of the PMCOUNTn register. This CPU has 4 channels (n = 0 to 3) of performance count control registers.

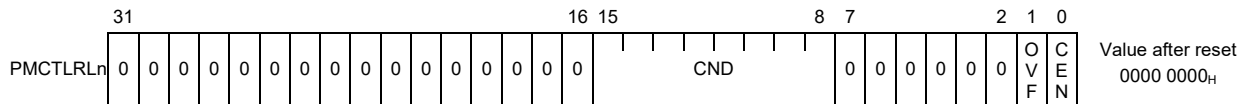


Table 3.88 PMCTRLn Register Contents (1/2)

Bit	Name	Description	R/W	Value after Reset																																																			
31 to 16	—	(Reserved for future expansion. Be sure to set to 0.)	R	0																																																			
15 to 8	CND	These bits specify the number of the event of which occurrences is to be counted. The PMCOUNTn register is not updated if the CEN bit is set (to 1) specifying an invalid event number.	R/W	0																																																			
		<table border="1"> <thead> <tr> <th>Number</th> <th>Details of Event</th> <th>Speculative Operation*1</th> </tr> </thead> <tbody> <tr> <td>00H</td> <td>Number of all clock cycles*2</td> <td>None</td> </tr> <tr> <td>10H</td> <td>Number of executions of all instructions*3</td> <td>None</td> </tr> <tr> <td>18H</td> <td>Number of instructions that cause branch*4</td> <td>None</td> </tr> <tr> <td>19H</td> <td>Number of executions of conditional branch instructions (Bcond/Loop)*5</td> <td>None</td> </tr> <tr> <td>1AH</td> <td>Number of branch prediction misses of conditional branch instructions (Bcond/Loop)*5</td> <td>None</td> </tr> <tr> <td>20H</td> <td>Number of EIINTn acceptances*6</td> <td>None</td> </tr> <tr> <td>21H</td> <td>Number of FEINT acceptances</td> <td>None</td> </tr> <tr> <td>22H</td> <td>Number of terminating type exception acceptances*6 (including EIINTn and FEINT)</td> <td>None</td> </tr> <tr> <td>23H</td> <td>Number of resumable and pending type exception acceptances</td> <td>None</td> </tr> <tr> <td>28H</td> <td>Number of clock cycles during no interrupt is processed (period during which the ISPR register holds 0000H*7)</td> <td>None</td> </tr> <tr> <td>29H</td> <td>Number of clock cycles during no interrupt is processed and interrupts are disabled (period during which the ISPR register holds 0000H*7 and PSW.ID = 1)</td> <td>None</td> </tr> <tr> <td>30H</td> <td>Number of instruction fetch requests*8</td> <td>Yes</td> </tr> <tr> <td>31H</td> <td>Number of instruction cache hits*9</td> <td>Yes</td> </tr> <tr> <td>40H</td> <td>Number of stall cycles during which instructions are not issued to the instruction execution unit*10</td> <td>Yes</td> </tr> <tr> <td>50H</td> <td>Number of instruction fetch requests to Flash ROM*11</td> <td>Yes</td> </tr> <tr> <td>51H</td> <td>Number of data read requests to Flash ROM*12</td> <td>Yes</td> </tr> </tbody> </table>	Number	Details of Event	Speculative Operation*1	00H	Number of all clock cycles*2	None	10H	Number of executions of all instructions*3	None	18H	Number of instructions that cause branch*4	None	19H	Number of executions of conditional branch instructions (Bcond/Loop)*5	None	1AH	Number of branch prediction misses of conditional branch instructions (Bcond/Loop)*5	None	20H	Number of EIINTn acceptances*6	None	21H	Number of FEINT acceptances	None	22H	Number of terminating type exception acceptances*6 (including EIINTn and FEINT)	None	23H	Number of resumable and pending type exception acceptances	None	28H	Number of clock cycles during no interrupt is processed (period during which the ISPR register holds 0000H*7)	None	29H	Number of clock cycles during no interrupt is processed and interrupts are disabled (period during which the ISPR register holds 0000H*7 and PSW.ID = 1)	None	30H	Number of instruction fetch requests*8	Yes	31H	Number of instruction cache hits*9	Yes	40H	Number of stall cycles during which instructions are not issued to the instruction execution unit*10	Yes	50H	Number of instruction fetch requests to Flash ROM*11	Yes	51H	Number of data read requests to Flash ROM*12	Yes		
Number	Details of Event	Speculative Operation*1																																																					
00H	Number of all clock cycles*2	None																																																					
10H	Number of executions of all instructions*3	None																																																					
18H	Number of instructions that cause branch*4	None																																																					
19H	Number of executions of conditional branch instructions (Bcond/Loop)*5	None																																																					
1AH	Number of branch prediction misses of conditional branch instructions (Bcond/Loop)*5	None																																																					
20H	Number of EIINTn acceptances*6	None																																																					
21H	Number of FEINT acceptances	None																																																					
22H	Number of terminating type exception acceptances*6 (including EIINTn and FEINT)	None																																																					
23H	Number of resumable and pending type exception acceptances	None																																																					
28H	Number of clock cycles during no interrupt is processed (period during which the ISPR register holds 0000H*7)	None																																																					
29H	Number of clock cycles during no interrupt is processed and interrupts are disabled (period during which the ISPR register holds 0000H*7 and PSW.ID = 1)	None																																																					
30H	Number of instruction fetch requests*8	Yes																																																					
31H	Number of instruction cache hits*9	Yes																																																					
40H	Number of stall cycles during which instructions are not issued to the instruction execution unit*10	Yes																																																					
50H	Number of instruction fetch requests to Flash ROM*11	Yes																																																					
51H	Number of data read requests to Flash ROM*12	Yes																																																					
7 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0																																																			
1	OVF	This bit serves as the overflow flag. This bit is set to 1 when an overflow occurs in the PMCOUNTn register as the result of its count operation. This bit is not automatically cleared to 0. To reset the overflow state, write a 0 into this bit. Since this bit can be written with a 1, it can be an overflow condition regardless of the count operation. Although it does not affect the counter operation, care must be taken not to take that condition for an overflow.	R/W	0																																																			

Table 3.88 PMCTRLn Register Contents (2/2)

Bit	Name	Description	R/W	Value after Reset
0	CEN	This bit enables or disables the count operation of the PMCOUNTh register. 0: Disables count operation. 1: Enables count operation. If this bit is written with a 1 when it is set to 0, the counter starts counting immediately. If this bit is written with a 0 when it is set to 1, the counter stops counting immediately. If this bit is 0, the values of the PMCOUNTh register is preserved. There is no factor that will automatically change the value of this bit.	R/W	0
Note 1.		For the events for which speculative operation column is "Yes", events whose execution results are cancelled are also counted.		
Note 2.		All clock cycles from the start of counting until the end, including the period during which no instruction is executed, are counted.		
Note 3.		The LDSR instruction which starts counting is not included but the LDSR instruction which stops counting is included in this number.		
Note 4.		Applicable instructions are JR, JMP, Bcond (only if condition is met), JARL, LOOP (only if condition is met), CALLT, DISPOSE, SWITCH, SYSCALL, CTRET, EIRET and FERET instructions. Other instructions with branch processing are not counted. Bcond and LOOP instructions are not counted if condition is not met.		
Note 5.		It is possible to obtain the branch prediction miss rate (hit rate) by measuring the numbers of executions of conditional branch instructions and branch prediction misses. Note that the unconditional branch (BR) of the Bcond instruction is counted in "Number of executions of conditional branch instructions" (CND = 19 _H), but it is not counted in "Number of branch prediction misses of conditional branch instructions" (CND = 1A _H).		
Note 6.		In EIINTn with table reference method, an exception which is detected during reading of the table or automatically saving the context to the register bank is not included in this number.		
Note 7.		Since this event presumes that the ISPR register is automatically updated on the acceptance of and returning from interrupt, it cannot be measured correctly if it is used with the INTCFG.ISPC bit set to 1.		
Note 8.		This CPU acquires the instruction code in 64-bit units. Every 64-bit acquisition is counted as a request for a single instruction fetching. Note that this is not the number of instructions. Instruction fetching by the CACHE instruction is not counted.		
Note 9.		In this CPU, a cache hit indicates the number of requests for a single instruction fetching but not the number of instructions. Cache operation by the CACHE instruction is not counted.		
Note 10.		Instruction issue stall occurs when issuing of an instruction (starting execution of an instruction) is stalled if the CPU satisfies any of the following conditions. <ul style="list-style-type: none"> At execution of an STSR instruction that reads the PSW register, issue of the instruction is stalled until completion of all preceding instructions. At execution of a SYNCP, SYNCM, or SYNCI instruction, issue of the instruction is stalled until waiting conditions for the instruction are cleared. For details on the waiting conditions, see Section 3.2.7.1, Synchronization Processing. At execution of an LDSR instruction or STSR instruction that manipulates the cache control function system registers, issue of the instruction is stalled until completion of all preceding CACHE instructions. At execution of an LDSR instruction or STSR instruction that manipulates the FPU system registers, issue of the instruction is stalled until completion of all preceding FPU instructions. At execution of an LDSR instruction or STSR instruction that manipulates the FXU system registers, issue of the instruction is stalled until completion of all preceding FXU instructions. If an LDSR instruction updates the system register which guarantees reflection of its update to the subsequent instruction, issue of the instruction is stalled until the update of the system register is completed. For details, see Section 3.2.7.3, Hazard Management after System Register Update. If execution of an STSR instruction is started after execution of an LDSR instruction is started, issue of the STSR instruction is stalled until completion of the LDSR instruction. (This applies even if the system register numbers of the LDSR and STSR instructions do not match). When a HALT instruction is executed, issue of the subsequent instruction is stalled until conditions for releasing the halt state are generated. When a SNOOZE instruction is executed, issue of the subsequent instruction is stalled until conditions for releasing the snooze state are generated. When any of the following instructions that includes branching is executed, issue of the subsequent instruction is stalled until completion of the branching and cancellation of execution of the subsequent instruction. SYSCALL, CALLT, EIRET, FERET, CTRET, TRAP, FETRAP, SYNCI 		

- When any of the following exceptions is detected, issue of the subsequent instruction is stalled until acknowledgement of the exception is completed and cancellation of execution of the subsequent instruction.
MIP, SYSERR (resumable), RIE, UCPOP, PIE

- Note 11. Actual Flash ROM access requests are counted. If an instruction fetch request hits the instruction cache, it is not counted. Instruction fetch requests to Flash ROM by the CACHE and PREF instructions are counted. Instruction pre-fetching to Flash ROM is also counted.
- Note 12. Actual Flash ROM access requests are counted. If a data read request hits the ROM data buffer, it is not counted. Pre-fetching to Flash ROM is counted. Note that the number of data read requests to Flash ROM is different from the number of Flash ROM accesses by instructions.
- Note 13. Since the number of instruction fetch request (30_H) and the number of instruction cache hits (31_H) include the speculative instruction fetching, the number of hits divided by the number of requests may not be the accurate instruction cache hit rate of this CPU. The instruction cache hit rate of this CPU can be calculated by the number of instruction fetch requests and the number of instruction fetch requests to Flash ROM (50_H). In this case, the number of instruction fetch request reduced by the number of instruction fetch requests to Flash ROM will be the number of instruction cache hits. However, even in this case, the number of instruction cache hits includes the speculative instruction fetching. Pre-fetching by the PREF instruction is counted as the number of instruction fetch requests, though accessing non-cacheable area is not counted as the number of instruction fetch requests to Flash ROM.

3.2.3.9 Hardware Function Registers

(1) Hardware Function System Registers

Hardware function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.89 Hardware Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR 0, 12	LSTEST0	Lock-step function self-diagnosis register 0	SV
SR1, 12	LSTEST1	Lock-step function self-diagnosis register 1	SV
SR5, 12	IFCR	Instruction fetch control register	SV
SR8, 12	BRPCTRL0	Branch prediction function control register	SV
SR24, 13	RDBCR	ROM data buffer control register	SV

(a) LSTEST0 — Lock-step Function Self-diagnosis Register 0

This register is for the self-diagnosis of the lock-step function. It can be used for the self-diagnosis of the lock-step function in combination with the LSTEST1 register. For details of the lock-step function and its availability, see **Section 38, Functional Safety**.

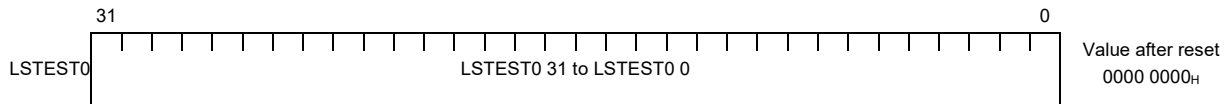


Table 3.90 LSTEST0 Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 0	LSTEST0 31 to LSTEST0 0	[Write] Write any value to bits of the LSTEST0 register. [Read] Value in LSTEST0 is read by the master CPU of the lock-stepped CPU. Value in LSTEST1 is read by the checker CPU of the lock-stepped CPU.	R/W	0

(b) LSTEST1 — Lock-step Function Self-diagnosis Register 1

This register is for the self-diagnosis of the lock-step function. It can be used for the self-diagnosis of the lock-step function in combination with the LSTEST1 register. For details of the lock-step function and its availability, see **Section 38, Functional Safety**.

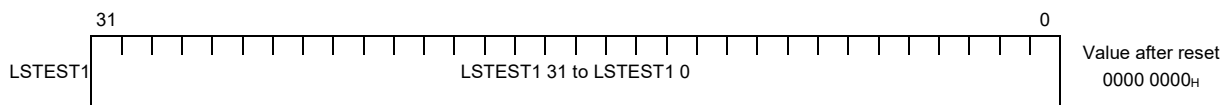


Table 3.91 LSTEST1 Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 0	LSTEST1 31 to LSTEST1 0	[Write] Write any value to bits of the LSTEST1 register. [Read] Value in LSTEST1 is read by the master CPU of the lock-stepped CPU. Value in LSTEST0 is read by the checker CPU of the lock-stepped CPU.	R/W	0

(c) IFCR — Instruction Fetch Control Register

This register controls the instruction fetching function.

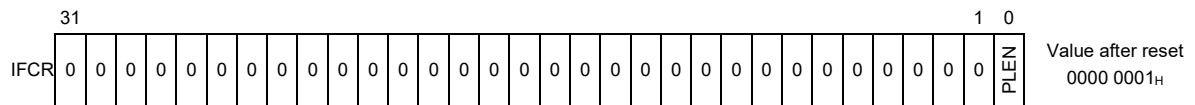


Table 3.92 IFCR Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	PLEN	This bit specifies whether the instruction fetch preload function is enabled or disabled. 0: Disabled 1: Enabled*1	R/W*2	1

Note 1. Instruction fetch preload is executed only when the instruction cache is enabled and the fetching is to the cacheable area.

Note 2. Instruction fetch access and a system register access are independent from each other. Therefore, the settings of this register can be changed even the preload function is running. In this case, the preload function currently running normally completes with the old settings and the new settings are applied from the next operations of the preload function. If you want to suppress preloading before, during, and after changing of the register settings, we recommend execution of instructions by which the register settings are changed in the non-cacheable area.

(d) BRPCTRL0 — Branch Prediction Function Control Register

This register controls the branch prediction function.

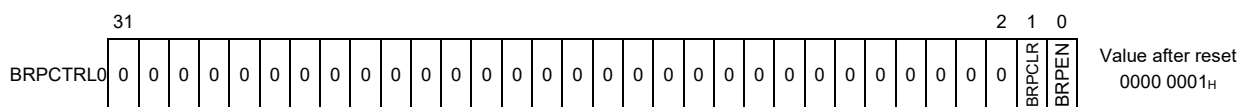


Table 3.93 BRPCTRL0 Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	BRPCLR	Setting this bit to 1 clears the buffers related to branch prediction function. This bit is always read as 0.	R/W*2	0
0	BRPEN	This bit specifies whether the branch prediction function*1 is enabled or disabled. 0: Disabled 1: Enabled	R/W*2	1

Note 1. When this function is enabled, the address of the branch destination of the branch instruction is predicted based on the prediction function incorporated in the product, and instructions are fetched from the predicted address. If this function is disabled, branch prediction is not performed and instructions are fetched from the next instruction address. Branch prediction is performed only when the instruction cache is enabled and fetching is to the cacheable area.

Note 2. Branch prediction and a system register access are independent from each other. Therefore, the settings of this register can be changed even the branch prediction function is running. In this case, the branch prediction function currently running normally completes with the old settings and the new settings are applied from the next operations of the branch prediction function. If you want to suppress branch prediction before, during, and after changing of the register settings, we recommend execution of instructions by which the register settings are changed in the non-cacheable area.

(e) RDBCR — ROM Data Buffer Control Register

This register controls the ROM data buffer.

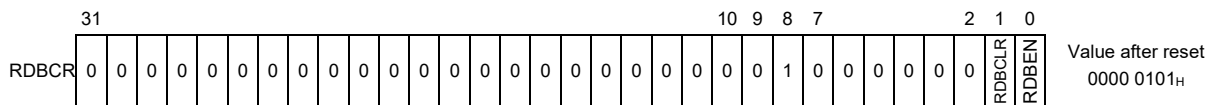


Table 3.94 RDBCR Register Contents

Bit	Name	Description	R/W	Value after Reset
31 to 9	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
8	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
7 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	RDBCLR	Setting this bit (to 1) clears the ROM data buffer and initializes the LRU information in the ROM data buffer. If this bit is set (to 1) during load processing or when a bus request issued by the prefetch function is present, the load data is not registered in the ROM data buffer. This bit is always read as 0.	R/W*1	0
0	RDBEN	This bit specifies whether the ROM data buffer is enabled or disabled. 0: ROM data buffer is disabled. 1: ROM data buffer is enabled.	R/W*1	1

Note 1. Any change in the value of this register is processed independently of any load instructions that use the ROM data buffer. To have the change in the value of this register exactly reflected, it is recommended that this register be manipulated after all load operations are completed.

3.2.4 Exceptions and Interrupts

An exception is a particular event that forces branching of operation from the current program to another program.

The program at the branch destination of a given exception is called an “exception handler”.

CAUTION

This CPU handles interrupts as types of exception.

3.2.4.1 Outline of Exceptions

This section describes the elements that assign properties to exceptions, and shows how exceptions work.

(1) Exception Cause List

This CPU supports the following exceptions.

Table 3.95 Exception Cause List

Exception	Name	Source	Type*1	Saved Resource	Return/Restoration	Exception Cause Code*2	Priority Level	Priority Order*3		Acknowledgment Condition (PSW)		Update (PSW)			
								Priority	ID	NP	UM	ID	NP	EP	EBV
RESET	Reset	Reset input	Terminating	—	—	None	1	—	x	0	1	0	0	0	0
MDP	Memory protection exception (access privilege)	Memory protection violation due to an interrupt using the table reference method	Terminating	FE	Yes	95H	2*7	—	x	0	1	1	1	1	s
FENMI	FENMI interrupt	Interrupt input terminal	Terminating	FE	No	E0H	4	1	x	0	1	1	1	0	s
FEINT	FEINT interrupt	Interrupt input terminal	Terminating	FE	Yes	F0H-FFH	4	2	x	0	1	1	1	0	s
SYSERR	System error	Error due to context saving to the register bank	Terminating	FE	No	1CH	4	3	x	0	1	1	1	1	s
EIINT0-2047	User interrupt	Interrupt input terminal	Terminating	EI	Yes	1000H-17FFH*5	5	*10	0	0	*9	s	0	0	s
MIP	Memory protection exception (execution privilege)	Memory protection violation due to instruction fetching	Resumable	FE	Yes	90H	7	1	x	0	1	1	1	1	s
SYSERR	System error	Error due to instruction fetching	Resumable	FE	No	10H-1FH*4	7	2	x	0	1	1	1	1	s
UCPOP	Coprocessor unusable exception	Execution of a coprocessor instruction/access permission violation	Resumable	FE	Yes	80H-82H*6	7	3	x	0	1	1	1	1	s
RIE	Reserved instruction exception	Execution of a reserved instruction	Resumable	FE	Yes	60H	7	4	x	0	1	1	1	1	s
PIE	Privilege instruction exception	Execution of a privileged instruction/access permission violation	Resumable	FE	Yes	A0H	7	5	x	0	1	1	1	1	s
SYSERR	System error	Error prior to context restoration from the register bank	Resumable	FE	No	1DH	7	6	x	0	1	1	1	1	s
MAE	Misalignment exception	Misaligned access occurrence	Resumable	FE	Yes	C0H	8	1	x	0	1	1	1	1	s
MDP	Memory protection exception (access privilege)	Memory protection violation due to operand access	Resumable	FE	Yes	91H	8	2	x	0	1	1	1	1	s
FPE	FPU exception (precise)	Execution of an FPU instruction	Resumable	EI	Yes	71H	8	3	x	0	1	s	1	1	s
FXE	FXU exception (precise)	Execution of an FXU instruction	Resumable	EI	Yes	75H	8	4	x	0	1	s	1	1	s
SYSCALL	System call	Execution of the SYSCALL instruction	Pending	EI	Yes	8000H-80FFH	9	*8	x	0	1	s	1	1	s
FETRAP	FE level trap	Execution of the FETRAP instruction	Pending	FE	Yes	31H-3FH	9		x	0	1	1	1	1	s
TRAP0	EI level trap 0	Execution of the TRAP instruction	Pending	EI	Yes	40H-4FH	9		x	0	1	s	1	1	s
TRAP1	EI level trap 1	Execution of the TRAP instruction	Pending	EI	Yes	50H-5FH	9		x	0	1	s	1	1	s

Note: s: Retained, x: Not an acknowledgment condition

Note 1. For details, see **Section 3.2.4.1(3), Types of Exceptions**.

Note 2. Represents lower-order 16 bits of the exception cause code. The higher-order 16 bits of the exception cause code are loaded with a detail code which is defined for each exception. The code is 0000H, unless it is specifically described in the individual functional descriptions.

Note 3. The acknowledgment priority for exceptions is checked by the priority level, and then priority. A smaller value has a higher priority. For details, see **Section 3.2.4.1(4), Exception Acknowledgment Conditions and Priority Order**.

Note 4. For details, see **Section 3.2.4.1(2), Overview of Exception Causes**.

- Note 5. 1000_H-17FF_H (channels 0 to 2047) are selected according to the channel.
- Note 6. 80_H-82_H correspond to the coprocessor use permissions (CU0-CU2), respectively.
- Note 7. The case in which an MDP exception occurs during the processing (table read or automatic context saving onto the register bank) which is performed after a table reference method interrupt (EIINT_n) is selected as the result of priority determination. The occurrence of this type of exceptions takes precedence over that of the terminating-type exceptions except the reset. For details, see **Section 3.2.4.1(2), Overview of Exception Causes**.
- Note 8. Occurs in an exclusive manner as it is caused by the execution of an instruction. There is no priority difference within the same priority level.
- Note 9. There are cases in which the PSW.ID bit is set to 0 on interrupts of table reference method in which the register bank is used. For details, see **Section 3.2.4.5(2), Automatic Context Saving**.
- Note 10. The priority of EIINT0-EIINT2047 varies depending on settings of the interrupt controller. For details, see **Section 3.2.4.1(5), Interrupt Exception Priority and Priority Masking**.

(2) Overview of Exception Causes

The following is an overview of the exception causes handled by the CPU.

(a) RESET

RESET is generated when inputting a reset signal. For details, see **Section 3.2.8, Reset**.

(b) FENMI, FEINT, and EIINT

These are interrupts generated by interrupt signals from the interrupt controller to activate a certain program. For details about the interrupt functions, see **Section 3.2.3.3, Interrupt Function Registers** and the specifications of the interrupt controller incorporated in your product.

(c) SYSERR

This is a system error exception.

An error occurring during automatic context saving using the register bank function is notified as a terminating-type SYSERR exception. In this case, as with the case of acknowledging an interrupt, the PC of the instruction that is interrupted when the exception occurred is loaded in the FEPC and the PSW at that time in the FEPSW, respectively. An error occurring during the restoration of the context with the RESBANK instruction is notified as a resumable-type SYSERR exception. In this case, the PC of the RESBANK instruction is loaded in the FEPC and the PSW at that time in the FEPSW, respectively.

An error that occurs at an instruction fetch access is notified as a resumable-type SYSERR exception. In this case, the PC of the instruction to be fetched is loaded in the FEPC and the PSW at that time in the FEPSW, respectively.

Table 3.96 lists the exception cause codes that are loaded in the lower-order 16 bits of the FEIC register when SYSERR exceptions occur. The higher-order 16 bits of the exception cause code are padded with 0s.

Table 3.96 Lower-order 16 Bits of the Exception Cause Codes Associated with the SYSERR Exception

Exception Cause Code	Cause
11 _H	A response error occurred in a bus slave at the time of instruction fetching.* ¹
13 _H	An error within the scope of safety functions, such as an ECC error or parity error, occurred at the time of instruction fetching.* ^{1,*2}
1C _H	An error occurred when automatically saving context to the register bank.
1D _H	An error occurred at the time of context restoration from the register bank (during the execution of the RESBANK instruction).

Note 1. For details, see **Section 3.8.8, Product Information of SYSERR Factor**.

Note 2. When an ECC or parity error is found in fetching from the instruction cache, it is handled as a cache miss so a SYSERR exception does not occur.

All the causes that generate a SYSERR exception for this CPU are listed in **Table 3.96**. An error that is detected externally to the CPU does not generate a SYSERR exception.

(d) FPE

These are exceptions that occur when a floating-point instruction is being executed. For details, see **Section 3.2.6.1, Floating-Point Operation**.

(e) FXE

These are exceptions that occur when an extended floating-point instruction is being executed. For details, see **Section 3.2.6.2, Extended Floating-Point Operation**.

(f) MIP and MDP

These are exceptions that occur when the MPU detects a violation. Detecting an exception is performed when the address at which the instruction will access the memory is calculated. For details, see **Section 3.2.5.1, Memory Protection Unit (MPU)**.

There are cases in which an MDP exception is detected during a table read or automatic context saving onto the register bank when a table reference method interrupt (EIINT n) is selected as the result of determining the priority level of the terminating-type exception. In such a case, the execution of the instruction is interrupted and an MDP exception (terminating-type) is generated as with an ordinary interrupt. In this case, the PC of the instruction that is interrupted is saved in the FEPC and the PSW that is established before the interrupt is accepted is saved in the FEPSW, respectively. For the table reference method, see **Section 3.2.4.4, Exception Handler Address**.

(g) RIE

This is a reserved instruction exception. This exception occurs when an attempt is made to execute the opcode of an instruction other than an instruction whose operation is defined. The operation is the same as the RIE instruction. For details, see *the RH850G4MH User's Manual: Software*.

(h) PIE

This is a privilege instruction exception. This exception occurs when an attempt is made to execute an instruction that does not have the required privilege. For details, see **Section 3.2.2.1(3), CPU Operating Mode and Privileges**, **Section 3.2.2.2, Instruction Execution**, and **Section 3.2.2.5(3)(a), LDSR and STSR**.

(i) UCPOP

This is an exception that occurs when an attempt is made to execute a coprocessor instruction when the coprocessor in question is not usable. For details, see **Section 3.2.2.4, Coprocessors**.

(j) MAE

This is an exception that occurs when the result of address calculation is a misaligned address. For details, see **Section 3.2.2.6(3), Data Alignment**.

(k) TRAP, FETRAP, and SYSCALL

These are exceptions that occur according to the result of instruction execution. For details, see *the RH850G4MH User's Manual: Software*.

(3) Types of Exceptions

This CPU divides exceptions into the following three types according to how they are executed.

- Terminating exceptions
- Resumable exceptions
- Pending exceptions

(a) Terminating Exceptions

In the case of an exception of this type, the exception is acknowledged by interrupting the current instruction before its operation is executed. These exceptions include interrupts, etc.

In the case of interrupts, their occurrence is not dependent upon the result of executing the current instruction, that is, generation is not related to that instruction. When an interrupt occurs, the PSW.EP bit is cleared to 0, unlike other exceptions. Consequently, termination of the exception handler routine is reported to the external interrupt controller when the return instruction is executed. Be sure to execute a return instruction from an interrupt while the PSW.EP bit is cleared to 0.

CAUTIONS

1. The PSW.EP bit is cleared to 0 only when an interrupt (EIINT n , FEINT, or FENMI) is acknowledged. It is set to 1 when any other exception occurs.
If a return instruction from the exception handler routine that has been started by an interrupt is executed while the PSW.EP bit is set to 1, the resources on the external interrupt controller might not be released, causing malfunctioning.
2. A terminating exception may be accepted during the execution of an instruction that performs multiple memory accesses. In this case, although the execution of the instruction is terminated, the result of memory accesses that have been already completed are not canceled. For example, memory is updated by the PREPARE instruction and the general-purpose registers are updated by the DISPOSE instruction. However, it is guaranteed that the PC and SP retain the original values required to re-execute the instruction. For details, see *the RH850G4MH User's Manual: Software*. The relevant instructions are listed below.
 - PREPARE, DISPOSE, PUSHSP, POPSP, RESBANK
3. For the instructions that perform load-access and branch, a terminating exception may be accepted after the load-access is completed. However, it is guaranteed that the PC and the system registers (such as CTPC) that are updated on completion of the instruction retain the original values required to re-execute the instruction. For details, see *the RH850G4MH User's Manual: Software*. The relevant instructions are listed below.
 - CALLT, SYSCALL, SWITCH

(b) Resumable Exceptions

A resumable exception is the one that occurs during the operation of an instruction and that is accepted without the completion of that instruction. It is also called a precise exception because it is accepted precisely without the completion of the instruction that was being executed and without the execution of the subsequent instructions. The occurrence of this exception suppresses the update of the general-purpose and system registers. In addition, since the return PC of the exception points to the very instruction that caused the exception, it is possible to resume the execution at the point before the exception occurred.

The return PC of a resumable exception is the PC of the instruction which caused the exception (current PC).

(c) Pending Exceptions

This is an exception acknowledged after the execution of an instruction finishes as a result of executing the instruction operation. Pending exceptions include software exceptions. Because pending exceptions occur as a result of normal instruction execution, the processing resumes with the instruction following the instruction that caused the pending exceptions when processing control is returned. The original processing can be normally continued after the exception handling.

The return PC of a pending exception is the PC of the next instruction (next PC).

(4) Exception Acknowledgment Conditions and Priority Order

This CPU acknowledges only one exception at specific timing based on the exception acknowledgment conditions and priority order. The exception to be acknowledged is determined based on the exception acknowledgment conditions and priority order, as shown in **Figure 3.16** below.

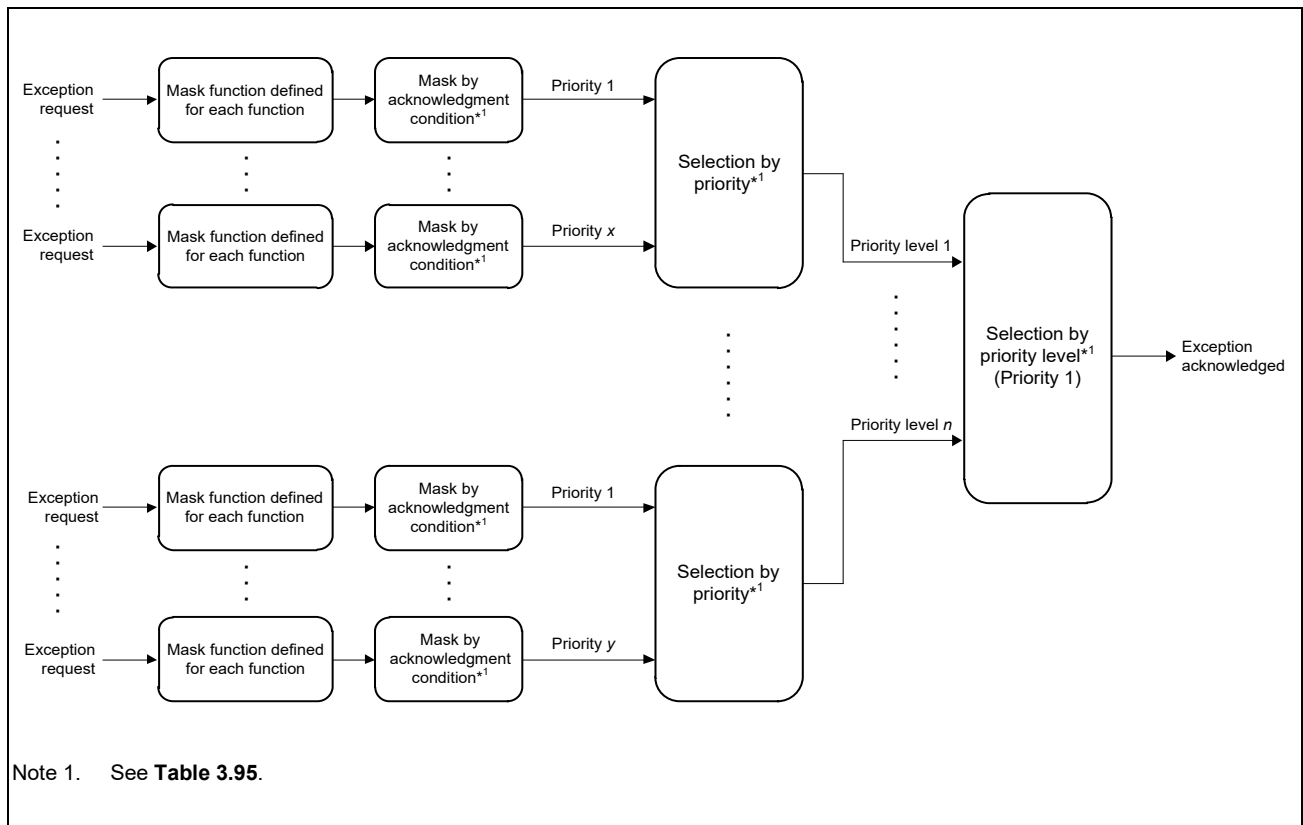


Figure 3.16 Exception Acknowledgment Conditions and Priority Order

In **Table 3.95**, an exception with “0” in the acknowledgment condition column can be acknowledged when the corresponding bit is “0”. This kind of exception is not accepted when the corresponding bit is “1”. When it changes to “0” and the acknowledgment conditions are met, acknowledgment of the exception becomes possible. If no value is specified for a bit, it is not an acknowledgment condition. If multiple bits are specified as conditions, all the conditions must be met simultaneously.

Some exceptions are provided with dedicated mask functions that are unique to their functionality. The decision on the acceptability of the exceptions based on the individually defined mask function is made before the decision on the acknowledgement (acceptance) conditions listed in **Table 3.95**. For details, see the description of the cause of the individual exceptions.

If more than two exceptions satisfy the acknowledgment conditions simultaneously, one exception is selected according to the priority order. The priority order is determined in multiple stages; priority level, and then priority. A smaller number has a higher priority.

When this CPU accepts an exception, it returns an acceptance response to the requesting module.

If a terminating-type exception is not accepted, the request for that exception is not made pending by this CPU. To keep the exception request pending for acceptance, the module such as the interrupt controller that issued the terminating-type exception request to the CPU makes the exception request pending.

Any exceptions that occur at the same time as a reset are not accepted. For details, see **Section 3.2.4.2(1), Special Operations**.

For details about acknowledgment conditions, priority level, and priority, see **Table 3.95**.

(5) Interrupt Exception Priority and Priority Masking

An interrupt (EIINT n) can be masked for each exception priority or interrupt priority by setting registers. This function allows the software implementation of an interrupt ceiling with a more flexible software structure and no maintenance.

Figure 3.17 shows an overview of the functions of interrupt exception priority and priority masking.

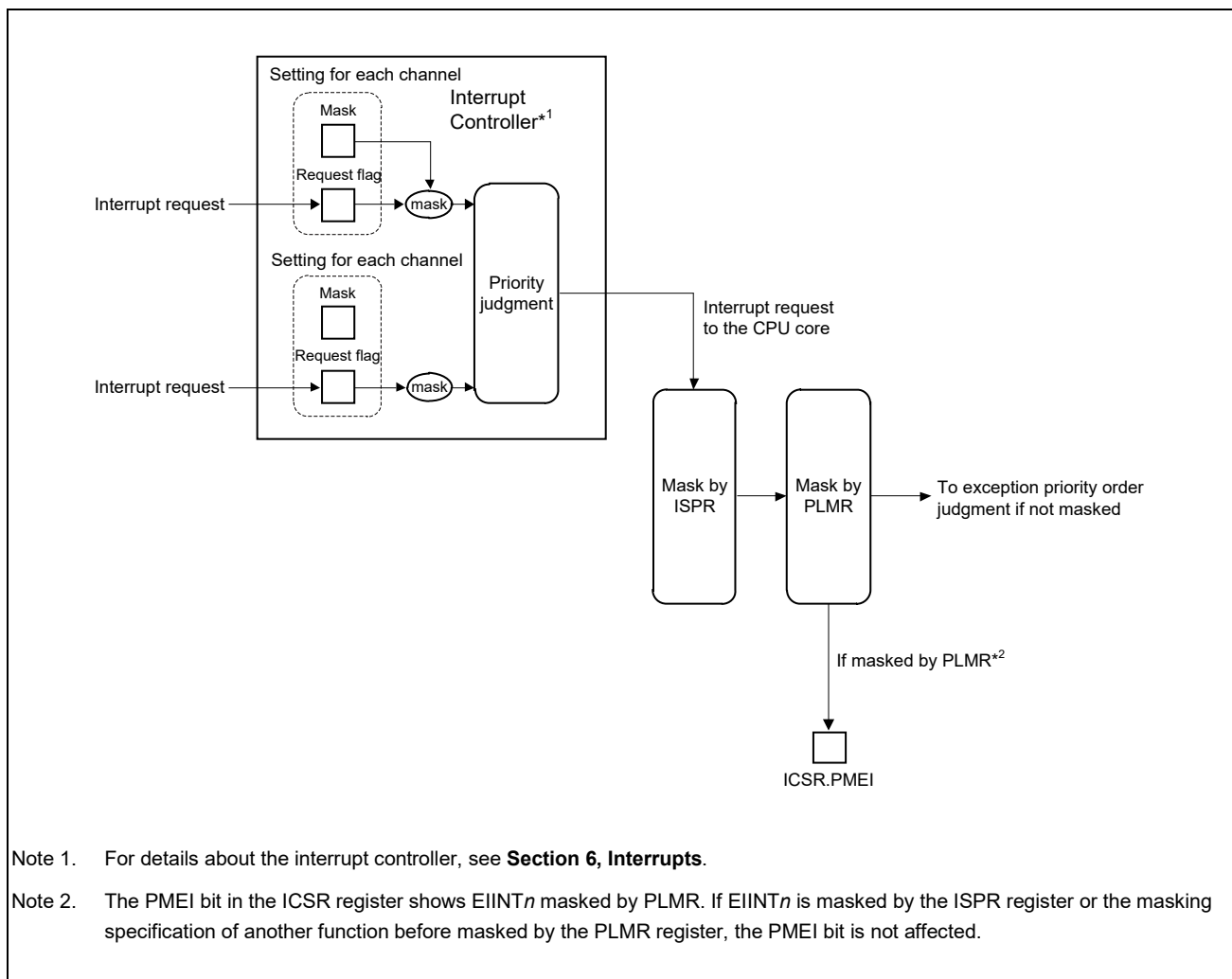


Figure 3.17 Interrupt Exception Priority and Priority Masking

(a) Interrupt Priority

This CPU supports a maximum of 16 priority levels for interrupts (EIINT n). For the details on the procedure to set interrupt priority, see **Section 6, Interrupts**.

(b) Interrupt Priority Mask

EIINT n might be masked at different priorities by the ISPR register and PLMR register. These registers should be used as follows.

For the ISPR register, the bit corresponding to the priority is set to 1 when the hardware acknowledges an interrupt, and interrupts with the same or lower priority are masked. When the EIRET instruction corresponding to the interrupt is executed, the corresponding bit of the ISPR register is cleared to 0 to clear the mask.

This automatic interrupt ceiling makes multiple interrupts servicing easy without using software control.

The PLMR register allows you to mask specific interrupt priorities with software. Use it to raise the priority level of the interrupt ceiling temporarily in a program. The mask setting specified by the ISPR register and the mask setting of PLMR might overlap, and an interrupt is masked if it is masked with one or the other of them. Normally, use the PLMR register to raise the ceiling value from the ceiling value of the ISPR register.

The function of the INTCFG register allows you to disable automatic update of the ISPR register upon acknowledgment of and return from an interrupt. To perform interrupt ceiling control by using software without using the function of the ISPR register, set the ISPC bit in the INTCFG register to 1, clear the ISPR register, and then control the ceiling value with software by using the PLMR register.

Also, when you are using the PLMR register, you can check if any interrupt is masked with the PLMR register by using the ICSR register.

(6) Return and Restoration

When exception handling has been performed, it might affect the original program that was interrupted by the acknowledged exception. This effect is indicated from two perspectives: “Return” and “Restoration”.

- Return: Indicates whether or not the original program can be re-executed from where it was interrupted.
- Restoration: Indicates whether or not the processor statuses (status of processor resources such as general-purpose registers and system registers) can be restored as they were when the original program was interrupted.

An exception that cannot be returned or restored from (“No” in **Table 3.95**) might cause the return PC to be lost, making it impossible to return from the exception to the original processing by using a return instruction. An exception whose trigger cannot be selected is an unreturnable or unrestorable exception.

For an unrestorable exception, it is possible to return to the original program flow. However, because the state before the occurrence of the exception cannot be restored at that point, care must be taken in continuing subsequent program operation.

(7) Context Saving

To save the current program sequence when an exception occurs, appropriately save the following resources according to the function definitions. There are resources that are automatically saved by hardware and resources that need to be saved by software.

- Program counter (PC)
- Program status word (PSW)
- Exception cause code (EIIC, FEIC)
- Work system register (EIWR, FEWR)

The resource to use as the saving destination is determined according to the exception type. Saved resource determination is described below.

For exceptions that use the register bank function, specific resources are automatically saved. For details, see **Section 3.2.4.5, Register Bank Function**.

(a) Context Saving

Exceptions with certain acknowledgment conditions might not be acknowledged at the start of exception handling, based on the pending bits (PSW.ID and NP bits) that are automatically set when another exception is acknowledged.

To enable multiple exception processing which makes exceptions of the same level acceptable again, the return registers and certain information about the corresponding exception causes must be saved, such as to a stack. This information that must be saved is called the “context”.

In principle, it is necessary to make sure that no exceptions of the same level can occur before saving the context.

The working system registers that can be used in the work of saving contexts and those for which the values are saved to enable the handling of multiple exceptions as required are referred to as the basic context registers.

These basic context registers are provided for each exception level. For this reason, it is possible to precisely return from the current exception since its context will not be overwritten when an exception of a different level occurs before saving the current context.

Table 3.97 Basic Context Registers

Exception Level	Basic Context Registers
EI level	EIPC, EIPSW, EIIC, EIWR
FE level	FEPC, FEPSW, FEIC, FEWR

See **Section 3.2.2.3(2), Exception Level**.

3.2.4.2 Operation when Acknowledging an Exception

Check whether each exception that is reported during instruction execution is acknowledged according to the priority. The procedure for exception-specific acknowledgment operation is shown below.

- <1> Check whether the acknowledgment conditions are satisfied and whether exceptions are acknowledged according to their priority.
- <2> Calculate the exception handler address according to the current PSW value*¹
- <3> For FE level exceptions, the following processing is performed.
 - Saving the PC to FEPC
 - Saving the PSW to FEPSW
 - Storing the exception cause code in FEIC
 - Updating the PSW*²
 - Store the exception handler address calculated in (2) in the PC, and then pass control to the exception handler.
- <4> For EI level exceptions, the following processing is performed.
 - Saving the PC to EIPC
 - Saving the PSW to EIPSW
 - Storing the exception cause code in EIIC
 - Updating the PSW*²
 - Store the exception handler address calculated in (2) in the PC, and then pass control to the exception handler.
 - When the exception is an interrupt that uses the register bank, save the context automatically^{Note 3}.

Note 1. For details, see **Section 3.2.4.4, Exception Handler Address**.

Note 2. For the values to be updated, see **Table 3.95**.

Note 3. For details on register banks, see **Section 3.2.4.5, Register Bank Function**.

Figure 3.18 shows the steps <1> to <4>.

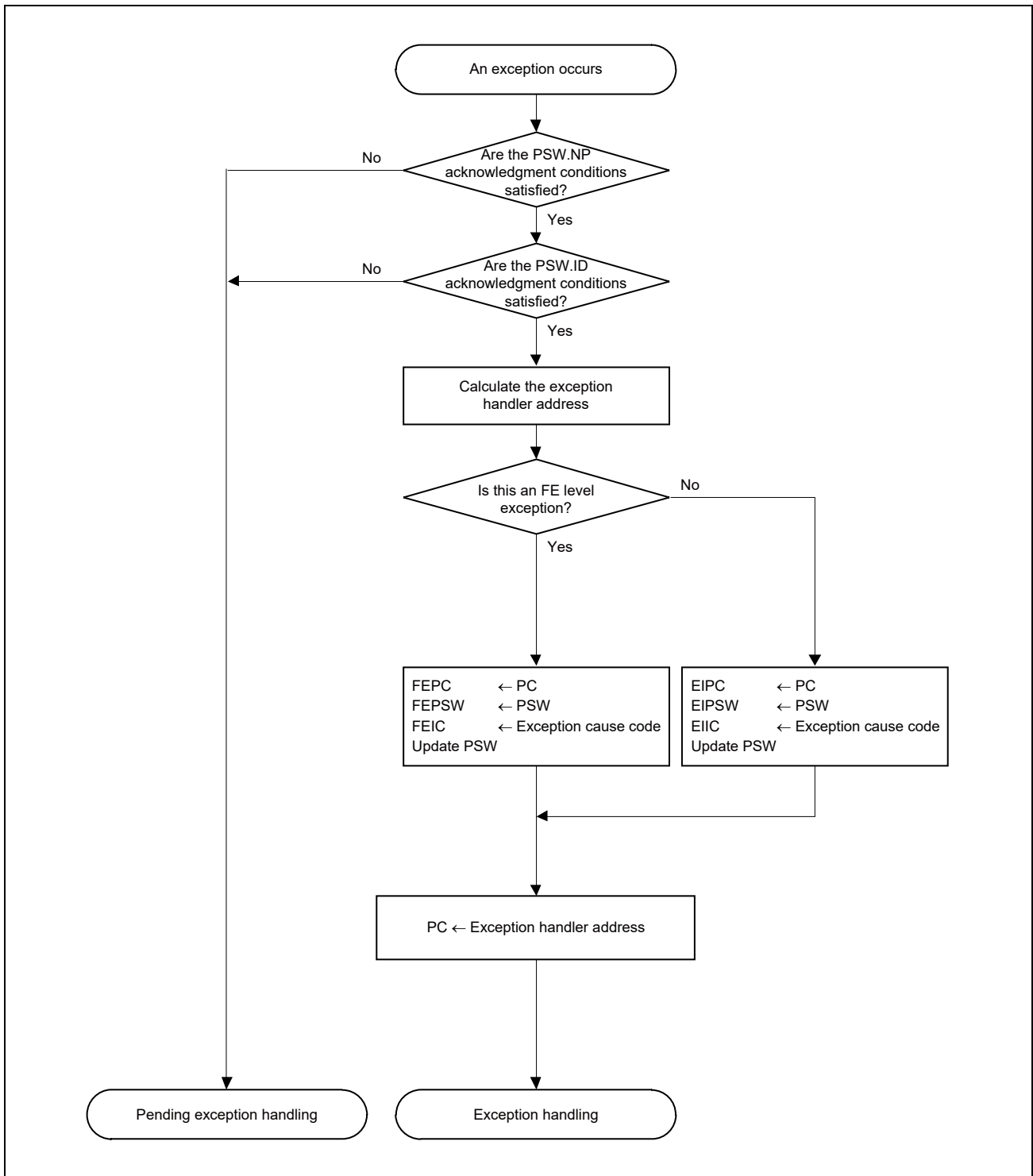


Figure 3.18 Operation When Acknowledging an Exception

(1) Special Operations

(a) EP Bit of PSW Register

If an interrupt is acknowledged, the PSW.EP bit is cleared to 0. If an exception other than an interrupt is acknowledged, the PSW.EP bit is set to 1.

The operation of the CPU when executing the EIRET and FERET instructions depends on the state of the EP bit. If the EP bit is cleared to 0, the CPU notifies the external interrupt controller of the termination of the exception processing routine. This function is necessary to properly control the request flag in the interrupt controller and other resources upon return from the interrupt. When the EP bit is cleared to 0, and the EIRET instruction is executed, the bit with the highest priority (0 is the highest) among the bits set to 1 in ISPR.ISP15 to ISPR.ISP0 is cleared to 0.

To return from an interrupt, be sure to execute the return instruction with the EP bit cleared to 0.

(b) Coprocessor Unusable Exception

For coprocessor unusable exceptions, the opcodes that cause the exception depend on the status of the CU bit of the PSW register.

When a coprocessor is not included in the product or it is not usable, if an attempt is made to execute a coprocessor instruction corresponding to the coprocessor, a coprocessor unusable exception (UCPOP) immediately occurs. If an LDSR or STSR instruction attempts to access a system register of the coprocessor, a coprocessor unusable exception (UCPOP) immediately occurs too.

For details, see **Section 3.2.2.4(3), Coprocessor Unusable Exceptions.**

(c) Reserved Instruction Exception

If an opcode that is reserved for future function extension and for which no instruction is defined is executed, a reserved instruction exception (RIE) occurs.

The opcode that always generates a reserved instruction exception is defined as the RIE instruction.

(d) Reset

Reset is performed in the same way as exception handling, but it is not regarded as EI level exception or FE level exception. The reset operation is the same that of an exception without acknowledgment conditions, but the value of each register is changed to the value after reset. In addition, returning to the original program from the reset is not possible.

All exceptions that have occurred at the same time as CPU initialization are canceled and not acknowledged even after CPU initialization.

For details, see **Section 3.2.8, Reset.**

3.2.4.3 Return from Exception Handling

To return from exception handling, execute the return instruction (EIRET or FERET) corresponding to the relevant exception level.

When a context has been saved, such as to a stack, the context must be restored before executing the return instruction. When execution is returned from an unrestorable exception, the status before the exception occurs in the original program cannot be restored. Consequently, the execution result might differ from that when the exception does not occur.

The EIRET instruction is used to return from EI level exception handling and the FERET instruction is used to return from FE level exception handling.

When the EIRET or FERET instruction is executed, the CPU performs the following processing and then passes control to the return PC address.

- <1> If the PSW.EP bit is set to 0, the CPU notifies the interrupt controller of the termination of the exception routine.
- <2> When the EIRET instruction is executed while PSW.EP = 0 and INTCFG.ISPC = 0, the CPU updates the ISPR register.
When the FERET instruction is executed, the CPU does not update the ISPR register.
- <3> When the EIRET instruction is executed, return PC and PSW are loaded from the EIPC and EIPSW registers. When the FERET instruction is executed, return PC and PSW are loaded from the FEPC and FEPSW registers.
- <4> Control is passed to the address indicated by the return PC that were loaded.

Figure 3.19 shows the flow for returning from exception handling using the EIRET or FERET instruction.

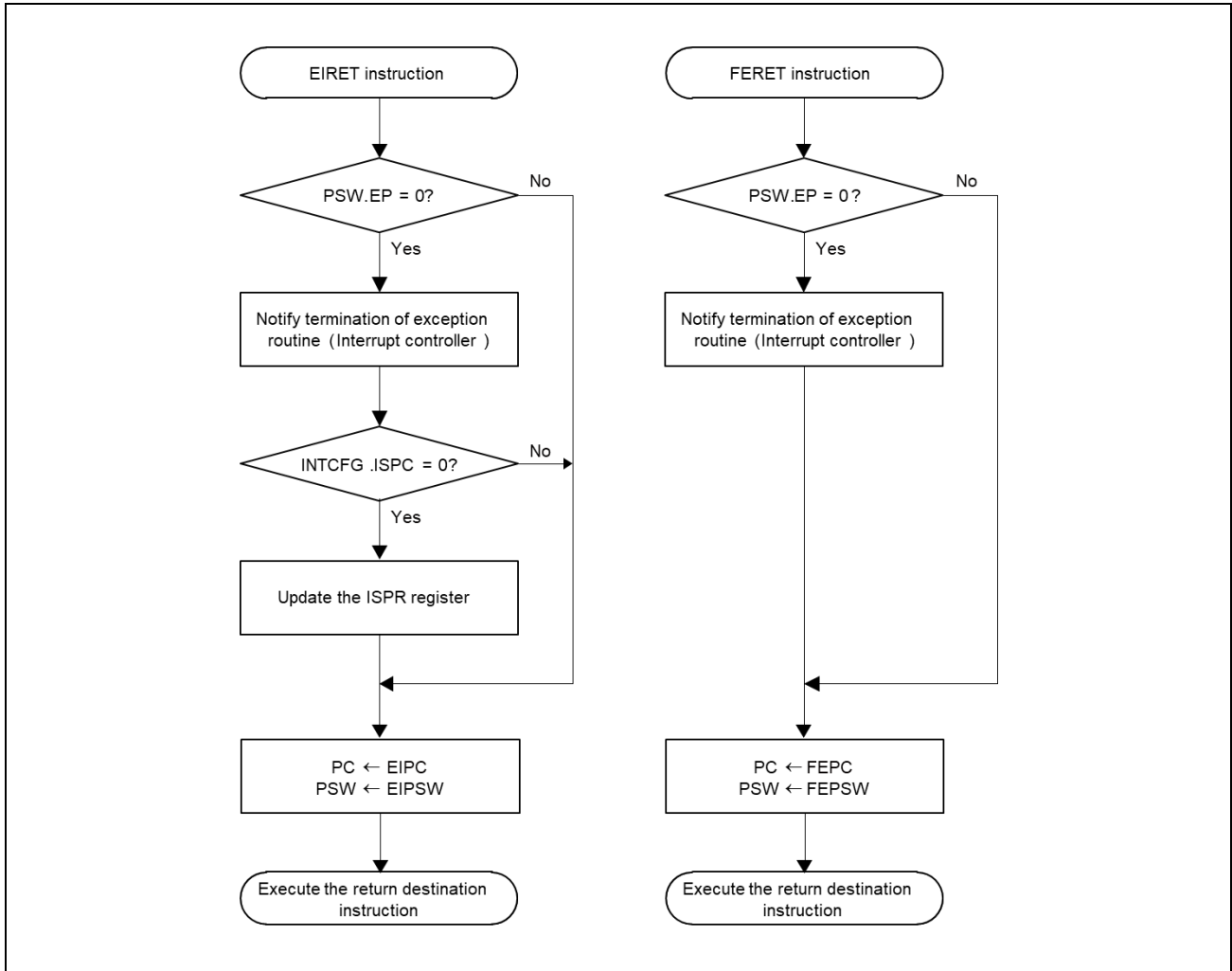


Figure 3.19 Return Instruction-Based Exception Return Flow

3.2.4.4 Exception Handler Address

For this CPU, the exception handler address used for execution during reset input, exception acknowledgment, or interrupt acknowledgment can be changed according to the settings.

(1) Resets, Exceptions, and Interrupts

The exception handler address for resets and exceptions is determined by using the direct vector method, in which the reference point of the exception handler address can be changed by using the PSW.EBV bit, RBASE register, and EBASE register. For interrupts, the direct vector method and table reference method can be specified. If the table reference method is selected, execution can branch to the address indicated by the exception handler table allocated in the memory.

(a) Direct Vector Method

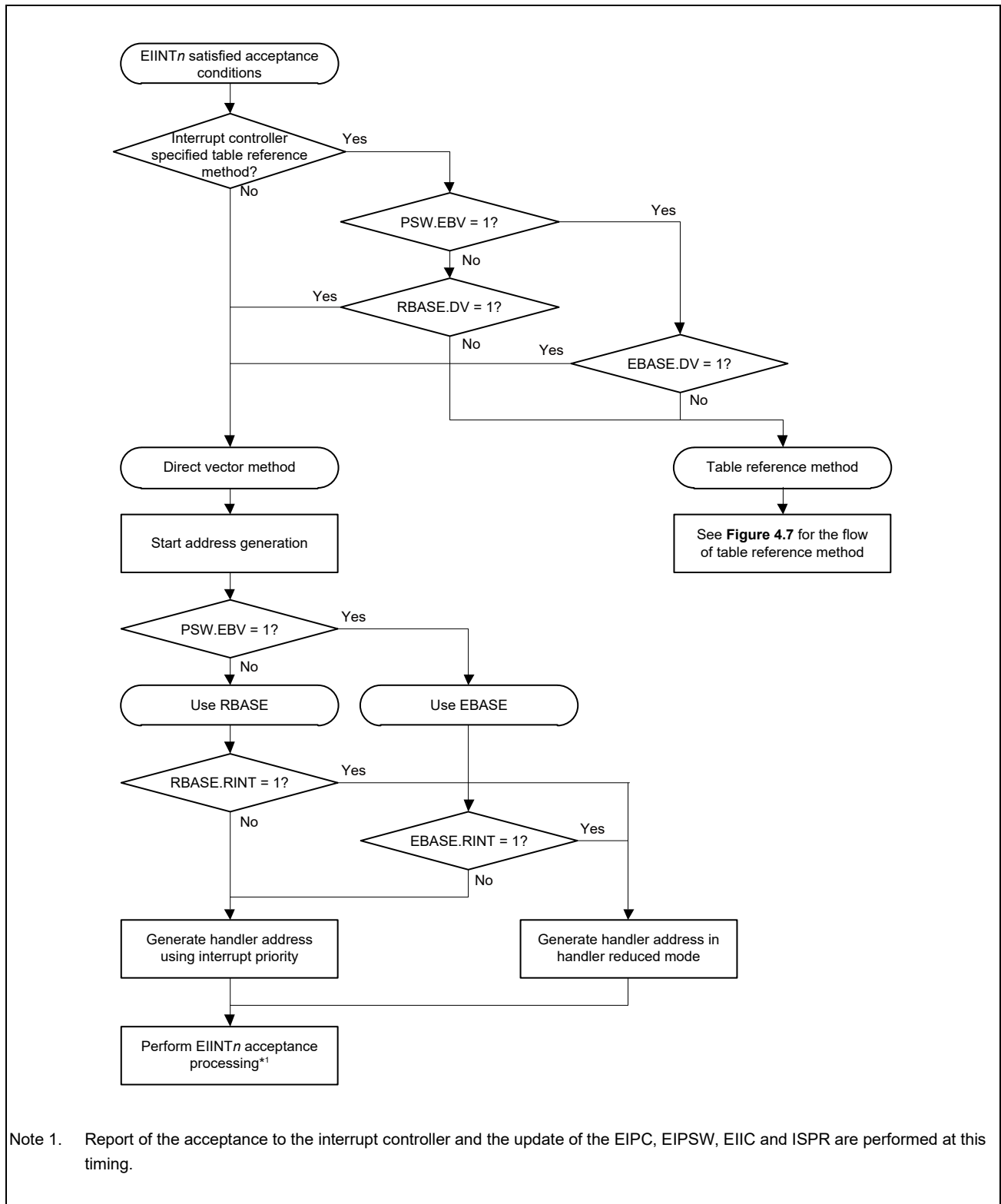
This CPU uses the result of adding the offset address shown in **Table 3.98** to the base address indicated by the RBASE or EBASE register as the exception handler address.

Whether to use the RBASE or EBASE register as the base address is selected according to the PSW.EBV bit. If the PSW.EBV bit is set to 1, the EBASE register value is used as the base address. If the bit is cleared to 0, the RBASE register value is used as the base address.

However, reset input always refers to the RBASE register.

In addition, user interrupts (EIINT n) refer to the RINT bit of the selected base register, and reduce the offset address according to the value of the bit. If the RBASE.RINT bit or EBASE.RINT bit is set to 1, all user interrupts are handled using an offset address of 100_H. If the bit is cleared to 0, the offset address is determined according to **Table 3.98**.

Figure 3.20 shows the flow of generating a handler address for the direct vector method.



Note 1. Report of the acceptance to the interrupt controller and the update of the EIPC, EIPSW, EIIC and ISPR are performed at this timing.

Figure 3.20 Flow of Generating a Handler Address of the Direct Vector Method

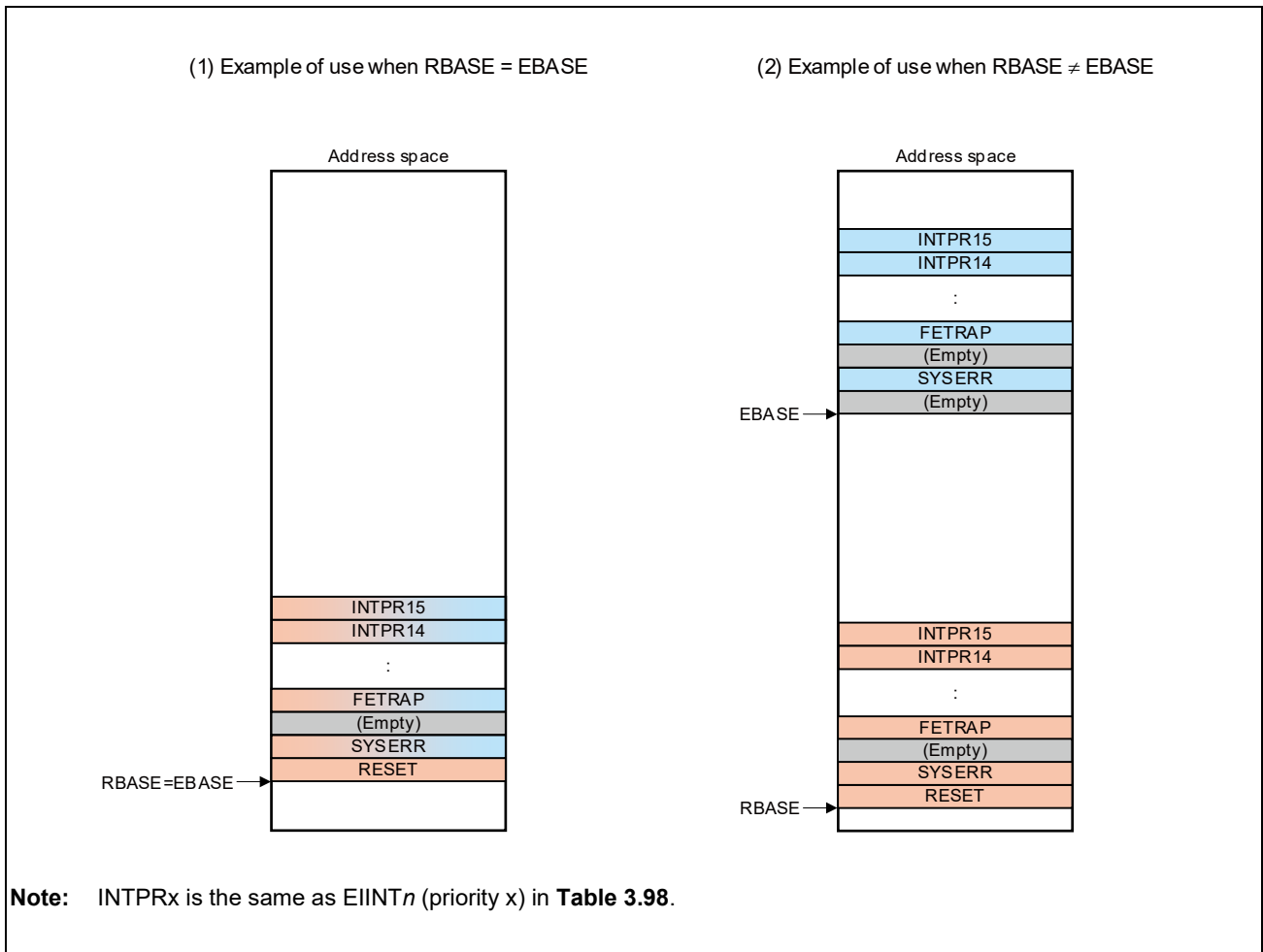


Figure 3.21 Direct Vector Method

The table below shows how base register is selected and offset address for each exception. The value of the PSW register used to determine the exception handler address is the value after being updated due to the acknowledgment of an exception.

Table 3.98 Selection of Base Register/Offset Address

	PSW.EBV = 0	PSW.EBV = 1	RINT = 0	RINT = 1
	Base Register		Offset Address	
RESET	RBASE	None*1	000 _H	000 _H
SYSERR		EBASE	010 _H	010 _H
(R.F.U.)			020 _H	020 _H
FETRAP			030 _H	030 _H
TRAP0			040 _H	040 _H
TRAP1			050 _H	050 _H
RIE			060 _H	060 _H
FPE/FXE			070 _H	070 _H
UCPOP			080 _H	080 _H
MIP/MDP			090 _H	090 _H
PIE			0A0 _H	0A0 _H
(R.F.U.)			0B0 _H	0B0 _H
MAE			0C0 _H	0C0 _H
(R.F.U.)			0D0 _H	0D0 _H
FENMI			0E0 _H	0E0 _H
FEINT			0F0 _H	0F0 _H
EIINT _n (priority 0)			100 _H	100 _H
EIINT _n (priority 1)			110 _H	
EIINT _n (priority 2)			120 _H	
EIINT _n (priority 3)			130 _H	
EIINT _n (priority 4)			140 _H	
EIINT _n (priority 5)			150 _H	
EIINT _n (priority 6)			160 _H	
EIINT _n (priority 7)			170 _H	
EIINT _n (priority 8)			180 _H	
EIINT _n (priority 9)			190 _H	
EIINT _n (priority 10)			1A0 _H	
EIINT _n (priority 11)			1B0 _H	
EIINT _n (priority 12)		1C0 _H		
EIINT _n (priority 13)		1D0 _H		
EIINT _n (priority 14)		1E0 _H		
EIINT _n (priority 15)		1F0 _H		

Note 1. An exception generated to update EBV to 0.

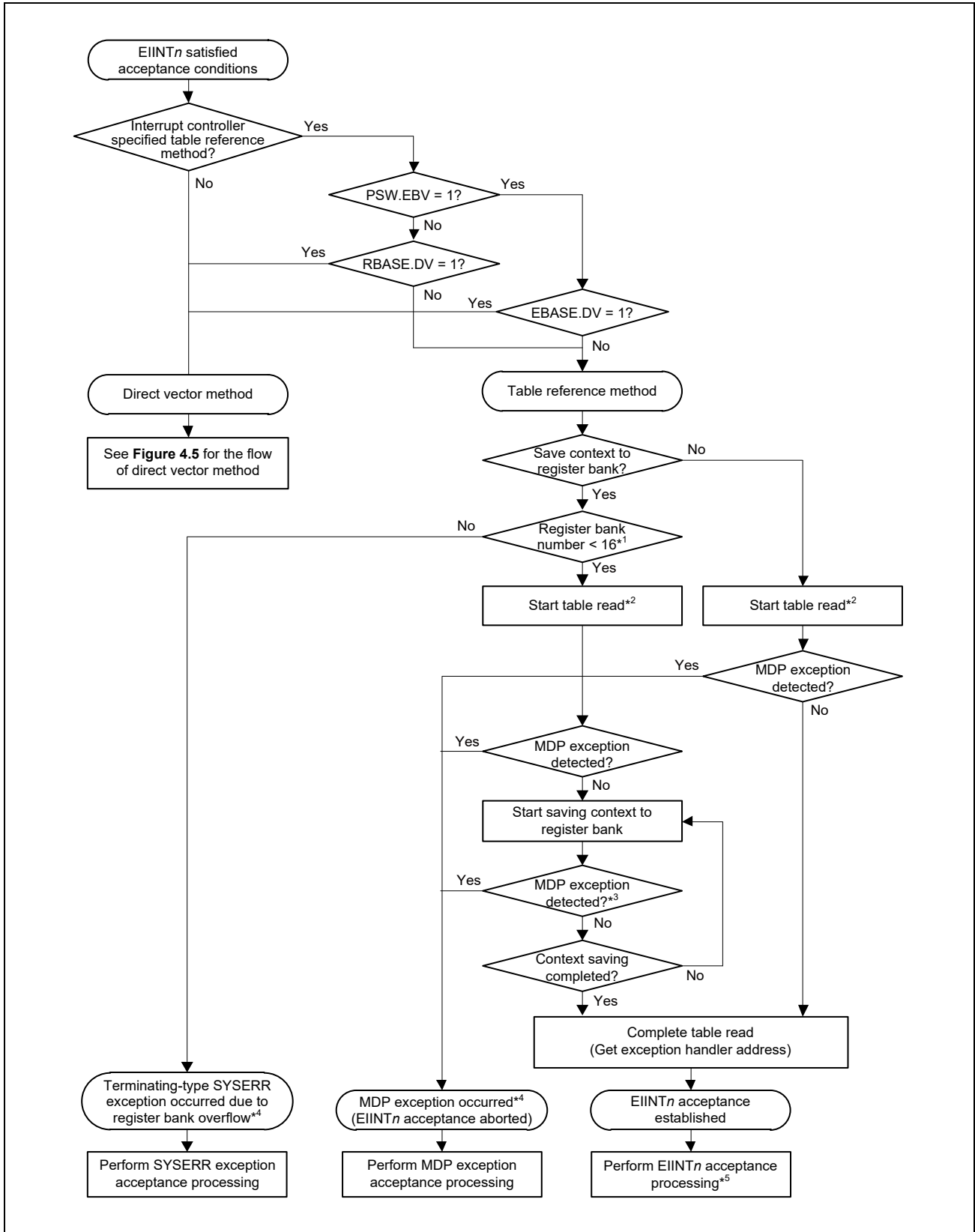
The user interrupt offset address reduction function is used to reduce the memory size required by the exception handler for specific operating modes of the system. The main purpose of this is to minimize the amount of memory consumed in operating modes that use only the minimum functionality, for example, during system maintenance and diagnosis.

(b) Table Reference Method

In the direct vector method, there is one user-interrupt exception handler for each interrupt priority, and user-interrupts with the same priority branch to the same interrupt handler, but some users might want to use code areas that differ from the start time for each interrupt handler.

This CPU defines a table reference method to accommodate to such uses.

Figure 3.22 shows the flow of generating a handler address for the table reference method.



- Note 1. For automatic context saving onto the register bank, the operation condition judgment is necessary. For details, see **Figure 3.24**.
- Note 2. This CPU updates the EIPC and EIPSW at this timing.
- Note 3. Detection of an MDP exception is performed every time an individual context is saved. An MDP exception occurs when an MDP violation is detected when saving an individual context rather than after all context saving is completed.
- Note 4. The PC of the instruction interrupted by the interrupt and the value of the PSW at that time are saved to the FEPC and the FEPSW respectively. These values are the same as the values saved to the EIPC and the EIPSW at the timing of Note 2. Therefore, when returning from the exception handler, it is possible to return exactly to the interrupted instruction. In addition, because no acceptance response has been reported to the interrupt controller, unless canceling the interrupt request in the exception handler, the interrupt request will remain held and interrupt acceptance processing will start again.
- Note 5. Report of the acceptance to the interrupt controller and the update of the EIIC and ISPR are performed at this timing.

Figure 3.22 Flow of Generating a Handler Address of Table Reference Method

- <1> In any of the following cases, the exception handler address is determined by using the direct vector method.
- When the interrupt channel setting is not the table reference method
 - When PSW.EBV = 0 and RBASE.DV = 1
 - When PSW.EBV = 1 and EBASE.DV = 1
- <2> In cases other than <1>, calculate the table read position.
Exception handler address read position = INTBP register + channel number × 4 bytes
- <3> Read word data starting at the exception handler address read position calculated in <2>.
- <4> Use the word data read in <3> as the exception handler address.
- <5> The acceptance of the interrupt is established when the exception handler address has been got. In addition to the exception acceptance processing shown in **Figure 3.18**, the CPU returns an acceptance response to the interrupt controller and updates the ISPR.

CAUTIONS

1. For details about the interrupt channel settings, see **Section 6, Interrupts**.
2. There are cases in which a memory protection exception (MDP) occurs while reading word data from the exception handler address read position depending on the memory protection settings. In such a case, the acceptance of the interrupt is temporarily cancelled. Since no acceptance response is returned to the interrupt controller, the interrupt request is held pending and is become acceptable again when execution is returned from the memory protection exception processing.
3. If a memory protection exception occurs during a word data read from the exception handler address read position, the PC of the instruction that is interrupted by the interrupt is saved in the FEPC. Since the acceptance of the interrupt is cancelled, the value of the PSW established when the interrupt occurred (which should have been saved in the EIPSW) is saved in the FEPSW.

Exception handler address read positions corresponding to interrupt channels and the placement of the interrupt handler address table in memory are shown below.

Table 3.99 Exception Handler Address Read Position

Type	Exception Handler Address Read Position
EIINT interrupt channel 0	INTBP + 0 × 4
EIINT interrupt channel 1	INTBP + 1 × 4
:	:
EIINT interrupt channel 2046	INTBP + 2046 × 4
EIINT interrupt channel 2047	INTBP + 2047 × 4

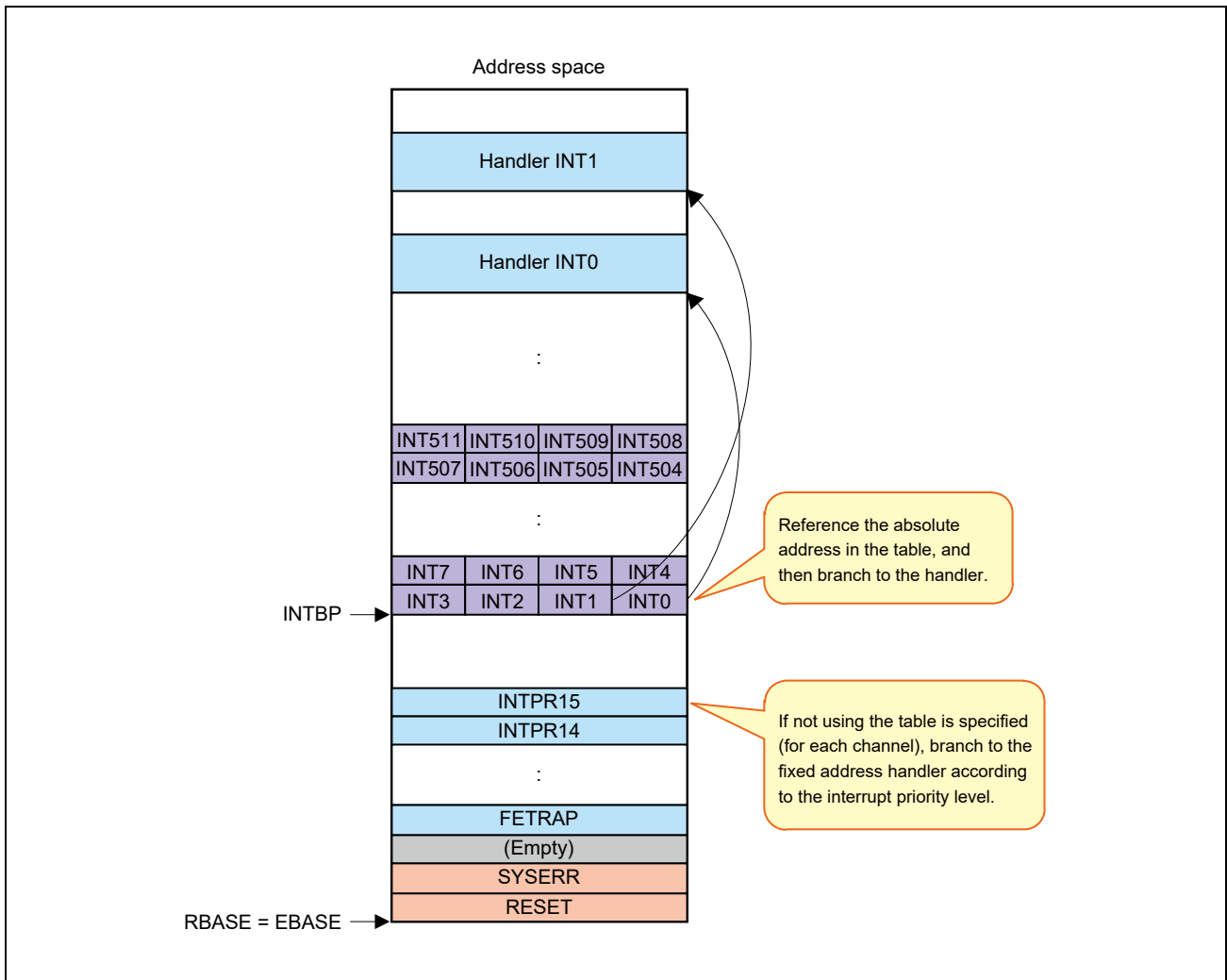


Figure 3.23 Placement of the Interrupt Handler Address Table in Memory

For details about the exception handler address selection method settings for each interrupt channel, see **Section 6, Interrupts**.

(2) System Calls

For system call exceptions, the referenced table entry is selected according to the value of the vector specified by the SYSCALL instruction and the value of the SCCFG.SIZE bit, and the exception handler address is calculated according to the contents of the table entry and the SCBP register value.

As an example, if table size n is specified by SCCFG.SIZE, the table entry is selected as shown below. Note that if the vector specified by the SYSCALL instruction (vector 8) is greater than table size n , the table entry referenced by vector $n + 1$ to 255 is table entry 0.

Table 3.100 System Calls

Vector	Exception Cause Code	Referenced Table Entry
0	0000 8000 _H	Table entry 0
1	0000 8001 _H	Table entry 1
2	0000 8002 _H	Table entry 2
(Omitted)	:	:
$n - 1$	0000 8000 _H + ($n - 1$) _H	Table entry $n - 1$
n	0000 8000 _H + n _H	Table entry n
$n + 1$	0000 8000 _H + ($n + 1$) _H	Table entry 0
(Omitted)	:	:
254	0000 80FE _H	Table entry 0
255	0000 80FF _H	Table entry 0

CAUTION

Because table entry 0 is selected if a vector that exceeds the table size specified by SCCFG.SIZE is specified, allocate the error processing routine at table entry 0.

3.2.4.5 Register Bank Function

This CPU provides the register bank function that automatically saves the context when it accepts an interrupt (EIINT n). Since the saving of the context proceeds in parallel with the interrupt acceptance processing, this function enables the CPU to make high-speed interrupt response.

(1) Outline of the Register Bank Function

The register bank function that this CPU provides has the following features:

- Automatically saves the context upon acceptance of an interrupt (EIINT n) meeting some conditions (see **Section 3.2.4.5(2), Automatic Context Saving.**)
- The destination of the context saving is not dedicated memory but an area of ordinary memory installed in this CPU (specified by the RBIP register).
- The context to be saved can be selected from two groups (specified by the RBCR0.MD bit).
- Supports a maximum of 16 levels of multiple interrupts (the RBNR.BN bits indicate the number of accepted interrupts).
- The context is restored from the register bank by executing the RESBANK instruction (no automatic restoration function).

(2) Automatic Context Saving

Automatic context saving is carried out when the following conditions are satisfied for an interrupt request (EIINT n) notified:

- Table reference method is specified for the interrupt request*¹.
- The use of the register bank is specified by the RBCR0.BE[i] bit which is associated with the priority (i) of the interrupt request (the automatic context saving onto the register bank is enabled).
- The RBNR.BN bits have a value no greater than 15 (the number of register banks in use is not greater than 15).

Note 1. See **Section 6, Interrupts** for the procedure to specify the table reference method.

Automatic context saving is not carried out if the table reference method is not specified for the interrupt request (EIINT n) or the use of the register bank is not specified by the RBCR0.BE[i] bit associated with the interrupt priority (i).

In addition, no automatic context saving is carried out if the value of the RBNR.BN bits are greater than 15. In such a case, it is assumed that an unexpected interrupt acceptance processing is being performed and a terminating-type SYSERR exception is notified for error processing.

Figure 3.24 shows the flow of checking the conditions for automatic context saving.

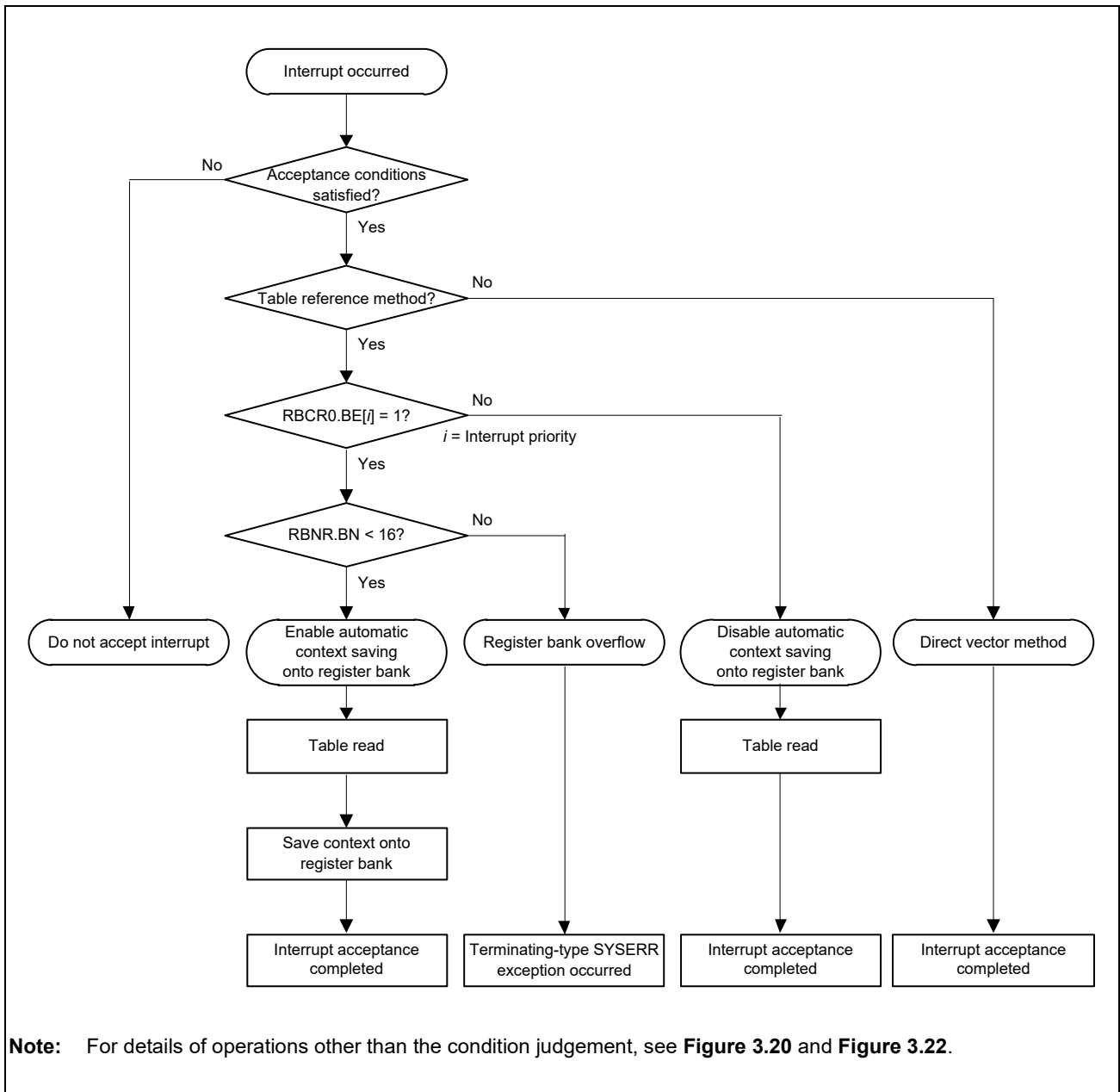


Figure 3.24 Flow of Checking the Conditions for Automatic Context Saving

(a) Automatic Context Saving

Shown below is the processing of automatic context saving onto the register bank.

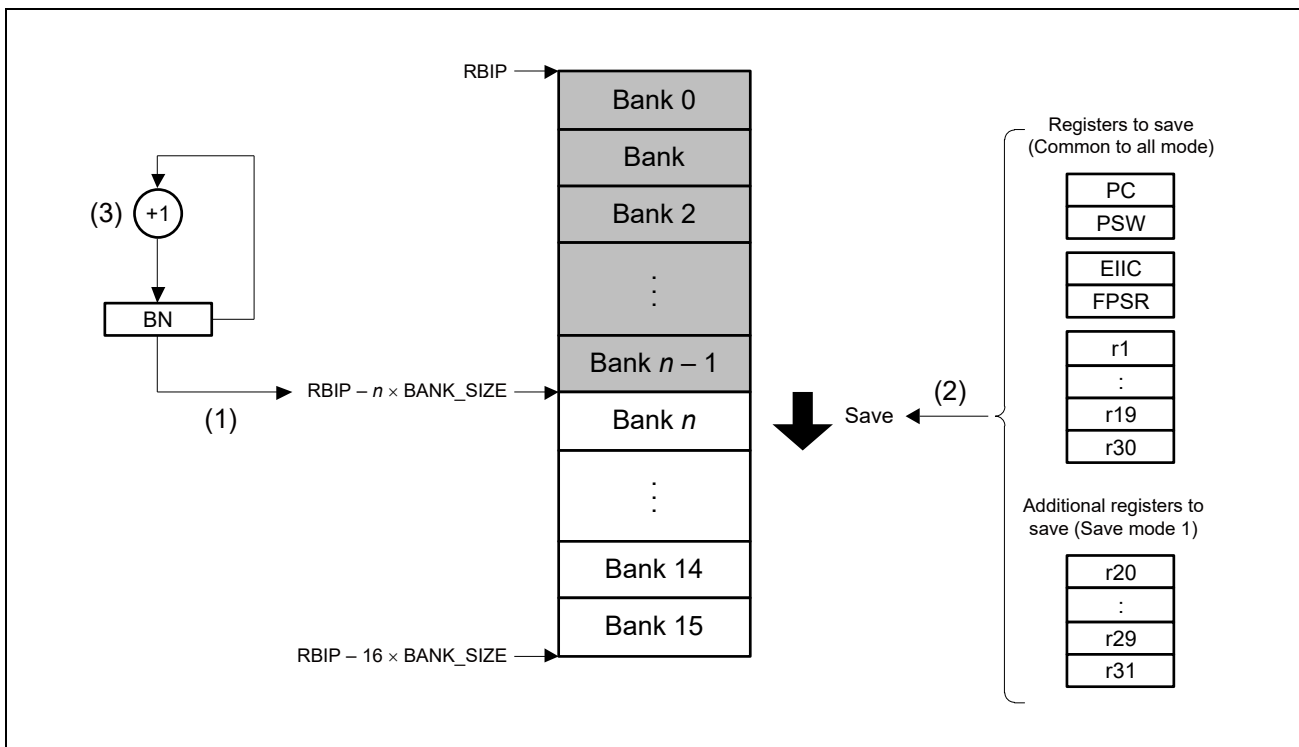


Figure 3.25 Processing of Automatic Context Saving

1. Calculate the start address of register bank n on which the context is to be saved based on the value of the register bank initial pointer (RBIP) and the value (n) of the BN bits of the register bank number register (RBNR) established upon acceptance of the interrupt. The size (BANK_SIZE) of the register bank differs depending on the save mode that is selected.
 - Start address:
 $RBIP - n \times BANK_SIZE$
 - BANK_SIZE:
 - Save mode 0: 60H
 - Save mode 1: 90H
2. Save the target registers onto the register bank n specified by the RBNR.BN bits according to the save mode specified by the RBCR0.MD bit*1. **Table 3.101** shows a list of the registers to save, their addresses, and save order.
3. Increment the value of the RBNR.BN bits by 1. This terminates the register bank saving processing*2 and initiates the execution of the interrupt handler*3.

Note 1. MDP exceptions (terminating-type) might be caused by memory accesses made during the register bank saving processing. The other types of interrupts and exceptions cannot be accepted.

Note 2. The saving of the registers into the memory area specified as the register bank does not have always been completed. To ensure the completion of register saving, this CPU can employ a procedure similar to the one with the ordinary store instruction. For details, see **Section 3.2.7.2, Guaranteeing the Completion of Store Instruction.**

Note 3. Exceptions of higher priorities can be accepted after the saving onto the register bank is finished.

Table 3.101 List of Registers to Save, Addresses, Save Order, and Restore Order

Addresses	Save mode 0	Save mode 1	Save Order	Restore Order
RBIP – n × BANK_SIZE	(MD = 0)	(MD = 1)	*1, *2	*1, *3
-04 _H	PC	PC	1	24 / 35
-08 _H	PSW	PSW	2	23 / 34
-0C _H	EIIC	EIIC	3	22 / 33
-10 _H	FPSR	FPSR	4	21 / 32
-14 _H	r1	r1	5	20 / 31
-18 _H	r2	r2	6	19 / 30
-1C _H	r3	r3	7	18 / 29
-20 _H	r4	r4	8	17 / 28
-24 _H	r5	r5	9	16 / 27
-28 _H	r6	r6	10	15 / 26
-2C _H	r7	r7	11	14 / 25
-30 _H	r8	r8	12	13 / 24
-34 _H	r9	r9	13	12 / 23
-38 _H	r10	r10	14	11 / 22
-3C _H	r11	r11	15	10 / 21
-40 _H	r12	r12	16	9 / 20
-44 _H	r13	r13	17	8 / 19
-48 _H	r14	r14	18	7 / 18
-4C _H	r15	r15	19	6 / 17
-50 _H	r16	r16	20	5 / 16
-54 _H	r17	r17	21	4 / 15
-58 _H	r18	r18	22	3 / 14
-5C _H	r19	r19	23	2 / 13
-60 _H	r30	r20	24	1 / 12
-64 _H	*4	r21	25	11
-68 _H		r22	26	10
-6C _H		r23	27	9
-70 _H		r24	28	8
-74 _H		r25	29	7
-78 _H		r26	30	6
-7C _H		r27	31	5
-80 _H		r28	32	4
-84 _H		r29	33	3
-88 _H		r30	34	2
-8C _H		r31	35	1
-90 _H		*5		
-94 _H		*6		

Note 1. Register saving and restoration start sequentially at sequence number 1.

Note 2. The target registers to be saved in save mode 0 (RBCR0.MD = 0) are the registers up to r30 with a save sequence number of 24.

Note 3. The restoration sequence numbers shown on the left side are for save mode 0 and those shown on the right side are for save mode 1.

Note 4. In save mode 0, the next bank (n + 1) starts from this address.

Note 5. In save mode 1, no register is saved in this address. This address is not used.

Note 6. In save mode 1, the next bank (n + 1) starts from this address.

(b) Suppressing the Update of the PSW.ID Bit

When an interrupt (EIINT n) is accepted in the normal state, the PSW.ID bit is set to 1 and interrupts of the same EI level cannot be accepted unless the acceptance of interrupts is enabled explicitly with the EI instruction.

For interrupts that make use of the register bank (satisfying the automatic context saving conditions), on the other hand, by clearing the RBCR1.NC[i] bit associated with the interrupt priority (i) to 0, the PSW.ID bit keeps its value 0 without being set to 1 when an interrupt is accepted. This makes it possible to accept interrupts of higher priorities without software intervention after the end of automatic context saving. Since necessary registers are automatically saved, the CPU can return to the original interrupt processing precisely.

(3) Context Restoration

The context automatically saved onto the register bank need to be restored before returning from the interrupt processing to the original program. Although the context saving onto the register bank is automatically done, its restoration needs to be accomplished explicitly by software executing the RESBANK instruction.

(a) Conditions for Executing the RESBANK Instruction

The RESBANK instruction must be executed to restore the context from the register bank. The RESBANK instruction is a supervisor-privileged instruction. A PIE exception will occur if it is executed in user mode.

A resumable-type SYSERR exception will occur if the RESBANK instruction is executed when the value of the RBNR.BN bits are 0. A 0 in the RBNR.BN bits means that no context has automatically been saved onto the register bank. Accordingly, an invalid value will be restored if the RESBANK instruction is executed in such case.

The above-mentioned operating conditions are summarized in **Figure 3.26**.

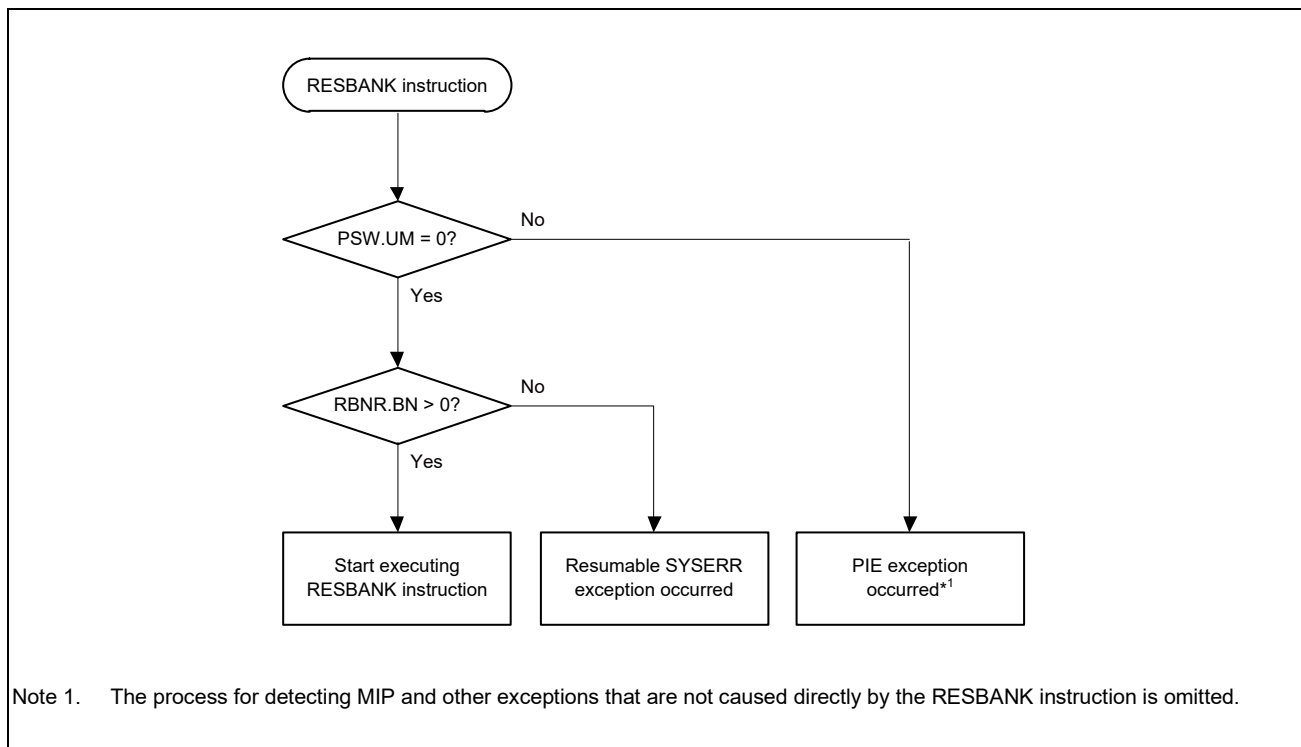


Figure 3.26 Flow of Checking the Conditions for Executing the RESBANK Instruction

(b) Context Restoration

The figure below shows the processing of the RESBANK instruction for restoring the context from the register bank.

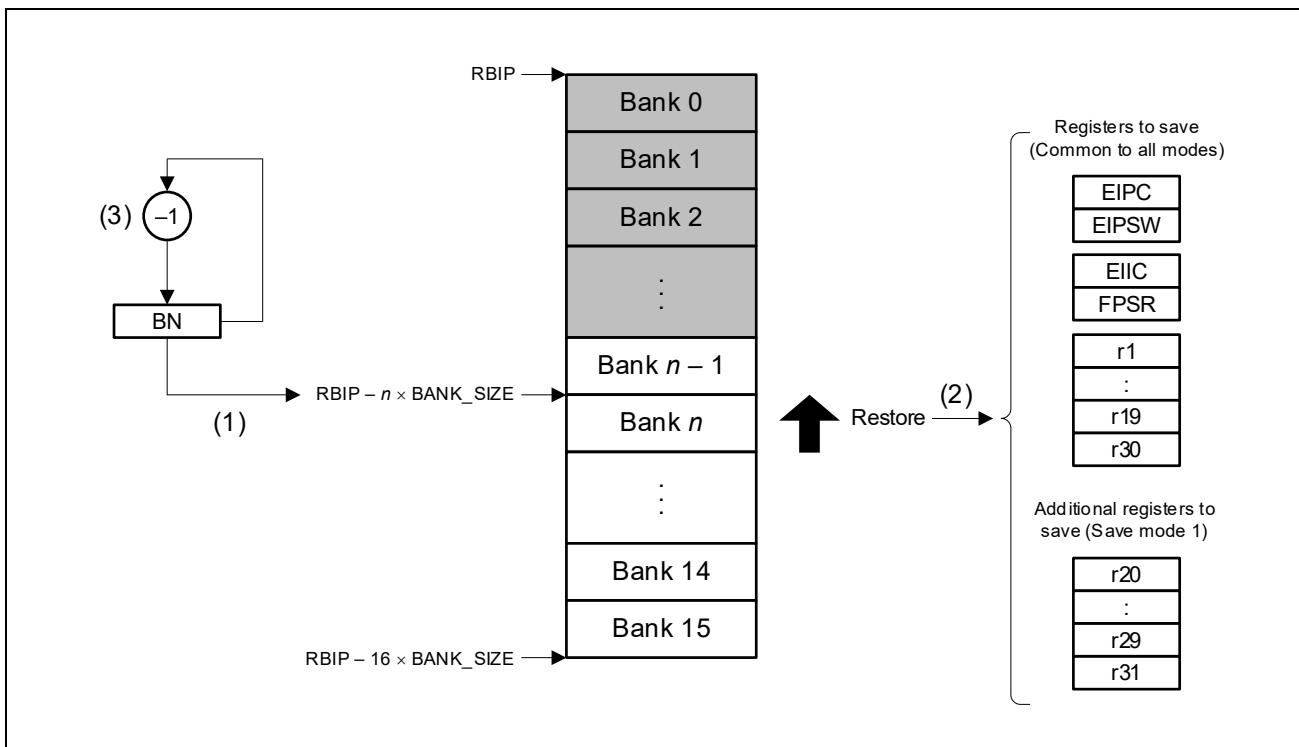


Figure 3.27 Processing of Context Restoration

1. Calculate the start address of register bank $n-1$ from which the context is to be restored based on the value of the register bank initial pointer (RBIP) and the value (n) of the BN bits of the register bank number register (RBNR) established upon acceptance of the interrupt. Unlike in the case of saving the context, the start address to be referenced when restoring the context is the lower-limit side of the register bank addresses. The size (BANK_SIZE) of the register bank differs depending on the save mode that is selected.

Start address:

$$\text{RBIP} - n \times \text{BANK_SIZE}$$

BANK_SIZE :

Save mode 0: 60_H

Save mode 1: 90_H

2. Restore the context from the register bank $n-1$ in the target registers according to the save mode specified by the RBCR0.MD bit*¹. **Table 3.101** shows a list of the registers to restore, their addresses, and restore order.
3. Decrement the value of the RBNR.BN bits by 1*². This terminates the register bank restoration processing and completes the execution of the RESBANK instruction*³.

Note 1. The CPU can accept terminating-type exceptions while restoring the context from the register bank. MDP exceptions (resumable-type) might be caused by memory accesses made during the register bank restoration processing.

Note 2. When the CPU accepts an exception before the value of RBNR.BN is updated, it stops the execution of the RESBANK instruction even if the restoration of all registers is not yet completed. In this case, though there are some registers of which restoration has been completed, the CPU cannot know what registers have been restored. Since the return PC of the exception is the PC of this RESBANK instruction, the CPU can re-execute

precisely the RESBANK instruction if none of the resources related to the RESBANK instruction are altered during the exception processing.

Note 3. After the RESBANK instruction is executed, execute the EIRET instruction to return from the interrupt handler.

3.2.4.6 List of Memory Access Exceptions

Table 3.102 shows a list of instructions and exceptions that make memory accesses and the exceptions that can be detected. Exceptions identified by the symbol “✓” can be detected during the memory accesses that are made during the processing of the listed instructions and exceptions. Exceptions identified by the symbol × are not detected.

Table 3.102 List of Memory Access Exceptions

Instruction/Exception	MAE	MDP	Instruction/Exception	MAE	MDP
SLD.B	×	✓	TST1	×	✓
SLD.BU	×	✓	PREPARE	×	✓
SLD.H	✓	✓	DISPOSE	×	✓
SLD.HU	✓	✓	PUSHSP	×	✓
SLD.W	✓	✓	POPSP	×	✓
SST.B	×	✓	SWITCH	×	✓
SST.H	✓	✓	CALLT	×	✓
SST.W	✓	✓	SYSCALL	×	✓
LD.B	×	✓	LDV.W	✓	✓
LD.BU	×	✓	LDV.DW	✓	✓
LD.H	✓	✓	LDV.QW	✓	✓
LD.HU	✓	✓	STV.W	✓	✓
LD.W	✓	✓	STV.DW	✓	✓
LD.DW	✓	✓	STV.QW	✓	✓
ST.B	×	✓	LDVZ.H4	✓	✓
ST.H	✓	✓	STVZ.H4	✓	✓
ST.W	✓	✓	CACHE (CHBII)	×	✓
ST.DW	✓	✓	CACHE (CIBII)	×	×
LDL.BU	×	✓	CACHE (CFALI)	×	✓
LDL.HU	✓	✓	CACHE (CISTI)	×	×
LDL.W	✓	✓	CACHE (CILDII)	×	×
STC.B	×	✓	CACHE (other commands)	×	×
STC.H	✓	✓	PREF (PREFI)	×	×*1
STC.W	✓	✓	PREF (other commands)	×	×
CAXI	✓	✓	Table reference type EIINT n	×	✓*2
SET1	×	✓	Register bank saving processing	×	✓*2
CLR1	×	✓	RESBANK	×	✓
NOT1	×	✓			

Note 1. When a violation is detected during the execution of the PREF instruction, no MDP exception is generated but the memory access is suppressed. Read access is not performed.

Note 2. Both of table read accesses due to table reference method interrupts and write accesses for automatically saving the context onto the register bank occur independently of the execution of an instruction. This CPU handles any MDP exception that is caused by these accesses as a terminating-type exception. The exception cause code of this MDP exception differs from that of resumable-type MDP exceptions. For details see **Table 3.95**.

3.2.5 Memory Management

This CPU provides the following functions for managing memory.

- Memory protection unit (MPU)
- Instruction cache function
- Mutual exclusion function
- Special instructions for synchronization processing
- Guaranteeing the completion of store instructions

3.2.5.1 Memory Protection Unit (MPU)

Memory protection functions are provided in an MPU (memory protection unit) to maintain a smooth system by detecting and preventing unauthorized use of system resources by unreliable programs, runaway events, etc.

(1) Features

(a) Memory Access Control

Multiple protection areas can be assigned to the address space. Consequently, unauthorized program execution or data manipulation by user programs can be detected and prevented. The upper and lower limit addresses of each area can be specified so that the address space can be used precisely and efficiently.

(b) Access Management for Each CPU Operation Mode

In this CPU, several status bits are used to control access to resources, and these bits are used in combination to perform protection that is appropriate, according to each program's level of reliability.

(c) Protection with the System Protection Identifier (SPID)

This CPU can use the system protection identifier (SPID) to check for area matching. The usage and limitations on settings of the SPID register depend on the system specifications of the product in which this CPU is mounted. Reflecting such system specifications in the control of memory access can enable the efficient management of access.

(2) Protection Area Settings

(a) Protection Area Attribute Settings

Set the respective protection areas appropriately. For details about registers, see **Section 3.2.3, Register Set**.

3.2.5.1.2.a.1 E bit

This sets the target protection area setup as enabled or disabled. When disabled, all settings are disabled. Make sure valid setting values have been stored for other protection area settings (MPUA, MPLA, and MPAT) at the time when this bit is set to 1.

3.2.5.1.2.a.2 UX, UR, and UW bits

These bits indicate the access privileges for the target protection area during user mode. A 1 in these bits indicates the presence of the corresponding access privilege.

3.2.5.1.2.a.3 SX, SR, and SW bits

These bits indicate the access privileges for the target protection area during supervisor mode. These bits, in addition to the E bit, are valid only when the MPM.SVP bit has been set to 1. If the MPM.SVP bit has been cleared to 0, even when the E bit is set to 1, protection is not performed while in supervisor mode, regardless of the values of the SX, SR, and SW bits, and the entire address space becomes access-enabled.

3.2.5.1.2.a.4 WG and WMPIDn bits and MPIDn registers (n = 0 to 7)

These bits indicate how to reference the system protection identifier (SPID) during the check for the permission to write to a protection area.

When the WG bit is set (to 1), the access permission for all write accesses is judged only by the settings of the UW or SW bit regardless of the value of the SPID that is set.

When the WG bit is cleared (to 0), if the value of the SPID register matches any of the values specified in the MPIDn registers and if the WMPIDn bit corresponding to the matching MPIDn register is set (to 1), the access permission is judged according to the settings of the UW or SW bit. If the value of the SPID register matches none of the values specified in the MPIDn registers or if the WMPIDn bit corresponding to the matching MPIDn register is cleared (to 0), the access permission is judged to be “write-disabled” regardless of the settings of the UW or SW bit.

In addition, when the values of multiple MPIDn registers match the value of the SPID register, writing is possible if any of the corresponding WMPIDn bit is set (to 1).

3.2.5.1.2.a.5 RG bit, RMPIDn bit, and MPIDn registers (n = 0 to 7)

These bits indicate how to reference the system protection identifier (SPID) during the check for the permission to read from a protection area.

When the RG bit is set (to 1), the access permission for all read accesses is judged only by the settings of the UX and UR bits or the SX and SR bits regardless of the value of the SPID that is set. When the RG bit is cleared (to 0), if the value of the SPID register matches any of the values specified in the MPIDn registers and if the RMPIDn bit corresponding to the matching MPIDn register is set (to 1), the access permission is judged according to the settings of the UX and UR bits or the SX and SR bits. If the value of the SPID register matches none of the values specified in the MPIDn registers or if the RMPIDn bit corresponding to the matching MPIDn register is cleared (to 0), the access permission is judged to be “read-disabled” regardless of the settings of the UX and UR bits or the SX and SR bits.

In addition, when the values of multiple MPIDn registers match the value of the SPID register, reading is possible if any of the corresponding RMPIDn bit is set (to 1).

(3) Caution Points for Protection Area Setup

(a) Crossing Protection Area Boundaries

When the specified protection areas overlap, the access control settings for the overlapping parts are set to “allow most”.

In other words, when multiple protection areas have been specified, if access is enabled for either of the protection areas, access is judged to be enabled.

(b) Invalid Protection Area Settings

Protection area settings are invalid in the following case.

- When value set to lower-limit address is larger than value set to upper-limit address

CAUTION

Addresses are handled as unsigned integers (0_H to FFFF FFFF_H).

For example, the following settings alone cannot make the target area accessible (MPAT is omitted):

MPLA0 = FFFF FF80_H, MPUA0 = 0000 00FC_H

This setting should be divided into the following two protection area settings.

MPLA0 = FFFF FF80_H, MPUA0 = FFFF FFFC_H

MPLA1 = 0000 0000_H, MPUA1 = 0000 00FC_H

MPLA0 refers to the MPLA register for the protection area 0. The same holds true for the other registers.

(c) Lower- and Upper-Limit Addresses Referenced during Protection Violation Checks

Table 3.103 lists the lower-limit addresses that are compared with MPLA and the upper-limit addresses that are compared with MPUA during the protection violation check.

Table 3.103 Lower- and Upper-limit Addresses Referenced during the Protection Violation Check

Instruction/Event	Access size	Lower-Limit Address	Upper-Limit Address
SLD, SST, LD, ST, LDL, STC, LDV, STV	Byte	Address calculated according to the addressing specified in the instruction	Same as lower-limit address
	Half word		Lower-limit address + 1
	Word		Lower-limit address + 3
	Double word* ¹		Lower-limit address + 7
	Quad word* ¹		Lower-limit address + 15
CAXI	Word	Value of reg1	Lower-limit address + 3
SET1, NOT1, CLR1, TST1	Byte	Address calculated according to the addressing specified in the instruction	Same as lower-limit address
PREPARE, DISPOSE, PUSHSP, POPSP	Word* ²	Address calculated from the value of the SP and the number of accesses	Lower-limit address + 3
SWITCH, CALLT	Half word	Address calculated according to the addressing specified in the instruction	Lower-limit address + 1
SYSCALL	Word	Address calculated according to the default addressing	Lower-limit address + 3
CACHE (CHBII, CFALI), PREF	Cache line size* ³	reg1 value rounded by cache line size	Lower-limit address + Cache line size – 1
Table reference EIINT _n	Word	Address from which the exception handler address is read when table reference method is selected	Lower-limit address + 3
Automatic context saving onto a register bank	Word* ²	Address calculated from the values of the RBIP and the RBNR.BN bit	Lower-limit address + 3
RESBANK	Word* ²	Address calculated from the values of the RBIP and the RBNR.BN bit	Lower-limit address + 3
Memory protection configuration check function	Value of MCS	Value of MCA	Value of MCA + MCS – 1
Instruction fetch	Instruction fetch size* ⁴	PC rounded by the instruction fetch size	Lower-limit address + Instruction fetch size – 1

Note 1. Since the minimum protection area unit is word-size, there are cases in which the lower- or upper-limit address specified for a double or quad word access does not fall within the address range designated by a single protection area setting. If either one of the lower- and upper-limit addresses does not fall within the specified address range, a protection violation is detected and an MDP exception is generated. See **Section 3.2.5.1(3)(d), Memory Access Spanning Contiguous Protection Areas** for the operation for two contiguous protection area settings.

Note 2. The instruction accesses contiguous multi-byte area, however, memory accesses for a single register are carried out on word-size basis. Memory protection is effected on each of these word-size accesses. For this reason, there are cases in which a data protection violation is detected when some of word-size memory access are executed and there still are memory accesses that are yet to be executed. In these cases, an MDP exception occurs during the processing of the instruction, and the instruction is aborted. Memory or registers are updated by the completed processing of the instruction.

Note 3. For the cache line size, see **Section 3.8.9, Product Information of Cache Structure**.

Note 4. For the instruction fetch size, see **Section 3.8.10, Product Information of Fetch Size**.

(d) Memory Access Spanning Contiguous Protection Areas

With respect to the memory protection against an operand access, it is necessary that the associated memory access is encompassed entirely in a single access enabled area. No accesses that span over two or more areas are allowed even when the access enabled areas are allocated in contiguous spaces. In this CPU, since the minimum protection area unit is word-size and misaligned accesses cannot be executed, memory accesses spanning over areas can occur during double word size memory accesses (LD.DW, ST.DW, LDV.DW, and STV.DW) or quad-word memory access (LDV.QW, STV.QW).

On the other hand, the memory accesses associated with the instructions PREPARE, DISPOSE, PUSHSP, POPSP, and RESBANK and the memory accesses associated with the automatic context saving onto a register bank may span over two or more access enabled areas since they are handled as repetitions of two or more word size accesses.

With respect to the memory protection against an instruction fetch, if a fetch access spans over protection areas, the instruction codes that are fetched are given the result of judging the memory protection which is made with the word size as a delimiter. Since only a specific part of the instruction codes that are fetched is required to execute the instruction, even if an instruction fetch which spans over enabled and disabled areas is made, the instruction code in the enabled area can be executed. If the instruction code in the disabled area is used, an MIP exception will occur.

Since the instruction codes are allocated on a half word basis, for an instruction whose instruction length is the word size or longer, there are cases in which the instruction itself spans over protection areas. In such a case, the instruction may be executed if both areas are enabled. If one area is disabled, however, the instruction cannot be executed even if the other area is enabled.

Table 3.104 below shows the examples of operand accesses which span over protection areas. **Table 3.105** below shows the examples of instruction fetches which span over protection areas. In the following examples, the LD.DW, DISPOSE instructions and the instruction fetch start to access address 0000 00FC_H and proceed in the direction of increasing addresses. The PREPARE instruction starts to access address 0000 0100_H and proceeds in the direction of decreasing addresses. The memory protection setting check function checks the area that begins from 0000 00FC_H and goes beyond the boundary of the protection area. In the examples, following two protection areas are set.

Protection area setting 0: MPLA0 = 0000 0000_H, MPUA0 = 0000 00FC_H

Protection area setting 1: MPLA1 = 0000 0100_H, MPUA1 = 0000 01FC_H

Table 3.104 Examples of Operand Memory Accesses Spanning Over Contiguous Protection Areas

Protection Setting		Operation			
Setting 0	Setting 1	LD.DW, etc.	PREPARE, etc. (save context)	DISPOSE, etc. (restore context)	Memory Protection Setting Check
Disabled	Disabled	MDP	MDP	MDP	Prohibited (0) ^{*5}
Disabled	Enabled	MDP	MDP ^{*1}	MDP ^{*3}	Prohibited (0) ^{*5}
Enabled	Disabled	MDP	MDP ^{*2}	MDP ^{*4}	Prohibited (0) ^{*5}
Enabled	Enabled	MDP	Access executed	Access executed	Prohibited (0) ^{*5}

- Note 1. Accesses made to addresses in the protection area setting 1 are enabled and update the memory. When an access is made to an address in the protection area setting 0, an MDP exception occurs and the execution of the instruction is aborted. The MEA register holds the address at which the first data protection violation is detected.
- Note 2. Since the address at which accessing is to start is disabled, an MDP exception occurs without the execution of any memory access.
- Note 3. Since the address at which accessing is to start is disabled, an MDP exception occurs without the execution of any memory access.
- Note 4. Accesses made to addresses in the protection area setting 0 are enabled and update the registers. When an access is made to an address in the protection area setting 1, an MDP exception occurs and the execution of the instruction is aborted. The MEA register holds the address at which the first data protection violation is detected.
- Note 5. The memory protection setting check function sets "Prohibited (0)" to the corresponding bits of the MCR register unless the target area falls within a single protection area.

Table 3.105 Examples of Instruction Fetch Memory Accesses Spanning Over Contiguous Protection Areas

Protection Setting		Operation			
Setting 0	Setting 1	Execute Instructions Covered by Only Setting 0	Execute Instructions Covered by Only Setting 1	Execute Instructions Spanning Over Settings 0 and 1	Manipulate Instruction Cache with CACHE Instruction/Prefetching by the PREF Instruction
Disabled	Disabled	MIP	MIP	MIP	MDP ^{*1} /Disabled ^{*2}
Disabled	Enabled	MIP	Instruction executed	MIP	MDP ^{*1} /Disabled ^{*2}
Enabled	Disabled	Instruction executed	MIP	MIP	MDP ^{*1} /Disabled ^{*2}
Enabled	Enabled	Instruction executed	Instruction executed	Instruction executed	MDP ^{*1} /Disabled ^{*2}

- Note 1. An MDP exception occurs when a memory protection violation is detected for the CACHE instruction which manipulates the instruction cache using the address specification method.
- Note 2. An MDP exception does not occur, and prefetching is disabled when a PREF instruction leads to a violation of memory protection.

(e) Memory Access which Spans Over Address 0000 0000_H

Table 3.106 summarizes the operations that are performed when a memory access which spans over addresses FFFF FFFF_H to 0000 0000_H is made with the protection setting such that all the address spaces are enabled by the following single protection area setting:

Protection area setting 0: MPLA0 = 0000 0000_H, MPUA0 = FFFF FFFC_H

Table 3.106 Operations of Memory Access That Span Over Address 0000 0000_H

Instruction/Event	Operation
LD.DW, ST.DW	MDP exception
PREPARE, DISPOSE, PUSHSP, POPSP, RESBANK, Automatic context saving onto a register bank	Access executed

For instructions and events other than those mentioned above, no memory access which spans over address 0000 0000_H is generated by a single instruction or event.

(4) Access Control

In this CPU, accesses are controlled appropriately according to the settings described in **Section 3.2.5.1(2), Protection Area Settings**. In any of the cases listed below, the CPU ensures logical integrity by limiting actual access, detecting violations before instruction execution is completed, and setting up exceptions.

- When about to execute an instruction at an address outside the executable area
- When about to execute an instruction that reads from an address outside the read-accessible area
- When about to execute an instruction that writes to an address outside the write-accessible area

The specifics of access control are as follows.

- The result of the access which is judged as prohibited is not reflected in memory or I/O devices.
- The result of the access which is judged as enabled is reflected in memory or I/O devices.

CAUTIONS

1. Even when access is enabled, there might be cases where access is blocked by another function that prohibits it.
2. In some cases, access judged to be prohibited may be executed for a memory. The cases are as listed below.
 - Reading local RAM
 - Reading of code flash memory by instruction prefetching of the instruction cache

Since execution of the instructions that read from the local RAM or execute the prefetched instruction is inhibited by an exception, such access does not affect the execution of instructions. However, when a debugger is monitoring access to the local RAM or code flash memory, it may observe access judged to be prohibited.

(5) Violations and Exceptions

In this CPU, violations are detected during instruction fetch access or operand access according to the protection area settings, and an exception is generated.

- Execution protection violation (during instruction fetch access)
- Data protection violation (during operand access)

(a) Execution Protection Violation (MIP Exception)

This violation is detected when an instruction is executed. An execution protection violation is detected when attempting to execute an instruction that has been placed in a non-executable area.

When an execution protection violation is detected, an MIP exception always occurs.

(b) Data Protection Violation (MDP Exception)

This violation is detected during operand access. A data protection violation is detected when a memory access instruction, etc. attempts to access data from an access-prohibited area.

When a data protection violation is detected, an MDP exception always occurs.

(c) Exception Cause Code and Exception Address

When an execution protection violation or data protection violation has been detected, the exception cause code is determined as shown in **Table 3.107**. The determined exception cause code is set to the FEIC register.

The MEA register is used to store either the PC of the instruction that detected the execution protection violation or the access address used when the data protection violation occurred. The MEA register is shared by MIP and MDP exceptions since these exceptions do not occur simultaneously. Also, when a data protection violation occurs, the information of the instruction or event that caused the violation is stored in the MEI register.

Table 3.107 Exception Cause Code of Memory Protection Violation

Exception	Operation Mode When Violation Occurred	Bit Number and Bit Name										
		31-25	24	23	22	21	20	19	18	17	16	15-0
		—	MS	BL	RMW	SX	SW	SR	UX	UW	UR	—
MIP	User mode	0	0	0	0	0	0	0	1	0	0	90 _H
	Supervisor mode	0	0	0	0	1	0	0	0	0	0	90 _H
MDP	User mode	0	*5	*4	*3	0	0	0	0	*2	*1	91 _H
	Supervisor mode	0				0	*2	*1	0	0	0	91 _H *6

- Remarks:**
- UR: A violation is detected during a read operation in user mode (PSW.UM = 1).
 - UW: A violation is detected during a write operation in user mode (PSW.UM = 1).
 - UX: A violation is detected during instruction execution in user mode (PSW.UM = 1).
 - SR: A violation is detected during a read operation in supervisor mode (PSW.UM = 0).
 - SW: A violation is detected during a write operation in supervisor mode (PSW.UM = 0).
 - SX: A violation is detected during instruction execution in supervisor mode (PSW.UM = 0).
 - RMW: Set to 1 when the instruction causing the violation contains a read-modify-write operation (SET1, NOT1, CLR1, or CAXI).
 - BL: Set to 1 when the instruction causing the violation performs a block transfer (PREPARE, DISPOSE, PUSHSP, or POPSP).
 - MS: Set to 1 when the instruction causing the violation performs a misaligned access.

Note 1. When a read violation is caused by an instruction that includes a read operation, either the SR or UR bit is set to 1.

Note 2. When a write violation is caused by an instruction that includes a write operation, either the SW or UW bit is set to 1.

Note 3. This bit is set to 1 when a violation is caused by the SET1, NOT1, CLR1, or CAXI instruction.

Note 4. This bit is set to 1 when a violation is caused by the PREPARE, DISPOSE, PUSHSP, or POPSP instruction.

Note 5. This bit is set to 1 when the instruction causing the violation performs a misaligned access.

Note 6. These bits are loaded with 95_H for a data protection violation that is detected during a table read initiated by table reference EIINTn or automatic context saving onto a register bank. These are executed only in supervisor mode.

(6) Memory Protection Setting Check Function

For the programs, such as OS, that provide services, this CPU provides a memory protection setting check function to enable implementation of a service protection function that checks in advance whether or not the address area to be used for the requested operations is within an area that is accessible by the source that requested the service. By using this function, when verifying the validity of the user-supplied parameters for a system service, the OS can complete the verification in a shorter time than by repeating reading and checking the area settings by software.

(a) Check Details

The memory protection setting check function checks what access permission settings are made for the specified address area according to the protection area settings that are stored in the MPU.

If the address area specified by the MCA and MCS registers is contained within one protection area (specified by a pair of the MPLA and MPUA registers) and the instruction execution (instruction fetch)/write access/read access for the SPID specified by the MCI register is permitted in supervisor mode/user mode, the corresponding bit in the MCR register is set (to 1).

If the specified address area is not contained within one protection area or an access for the specified SPID is not permitted in an operation mode, the corresponding bit in the MCR register is cleared (to 0).

This check operation is realized by executing the same memory protection judgement procedure that is applied to the operand access or instruction fetch access to the specified address area and SPID. Therefore, when the checked address area is accessed by a load instruction, store instruction or instruction fetch, the access is judged as permitted or a memory protection exception occurs according to the check result.

CAUTIONS

1. For the memory protection setting check function to judge the access as permitted, it is necessary that the specified address area is contained within one protection area. For example, when two contiguous protection areas are set as access permitted (see **Section 3.2.5.1(3)(d), Memory Access Spanning Contiguous Protection Areas**), the memory protection setting check result for the address area spanning over the contiguous protection areas becomes "access prohibited (0)". Therefore, if there are such memory protection settings, there may be a difference between the actual access result by PREPARE instruction, instruction fetch, etc. and the result of memory protection setting check function.
2. The memory protection check function makes the same judgement as the memory protection judgement that is applied to the actual memory accesses. For example, if the MPM.MPE bit is cleared (to 0), all check results are always set to "access permitted (1)". If the MPM.SVP bit is cleared (to 0), the bits MCR.SXE, SWE, and SRE are always set to "access permitted (1)". Or if the MPM.MPE bit is set (to 1) (MPM.SVP is assumed to be cleared) and the MPAT.E bit of all protection area settings is cleared (to 0), the MCR.UXE, UWE, and URE bits are always set to "access prohibited (0)".

(b) Checking Procedure

Set the base address (lower limit) of the target address area to the MCA register, the size of the target area to the MCS register, and the system protection identifier (SPID) of the source requesting the service to the MCI register, then use the LDSR instruction (r0 specification is recommended) to access the MCC register and execute a check. The results can be read from the MCR register by the STSR instruction.

CAUTION

If the specified area to be checked crosses 0000 0000_H or 7FFF FFFF_H, it is judged as an area setting error, and the MCR.OV bit is set to 1. This means that the MCR.OV bit must be checked before accessing the check results. Do not use the check result until it is confirmed that the result is not invalid (OV = 0).

(c) Sample Code

It is assumed that the memory protection setting check function will be used for the following operations.

```

_service_protection:
    ...
    MOV     ADDRESS, r10           // Store the start address of the area to be checked to r10
    MOV     SIZE, r11             // Store the size of the area to be checked to r11
    MOV     SPID, r12            // Store the system protection identifier to r12
    DI
    LDSR    r10, sr8, 5           // Set the address to MCA
    LDSR    r11, sr9, 5           // Set the size to MCS
    LDSR    r12, sr12, 5          // Set the system protection identifier to MCI
    LDSR    r0, sr10, 5           // Start checking with MCC
    STSR    sr11, r13, 5         // Get the results from MCR
    EI
    ANDI    0x0100, r13, r0
    BNZ     _overflow            // Processing of invalid input when OV = 1
    BR     _result_check         // Otherwise, result is checked

```

(d) Method of Calculating Address Areas

The address area to be checked is treated as follows and compared with the memory protection setting.

Lower-limit address of the area: MCA setting

Upper-limit address of the area: Value calculated from $MCA + (MCS - 1)$

In addition, when MCS is set to 0000 0000_H, the value of “MCS – 1” is treated as FFFF FFFF_H. Moreover, if the result of calculating an upper address exceeds FFFF FFFF_H, it will be calculated as a 33-bit value, but the most significant bit, which represents an overflow, is discarded.

When MCA is 0000 0000_H and MCS is 0000 0000_H, the upper-limit address is FFFF FFFF_H.

When MCA is FFFF FFFF_H and MCS is 0000 0002_H, the upper-limit address is 0000 0001_H.

(e) Checking Results of the Memory Protection Setting Checking Function

Table 3.108 lists the results of the memory protection setting checking function for various settings.

Table 3.108 List of Checking Results of Memory Protection Setting Checking Function

MPM Setting		Check Setting (MCA, MCS, MCI)		Operation in the Various Types of Protected Area						Final Result of Checking (in the MCR) ^{*2}		
				Protection Setting			Result of Checking for One Protected Area ^{*1}					
MPE	SVP	Address	MCI, MPIDn	E	Address	xMPIDn ^{*3}	SXE, SWE, SRE	UXE, UWE, URE	OV	SXE, SWE, SRE	UXE, UWE, URE	
0	—	—	—	—	—	—	—	—	0	All bits are set to 1	All bits are set to 1	
1	0	Spanning 0000 0000 _H	—	—	—	—	—	—	1	All bits are set to 1	All bits are set to 0	
		Spanning 7FFF FFFF _H	—	—	—	—	—	—	1	All bits are set to 1	All bits are set to 0	
		Other than above	*4	*4	*4	*4	All bits are set to 1	*4	0	All bits are set to 1	*4	
	1	Spanning 0000 0000 _H	—	—	—	—	—	—	—	1	All bits are set to 0	All bits are set to 0
			Spanning 7FFF FFFF _H	—	—	—	—	—	—	1	All bits are set to 0	All bits are set to 0
			Other than above	Mismatch	—	—	—	All bits are set to 0	All bits are set to 0	0	All bits are set to 0	All bits are set to 0
		Match	0	—	—	—	All bits are set to 0	All bits are set to 0	0	Results for each of the areas	Results for each of the areas	
			1	Not included	—	—	All bits are set to 0	All bits are set to 0	0	Results for each of the areas	Results for each of the areas	
				—	Disabled	—	—	All bits are set to 0	All bits are set to 0	0	Results for each of the areas	Results for each of the areas
1	Included	Enabled	—	—	Results corresponding to the setting	Results corresponding to the setting	0	Results for each of the areas	Results for each of the areas			

Note 1. For the MPM and other settings, this indicates the result produced by a single protection-setting area, but it cannot be directly monitored.

Note 2. The values to be stored in each bit of the MCR are the overall results for all areas covered by the given protection setting. “Results for each of the areas” will be enabled if the result of checking for even one area is enabled. However, for the value of the OV bit or all bits being set to 1 or 0, the result is as defined regardless of the result of each protection setting area.

Note 3. This indicates the result of enabling when the WG or RG bit is set (to 1), or the collective results of enabling or disabling by the individual WMPIDn and RMPIDn bits.

Note 4. When the MPM.MPE bit is set (to 1) and the MPM.SVP bit is cleared (0), access to all memory in supervisor mode is enabled. On the other hand, the judgment in user mode is different. The results obtained in accord with the settings for checking and protection are stored in the same way as the results when both the MPM.MPE and MPM.SVP bits are set (to 1).

3.2.5.2 Cache

This section describes the function to control the cache memory that is installed in this CPU.

In normal operation, the cache memory is used implicitly in instruction fetching by the CPU. However, if the setting is to be changed or the state of the cache memory is directly manipulated by the CACHE instruction, consistency with the normal operation of CPU must be maintained. In many cases, the hardware makes adjustments so that inconsistencies between the normal operation of the CPU and the manipulation of the cache do not arise. However, to make sure that the results of cache manipulation are the intended state, take care that the manipulation does not affect the implicit operation.

For a detailed description of the cache memory that is installed in the individual products, see **Section 3.8.9, Product Information of Cache Structure**.

(1) Features

A 16-KB and parity-based way select (PBS) 4-way set associative instruction cache is placed between the CPU and the code flash. The instruction cache and the code flash are connected to each other via a 256-bit dedicated bus to minimize penalties caused by a cache miss. A data buffer is also mounted between the CPU and the code flash to achieve high-speed data access. The 256-MB area from 0000 0000_H to 0FFF FFFF_H in the address space is intended for the instruction cache and data buffer.

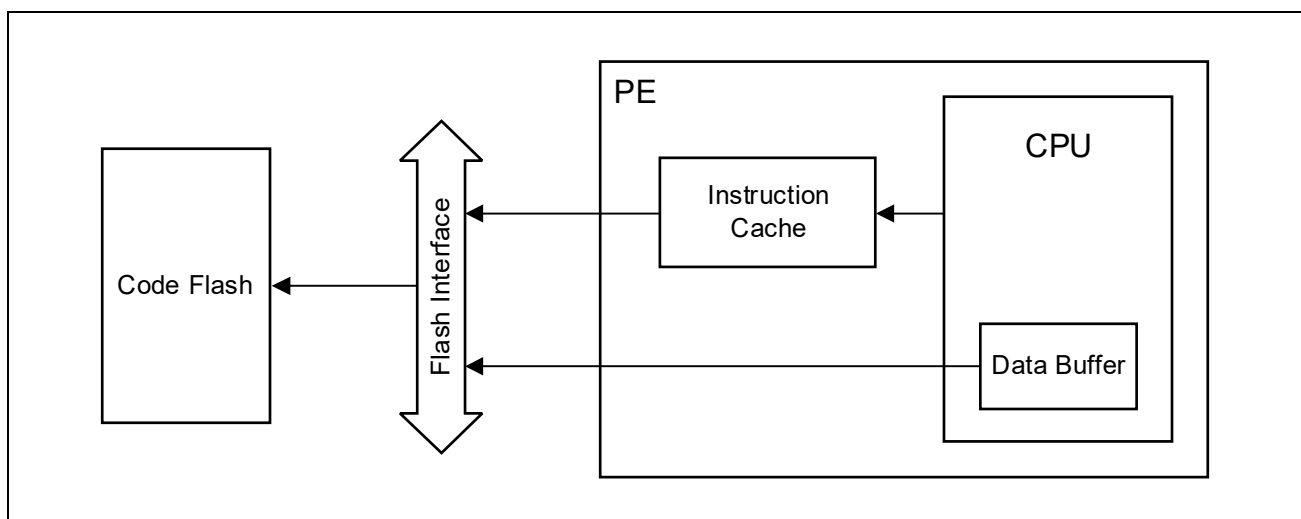


Figure 3.28 Instruction Cache and Data Buffer

(2) Cache Operation Registers

Figure 3.29 shows the system registers for cache operation. The supervisor privilege is required for the operation.

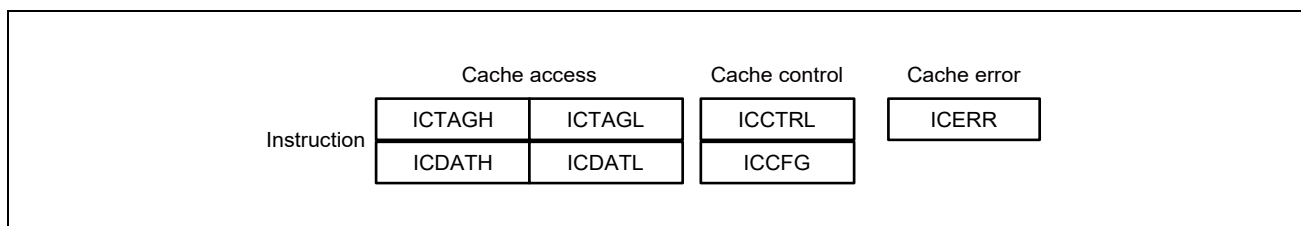


Figure 3.29 Cache Operation Registers

(3) Change Cache Use Mode

(a) Change Use Mode of Instruction Cache

The instruction cache use mode can be changed by using the ICCTRL.ICHEN bit. To enable an instruction cache, set the ICHEN bit to 1.

To disable the instruction cache, clear the ICHEN bit to 0.

Completion of the LDSR instruction that sets ICHEN might not coincide with completion of the instruction cache setting change. As in the following sample code, by executing the SYNCI instruction after the LDSR instruction, it is ensured that the change of the instruction cache setting takes effect after the SYNCI instruction.

```
LDSR r10, sr24, 4    // Change the instruction cache settings (ICCTRL)(setting value is stored in r10)
SYNCI                // Wait for completion of the LDSR instruction, then refetch the subsequent
                    // instructions.
```

(4) Cache Operations Using CACHE Instruction

The CACHE instruction manipulates the specified data in the cache memory.

Such data manipulation by the CACHE instruction starts after updating of the cache memory by all preceding memory access has been completed. Consequently, the result of preceding memory access is guaranteed to be the target for operations using the CACHE instruction. Additionally, a suitable synchronization operation is needed following execution of the CACHE instruction to ensure that the results are reflected in subsequent instructions.

The CACHE instruction can manipulate the cache memory even when the cache is disabled.

(a) Specification Method for Target of CACHE Instruction

There are basically two ways to specify the target for operations.

- Directly specify the address to be accessed:
In this CPU, this is called the address specification method. In this case, the cache line containing the specified address is subject to operation.
- Directly specify the cache memory's way number and line number:
In this CPU, this is called the index specification method. In this case, no hit judgment for the cache is performed, and the operation is performed on the specified cache index. For details about the cache index specification method, see **Section 3.2.5.2(6), Cache Index Specification Method**.

(b) Operations Performed using the CACHE Instruction

The operations performed on the cache memory are divided into the following. For details about each operation, see *the RH850G4MH User's Manual: Software*.

3.2.5.2.4.b.1 Cache Hit Block Invalidate / Cache Indexed Block Invalidate (CHBI / CIBI)

This disables the specified cache line. When using the address specification method, the cache line is disabled only when there is a hit. When using the index method, specified cache line is disabled. If the specified cache line is locked, it is unlocked. This operation can be used in cases such as when the entire memory cache is initialized by software.

Disabling a cache line refers to the process of clearing the cache information corresponding to the ICTAGL.V and ICTAGL.L bits (to 0).

3.2.5.2.4.b.2 Cache Fetch and Lock (CFAL)

This stores the data at the specified address to the cache memory. At this time, the cache line where the data is stored is locked. This prevents the cache line from being replaced. If the target cache line has already been stored in the cache memory, it is simply locked. If the target cache line has already been stored in the cache memory and is locked, this operation does nothing.

This operation can be used to improve execution efficiency by reducing variations in instruction execution time that occur due to cache misses in the specified memory area.

Cache lines are locked only when they are enabled. Although enabling and locking of a cache line are carried out simultaneously during the execution of the CFAL instruction, no cache line is locked when only locking is configured using the CIST instruction. For details, see **Section 3.2.3.7(1)(a), ICTAGL — Instruction Cache Tag Lo Access**.

CAUTION

In the instruction cache supported by this CPU, if all the cache lines at the same index in the target way group are locked, the cache lines are not replaced. Care must be taken with respect to the cache lock specifications and the number of cache ways when monopolizing the cache memory efficiently using this operation. For the way group, see **Section 3.2.5.2(10), Configuration of Instruction Cache**.

For details, see **Section 3.8.9, Product Information of Cache Structure**.

3.2.5.2.4.b.3 Cache Indexed Load / Cache Indexed Store (CILD / CIST)

This operation is used to directly access the cache memory. Values can be written and read, via a system register, at a position in the cache memory specified by using an index. Because cache data and cache tags can be accessed directly, this operation can be used for purposes such as software debugging.

(5) Cache Operation by the PREF Instruction

The PREF instruction is provided to realize efficient cache access by advising the CPU that an address is likely to be used in the near future. Getting the CPU to prefetch data into the cache memory before using the data can reduce the read wait time caused by a cache miss.

Assuming support by compilers and other tools, the PREF instruction can be executed regardless of the CPU operating mode. Execution of the PREF instruction does not cause an exception generated by the MPU, and has no effect on logical operations, just like the NOP instruction.

CAUTION

Because a data read request by the PREF instruction is rather speculative, it might not be executed depending on the system conditions. No cache fill is performed if all the cache lines at the same index in the target way group are locked. If an area outside the cacheable area is specified, the read itself is not carried out.

(6) Cache Index Specification Method

For a cache instruction that uses the index specification method, explicitly specify the cache memory subject to operation in the format shown in **Figure 3.30**, instead of specifying an address. The bit positions of each field depend on the size of the cache memory incorporated in the CPU core. Information about the incorporated cache memory and size can be read from the ICCFG register.

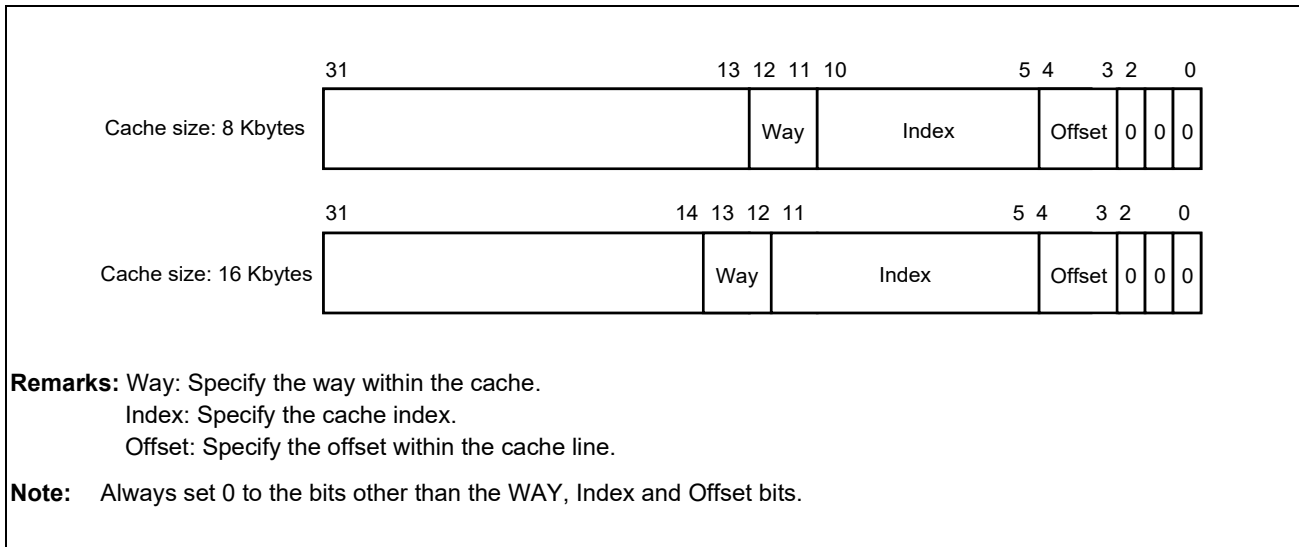


Figure 3.30 Cache Index Specification Method

NOTE

The Offset field indicates the byte position within the cache line. This setting is not required (i.e., ignored) in normal index specification operations. For a CILD/CIST operation, it is used to specify a position within the cache line because the ICDATH/ICDATL register is shorter than the cache line length.

(7) Execution Privilege of the CACHE/PREF Instruction

Because the CACHE instruction directly manipulates the contents of the cache memory, privileges are specified according to the type of operation. When the CACHE instruction is executed without the privilege required for the CACHE operation, a privilege instruction exception (PIE) occurs.

On the other hand, the PREF instruction provides information for speculative execution, so it can be executed in any mode.

The privileges required by the different operations performed by the CACHE instruction are shown below.

(a) Operations allowed with the user privilege

Among address specification method operations, operations without a cache lock (CHBI) can be executed in any operation mode.

(b) Operations requiring the supervisor privilege

Among address specification method operations, operations with a cache lock (CFAL) require the supervisor privilege.

In addition, index specification method operations also require the supervisor privilege.

(8) Memory Protection for the CACHE and PREF Instructions

The memory protection judgment for the CACHE and PREF instructions proceeds in accord with the operating mode when the instruction is executed. These instructions are handled as read access. When memory protection is enabled and the MPAT.SR or UR bit is set (to 1), the instruction can be executed according to the operating mode.

The CACHE instructions using the address specification mode are subject to the memory protection provided by the MPU. If a protection violation is detected, the cache operation is not executed and an MDP exception occurs.

On the other hand, since the CACHE instructions using the index specification method specify the position of data in the instruction cache but not the memory address, they are not subject to the memory protection. No matter what value is specified for the index, an MDP exception does not occur.

If a protection violation for a PREF instruction is detected, the prefetching is not performed and an MDP exception does not occur.

For both CACHE and PREF instructions, if operation is undefined for the specified opcode, cache operation and memory access are not performed at all, so they are not subject to the memory protection.

Table 3.109 shows the correspondence between operations and access permissions.

Table 3.109 Relationship between Cache Operations and Permissions

Instruction	Address/Index	Instruction Execution Privilege	Access Permission
CHBII	Address	UM	Read
CIBII	Index	SV	—
CFALI	Address	SV	Read
CISTI	Index	SV	—
CILDI	Index	SV	—
PREF	Address	UM	Read

(9) Example of Using the CACHE Instruction to Manipulate Cache Memory

This section gives an example of cache memory manipulation by the CACHE instruction, with setting up self-modifying code as the intention.

In the following example, an instruction code is generated in memory that is being cached in the instruction cache, and the instruction cache is disabled. Regarding whether memory that supports such cache memory manipulation is present, see **Section 3.8.9, Product Information of Cache Structure**.

```

    ST.W r20, 0[r10]      // Write the target data to memory
    SYNCM                // Wait for the completion of memory writing
    CACHE 0x00, [r10]    // CHBII: Disable caching of the target area by the instruction cache
    SYNCI                // Wait for the completion of instruction cache manipulation
    JMP [r12]            // Branch to the target area and execute the updated instruction code

```

(10) Configuration of Instruction Cache

This section describes the configuration of instruction cache mounted for this CPU. For more information, see **Section 3.8.9, Product Information of Cache Structure**.

This CPU supports an instruction cache in the following configuration.

- Four ways
- 256-bit cache line size (unit of filling)
- 256-Mbyte target area for caching
- ECC protection
- Cache error notification

(a) Four ways

Since the instruction cache mounted for this CPU is configured with four ways, the capacity of each way is one-quarter of the capacity of the whole cache.

The four ways are divided into two way groups; way 0 and way 1 belong to way group A, and way 2 and way 3 belong to way group B. As for the access address, if the parity of the bits to be stored in the tag is 0, way group A is used, whereas way group B is used if the parity is 1.

When storing data in a new cache line, the LRU (least recently used) method determines the destination way for storage from within the given way group.

(b) Cache line size

The cache line size (unit of filling) is the amount of data that is stored in one line of the cache, and is 256 bits in the instruction cache for this CPU.

(c) Allocation of cache data

Data from memory are allocated to the cache memory in the way shown in **Figure 3.31**.

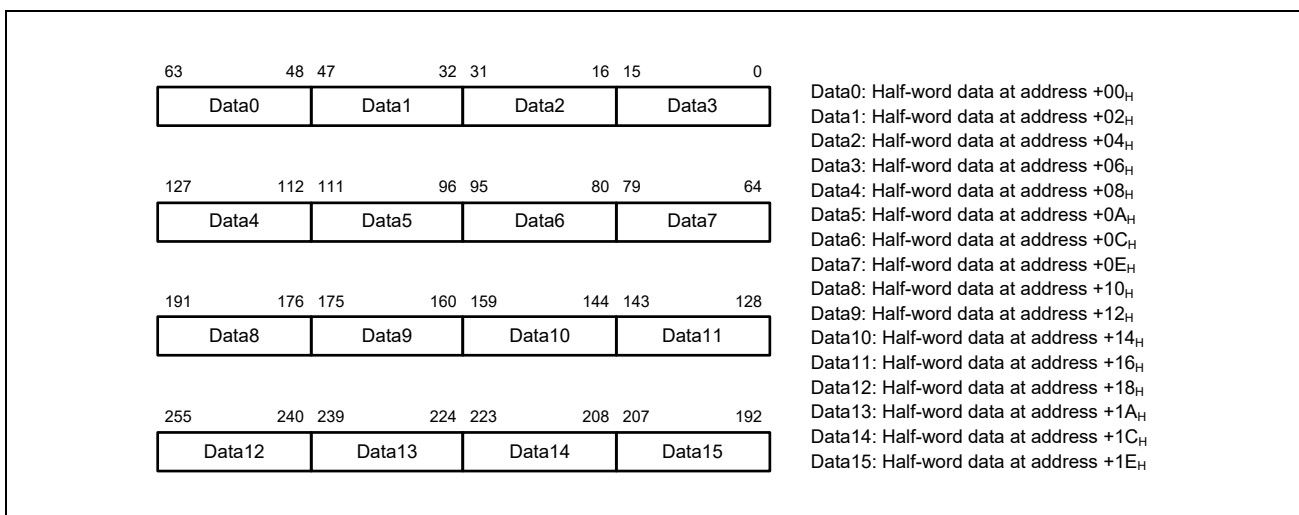


Figure 3.31 Allocation of Data to the Cache

(d) Cacheable area

The cacheable area for the instruction cache of this CPU is the 256 Mbytes from 0000 0000_H to 0FFF FFFF_H. Although instructions can be fetched from locations beyond the above area, they are never stored in the instruction cache in that case.

(e) ECC protection

The data RAM and the tag RAM of the instruction cache for this CPU are protected by ECCs. Upon detection of an ECC error in either area of RAM, the target line is disabled, and the CPU provides external notification of the occurrence of the error at the same time as re-fetching proceeds.

For details on the ECC protection, refer to **Section 38, Functional Safety**.

(f) Cache error notification

The instruction cache for this CPU detects the following five types of errors. A cache error is detected at the time of read operation by the instruction fetching or the CACHE instruction using the address specification method. In addition to the read operation, an address-feedback error is also detected at the time of write operation by cache fill. This CPU provides external notification of these errors. For the operation in cases of these errors occurring, refer to **Section 38, Functional Safety**.

- ECC error of data RAM
- ECC error of tag RAM
- Multi-hit error
- Way error
- Address-feedback error

A multi-hit error is detected when access to an address hits two ways in a way group at the same time. Upon detection of a multi-hit error, the target line is disabled, and the CPU provides external notification of the occurrence of the error at the same time as re-fetching proceeds.

A way error is detected when the parity in a way group does not match the parity of data stored in the tag RAM. Upon detection of a way error, the target line is disabled, and the CPU provides external notification of the occurrence of the error at the same time as re-fetching proceeds.

An address-feedback error is detected when the address for access to the RAM does not match the feedback address. When the error is detected at the time of cache reading, the target line is disabled, and the CPU provides external notification of the occurrence of the error at the same time as re-fetching proceeds. When the error is detected at the time of cache writing, the target line is disabled, and the CPU provides external notification of the occurrence of the error.

(11) Data Buffer Function

The four-line buffer with 256 bits per line is mounted as a data buffer. 256-bit data read from the code flash is stored in the data buffer in 256-bit units. When the data buffer hit occurs, the data is read out from the data buffer, so the code flash is not accessed again.

3.2.6 Coprocessor

3.2.6.1 Floating-Point Operation

The floating-point unit (FPU) operates as the CPU coprocessor, and executes floating-point instructions.

Either single-precision (32-bit) or double-precision (64-bit) data can be used. In addition, the conversion between a floating point type and an integer type is possible.

The FPU of this CPU conforms to ANSI/IEEE standard 754-2008 (IEEE Standard for Floating-Point Arithmetic).

(1) Configuration of Floating-Point Operation Function

(a) Not implemented

If the floating-point operation function is not implemented, all the floating-point instructions cannot be used. If an attempt is made to execute such an instruction, a coprocessor unusable exception occurs. In addition, all the floating-point system registers become inaccessible, and if an attempt is made to access any of the registers using LDSR/STSR instruction, a coprocessor unusable exception occurs.

(b) Implementing only single precision

If only the floating-point operation function with single precision is implemented, only floating-point instructions classified as single precision*¹ can be used. If an attempt is made to execute a floating-point instruction classified as double precision*², a coprocessor unusable exception occurs. All the floating-point system registers supply the function described in **Section 3.2.3.4, FPU Function Registers**.

Note 1. The single-precision floating-point instruction is the instruction described as (Single) in the description of each instruction in *the RH850G4MH User's Manual: Software*.

Note 2. The double-precision floating-point instruction is the instruction described as (Double) in the description of each instruction in *the RH850G4MH User's Manual: Software*.

(c) Implementing single precision and double precision

All the floating-point instructions can be used when floating-point instructions of single precision and double precision are implemented. All the floating-point system registers supply the functions described in **Section 3.2.3.4, FPU Function Registers**.

(2) Data Types

(a) Floating-point format

The FPU supports 32-bit (single precision) and 64-bit (double precision) IEEE754 floating-point operations.

The single-precision floating-point format consists of a 24-bit signed fraction (s + f) and an 8-bit exponent (e), as shown in **Figure 3.32**.

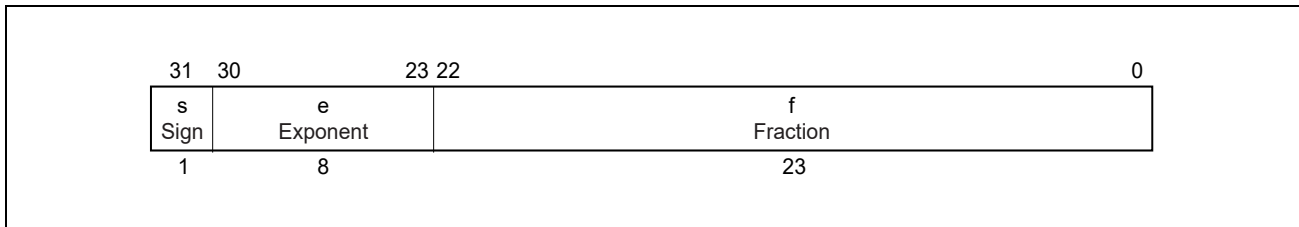


Figure 3.32 Single-Precision Floating-Point Format

The double-precision floating-point format consists of a 53-bit signed fraction (s + f) and an 11-bit exponent (e), as shown in **Figure 3.33**.

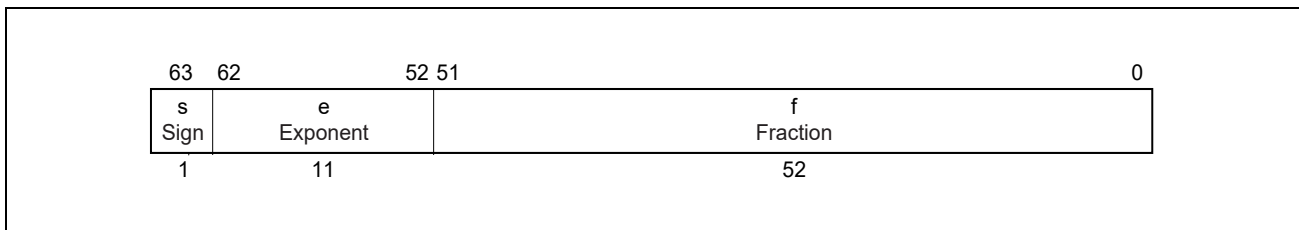


Figure 3.33 Double-precision Floating-Point Format

A numerical value in the floating-point format includes the following three areas.

- Sign bit: s
- Exponent: $e = E + \text{bias value}$
- Fraction: $f = .b_1b_2\dots b_{p-1}$ (value lower than the first decimal place)

The bias value for the single-precision format is 127. For double-precision format, the bias value is 1023.

The range of the exponent value E when unbiased covers all integers from E_{\min} to E_{\max} , along with two reserved values, $E_{\min} - 1$ (± 0 or subnormal number), and $E_{\max} + 1$ ($\pm \infty$ or NaN: not-a-number). A numeric value other than 0 is represented in one format, depending on the single-precision and double-precision formats.

The numeric value (v) represented in this format can be calculated by the expression shown in **Table 3.110**.

Table 3.110 Calculation Expression of Floating-Point Value

Type	Calculation Expression	
NaN (not-a-number)	If $E = E_{\max} + 1$ and $f \neq 0$	then $v = \text{NaN}$ regardless of s
$\pm \infty$ (infinite number)	If $E = E_{\max} + 1$ and $f = 0$	then $v = (-1)^s \infty$
Normalized number	If $E_{\min} \leq E \leq E_{\max}$	then $v = (-1)^s 2^E (1.f)$
Subnormal number	If $E = E_{\min} - 1$ and $f \neq 0$	then $v = (-1)^s 2^{E_{\min}} (0.f)$
± 0 (zero)	If $E = E_{\min} - 1$ and $f = 0$	then $v = (-1)^s 0$

- NaN (not-a-number)

IEEE754 defines a floating-point value called NaN (not-a-number). Because this value is not a numerical value, it does not have any “greater than” or “less than” relationships to other values.

If v is NaN in all of the floating-point formats, it might be either SignalingNaN (S-NaN) or QuietNaN (Q-NaN), depending on the value of the most significant bit of “f”. If the most significant bit of “f” is set, v is QuietNaN; if the most significant bit is cleared, it is SignalingNaN.

Table 3.111 shows the value of each parameter defined in floating-point formats.

Table 3.111 Floating-Point Formats and Parameter Values

Parameter	Format	
	Single Precision	Double Precision
E _{max}	+127	+1023
E _{min}	-126	-1022
Bias value of exponent	+127	+1023
Length of exponent (number of bits)	8	11
Integer bits	Cannot be seen	Cannot be seen
Length of fraction (number of bits)	23	52
Length of format (number of bits)	32	64

Table 3.112 shows the minimum and maximum values that can be represented in floating-point formats.

Table 3.112 Floating-Point Minimum and Maximum Values

Type	Value
Minimum value of single-precision floating point	1.40129846e - 45
Minimum value of single-precision floating point (normal)	1.17549435e - 38
Maximum value of single-precision floating point	3.40282347e + 38
Minimum value of double-precision floating point	4.9406564584124654e - 324
Minimum value of double-precision floating point (normal)	2.2250738585072014e - 308
Maximum value of double-precision floating point	1.7976931348623157e + 308

(b) Fixed-point formats

The value of a fixed point is held in the format of 2's complement. **Figure 3.34** shows a 32-bit fixed-point format and **Figure 3.35** shows a 64-bit fixed-point format. No signed bits exist in the unsigned fixed-point format, and all bits represent the integer value.

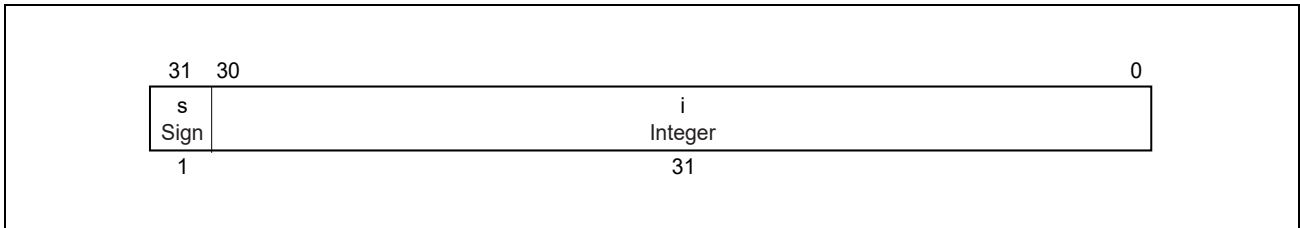


Figure 3.34 32-bit Fixed-Point Format

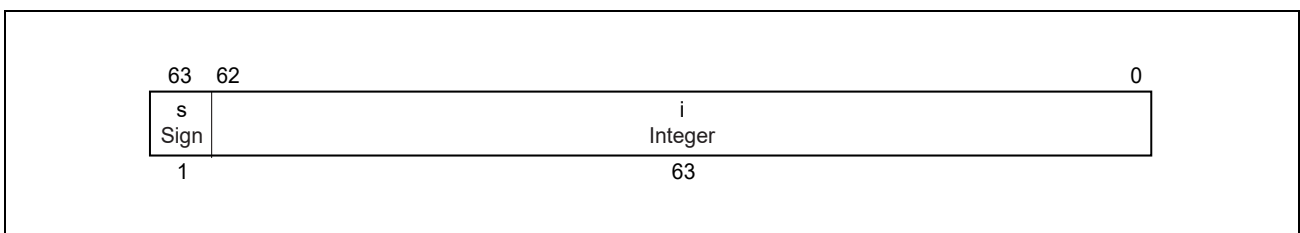


Figure 3.35 64-bit Fixed-Point Format

(c) Expanded floating-point format

This CPU supports the 16-bit (half-precision) IEEE754 floating-point format as a floating-point format for storing data. The half-precision floating-point format is used to decrease the amount of data; it is not supported for arithmetic operations. Instructions are available for converting single-precision floating-point format data into half-precision floating-point data and vice-versa. The half-precision floating-point format consists of an 11-bit signed fraction (s + f) and a 5-bit exponent (e), as shown in **Figure 3.36**.

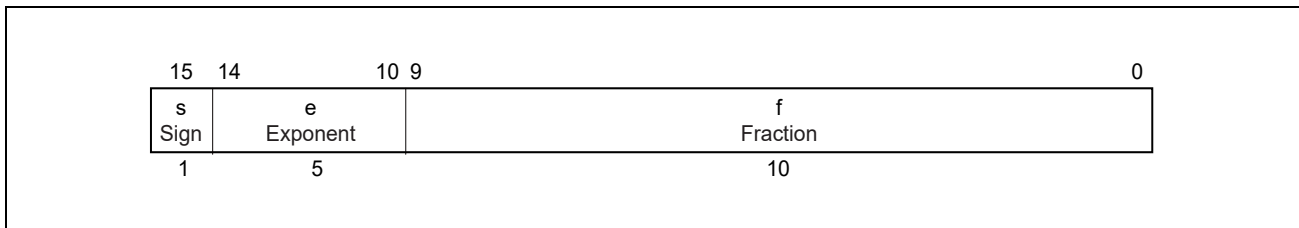


Figure 3.36 Half-Precision Floating-Point Format

Like other floating-point formats, the numeric values represented in this format can be calculated by using the expressions shown in **Table 3.110**. The values of the parameters defined by the half-precision floating-point format are shown in **Table 3.113**.

Table 3.113 Half-Precision Floating-Point Format and Parameter Values

Parameter	Half Precision
Emax	+15
Emin	-14
Bias value of exponent	+15
Length of exponent (number of bits)	5
Integer bits	Cannot be seen
Length of fraction (number of bits)	10
Length of format (number of bits)	16

Table 3.114 shows the minimum and maximum values that can be represented in the half-precision floating-point format.

Table 3.114 Half-Precision Floating-Point Minimum and Maximum Values

Type	Value
Minimum value of half-precision floating point	5.96046e - 8
Maximum value of half-precision floating point (normal)	6.10352e - 5
Maximum value of half-precision floating point	65504

(3) Register Set

For details about the register set, see **Section 3.2.3.4, FPU Function Registers**.

(4) Floating-Point Instructions

Floating-point instructions are divided into single-precision instructions (single) and double-precision instructions (double).

For details about the floating-point instructions, see *the RH850G4MH User's Manual: Software*.

(5) Floating-Point Operation Exceptions

This section describes how the FPU processes floating-point operation exceptions.

(a) Types of exceptions

When floating-point operations or processing of operation results cannot be done using the ordinary method, a floating-point operation exception occurs.

One of the following two operations is performed when a floating-point operation exception has occurred.

- When exceptions are enabled
The cause bit is set in the floating-point configuration/status register (FPSR), and processing (by software) is passed to the exception handler routine.
- When exceptions are prohibited
The preservation bit is set in the floating-point configuration/status register (FPSR), an appropriate value (initial value) is stored in the FPU destination register, then execution is continued.

The FPU uses cause bits, enable bits, and preservation bits (status flags) to support the following five types of IEEE754-defined exception causes.

- Inexact operation (I)
- Overflow (O)
- Underflow (U)
- Division by zero (Z)
- Invalid operation (V)

The sixth type of exception cause is unimplemented operation (E), which causes an exception when a floating-point operation cannot be executed. This exception requires processing by software. An unimplemented operation exception (E) occurs when exceptions are always enabled, rather than by using properties, enable bits, or preservation bits.

Figure 3.37 shows the FPSR register bits that are used to support exceptions.

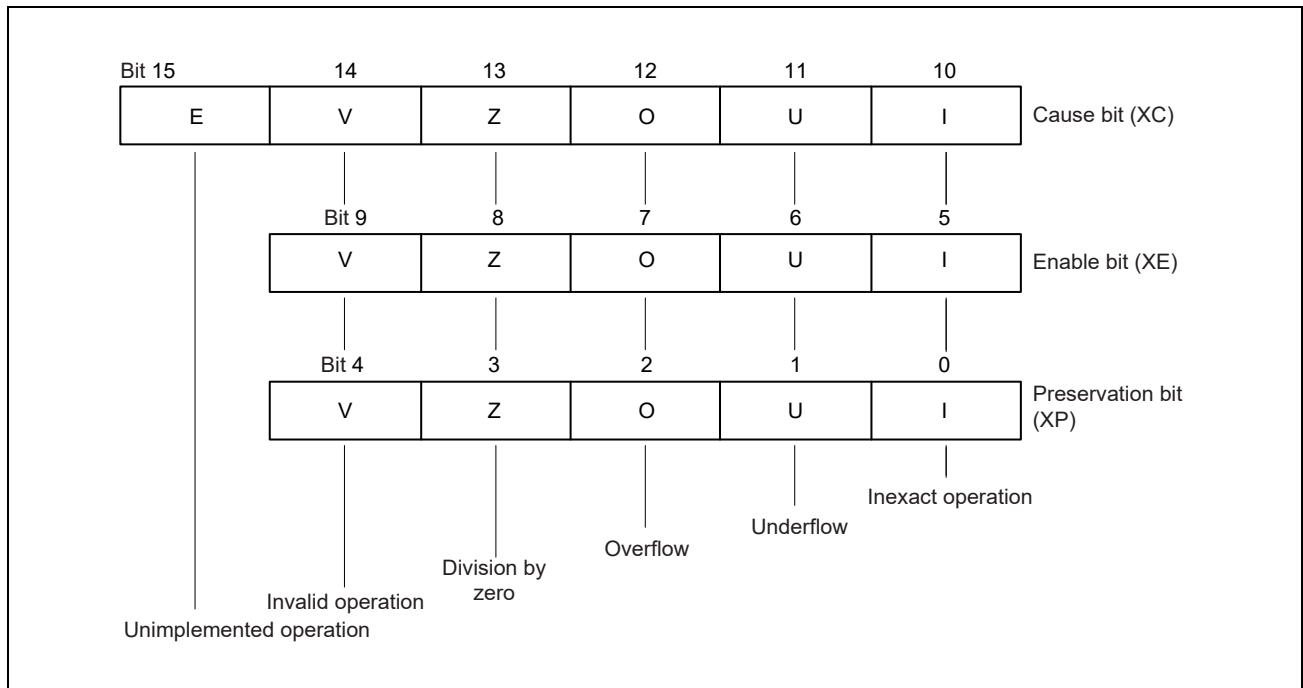


Figure 3.37 Cause, Enable, and Preservation Bits of FPSR Register

The five exceptions (V, Z, O, U, and I) defined by IEEE754 are enabled when the corresponding enable bits are set. When an exception occurs, if the corresponding enable bit has been set, the FPU sets the corresponding cause bit. If the exception can be acknowledged, processing is passed to the exception handler routine. If exceptions are prohibited, the exception corresponding preservation bit is set, and processing is not passed to the exception handler routine.

(b) Exception handling

When a floating-point operation exception occurs, the cause bits of the FPSR register indicate the cause of the floating-point operation exception.

3.2.6.1.5.b.1 Status flag

A corresponding preservation bit is available for each IEEE754-defined exception. The preservation bit is set when the corresponding exception is prohibited but the exception condition has been detected. The preservation bit is set or reset whenever new values are written to the FPSR register by the LDSR instruction.

If an exception is prohibited by an enable bit, predetermined processing is performed by the FPU. This processing provides an initial value as the result, rather than a floating-point operation result. This initial value is determined according to the type of exception. For an overflow exception or underflow exception, the initial value also differs depending on the current rounding mode. **Table 3.115** shows the initial values provided for each of the FPU IEEE754-defined exceptions.

Table 3.115 FPU Initial Values for IEEE754-Defined Exceptions

Area	Description	Rounding Mode	Initial Value
V	Invalid operation	—	Quiet not-a-number (Q-NaN)
Z	Division by zero	—	Correctly signed ∞
O	Overflow	RN	∞ with sign of intermediate result
		RZ	Maximum normalized number with sign of intermediate result
		RP	Negative overflow: Maximum negative normalized number Positive overflow: $+\infty$
		RM	Positive overflow: Maximum positive normalized number Negative overflow: $-\infty$
U	Underflow*1	RN*2	0 with sign of intermediate result
		RZ	0 with sign of intermediate result
		RP	Positive underflow: Minimum positive normalized number Negative underflow: 0
		RM	Negative underflow: Minimum negative normalized number Positive underflow: 0
I	Inexact operation	—	Rounded result

Note 1. If the FPSR.FS bit is cleared, an unimplemented operation exception (E) will occur if an underflow occurs in the rounded result; an underflow exception (U) will not occur. If the FS bit of the FPSR register is set, the flushed result is used as the initial value.

Note 2. If the rounding mode is RN and the FN bit of the FPSR register is set, flushing will occur in the direction of higher accuracy. For details, see **Section 3.2.6.1(9), Flush to Nearest**.

(6) Exception Details

The following describes the conditions under which each of the FPU exceptions occurs and the FPU responses.

(a) Inexact exception (I)

In the following cases, the FPU detects an inexact exception.

- When the precision of the rounded result is dropped
- When the rounded result overflows while overflow exceptions are prohibited
- When the rounded result underflows while underflow exceptions are prohibited
- When the operand that is a subnormal number is flushed, neither an invalid operation exception (V) nor a division by zero exception (Z) is detected, and the other operands are not Q-NaN

CAUTION

If the FS bit of the FPSR register is cleared and the operation result underflows, an unimplemented operation exception (E) occurs. In such cases, the underflow exception is not detected, so the inexact exception is not detected either.

3.2.6.1.6.a.1 If exception is enabled

The contents of the destination register are not changed, contents of the source register are saved, and an inexact exception occurs.

3.2.6.1.6.a.2 If exception is not enabled

If no other exception occurs, the rounded result or the result that underflows or overflows is stored in the destination register.

(b) Invalid operation exception (V)

An invalid operation exception occurs when either or both of the operands are invalid.

- Arithmetic operation with S-NaN included in operands. The conditional move instruction (CMOV), absolute value (ABS), and arithmetic negation (NEG) are not handled as arithmetic operations, but minimum value (MIN) and maximum value (MAX) are handled as arithmetic operations.
- Multiplication: $\pm 0 \times \pm\infty$ or $\pm\infty \times \pm 0$
- Fused-multiply-add: $(\pm 0 \times \pm\infty) + c$ or $(\pm\infty \times \pm 0) + c$. But only if c is not Q-NaN.
- Addition/subtraction or multiply-add operation*¹:
Addition of infinite values with different signs or subtraction of infinite values with the same sign
- Division: $\pm 0 \div \pm 0$ or $\pm\infty \div \pm\infty$
- Square root: When operand is less than 0
- Conversion to integer when source is outside of integer range.
- Comparison: With condition codes 8 to 15, if the operand is unordered (see *the RH850G4MH User's Manual: Software*)

Note 1. When the multiplication result is infinite or when adding or subtracting between infinities.

3.2.6.1.6.b.1 If exception is enabled

The contents of the destination register are not changed, contents of the source register are saved, and an invalid operation exception occurs.

3.2.6.1.6.b.2 If exception is not enabled

If no other exception occurs, and the destination is a floating-point format, Q-NaN is stored in the destination register. If the destination has an integer format, see the operation result description of each instruction for the value to be stored in the destination register.

(c) Division by zero exception (Z)

A division by zero exception occurs when a divisor is 0 and a dividend is a finite number other than 0.

3.2.6.1.6.c.1 If exception is enabled

The contents of the destination register are not changed, contents of the source register are saved, and a division by zero exception occurs.

3.2.6.1.6.c.2 If exception is not enabled

If no other exception occurs, a correctly signed infinite number ($\pm\infty$) is stored in the destination register.

(d) Overflow exception (O)

An overflow exception is detected if the exponent range is infinite and if the result of the rounded floating point is greater than maximum finite number in the destination format.

3.2.6.1.6.d.1 If exception is enabled

The contents of the destination register are not changed, the contents of the source register are saved, and an overflow exception occurs.

3.2.6.1.6.d.2 If exception is not enabled

If no other exception occurs, the initial value that is determined by the rounding mode and the sign of the intermediate result is stored in the destination register (see **Table 3.115, FPU Initial Values for IEEE754-Defined Exceptions**).

(e) Underflow exception (U)

If the operation result is $-2^{E_{min}}$ to $+2^{E_{min}}$ (but not zero), an underflow exception is detected.

Although IEEE754 defines several methods for detecting an underflow, the same method should be used to detect underflows, regardless of the processing to be performed.

The following two methods can be used to detect an underflow for binary floating point numbers.

- The result calculated after rounding and using an infinite exponent range is not zero and is within $\pm 2^{E_{min}}$.
- The result calculated before rounding and using an infinite exponent range and precision is not zero and is within $\pm 2^{E_{min}}$.

In this CPU, an underflow is detected before rounding. Or the rounded result is one of the following, an inexact result is detected.

- When a given result differs from the result calculated when the exponent range and precision are infinite

In this CPU, the behavior when an inexact result is detected differs as follows depending on whether underflow exceptions are enabled or disabled:

3.2.6.1.6.e.1 If exception is enabled

When the FS bit of the FPSR register has been set, if underflow exceptions are enabled, an underflow exception (U) occurs. When the FS bit of the FPSR register has been set, if underflow exceptions are not enabled but inexact exceptions are enabled, an inexact exception (I) occurs.

3.2.6.1.6.e.2 If exception is not enabled

If the FS bit of the FPSR register has been set, the initial value determined according to the rounding mode and intermediate result value is stored in the destination register (see **Table 3.115, FPU Initial Values for IEEE754-Defined Exceptions**).

CAUTION

If the FS bit of the FPSR register has not been set, an unimplemented operation exception (E) occurs regardless of whether or not exceptions are enabled. Because an unimplemented operation exception (E) must occur, an underflow exception (U) does not occur.

(f) Unimplemented operation exception (E)

The E bit is set and an unimplemented operation exception (E) occurs when an abnormal operand or abnormal result that cannot be correctly processed by hardware has been detected. The operand and destination register contents do not change.

If the FS bit of the FPSR register has been set, an unimplemented operation exception (E) will not occur.

If the FS bit of the FPSR register has been cleared, an unimplemented operation exception (E) will occur under the following conditions (except for CMOVF.D, CMOVF.S, CMPF.D, CMPF.S, CVTF.HS, ABSF.D, ABSF.S, MAXF.D, MAXF.S, MINF.D, MINF.S, NEGF.D, and NEGF.S instructions).

- When the operand is a subnormal number
- When the operation result is a subnormal number, or an underflow has occurred

CAUTION

If the FS bit of the FPSR register is set to 1, an unimplemented operation exception (E) will not occur under any circumstances.

(7) Saving and Returning Status

When a floating-point operation exception occurs, the PC and PSW are saved to the EIPC and EIPSW registers respectively, and the exception cause code is saved to the EIIC register.

The exception cause code of the floating-point operation exception is 71_H.

When an EI level exception is acknowledged while processing a floating-point operation exception, an EIPC register override occurs, which prevents the returning to the instruction that caused the floating-point operation exception to occur. When acknowledgment of EI level exceptions is required, the contents of the EIPC, EIPSW, and EIIC registers must be saved, such as to a stack.

When a floating-point instruction is used in a floating-point operation exception handler routine, the FPSR and FPEPC registers will be overridden if another floating-point operation exception occurs. In such cases, the FPSR and FPEPC registers should be saved at the start of the floating-point operation exception handler processing, and should be returned at the end of the handler processing.

The cause bits of the FPSR register hold the results from only one enabled exception. In any case, the previous results are held until the next enabled exception occurs.

(8) Flushing Subnormal Numbers

This CPU can process subnormal numbers—very small numbers that are lower than the minimum normalized number—in one of the following two ways:

- Normalize the operand or operation result and continue executing arithmetic processing
- Generate an unimplemented operation exception (E) and execute exception handling

Executing software-based exception handling will obtain a more accurate result, but the amount of time required to obtain the result will vary depending on the input value. In control systems that require a real-time performance, therefore, this is usually unacceptable. In this case, it is important to obtain the result within a certain amount time rather than focus on accuracy.

(a) Normalize the subnormal numbers and continue executing arithmetic processing

By setting the FS bit of the FPSR register to 1, this CPU can normalize the operand or operation result to a specific value and continue executing arithmetic processing if a subnormal number is input as the operand or obtained as the operation result. At this time, extremely small differences in values might not appear in the operation result.

For the operand and operation result, the values to which subnormal numbers are flushed when the FS bit is set (1) are shown in **Table 3.116** and **Table 3.117** below.

Table 3.116 Rounding Mode and Flush Value of Input Operand

Sign of Subnormal Operand	Rounding Mode and Value to Which Input Operand Is Flushed			
	RN	RZ	RP	RM
+			+0	
-			-0	

Table 3.117 Rounding Mode and Flush Value of Operation Result

Sign of Subnormal Operand Result	Rounding Mode and Value to Which Operation Result Is Flushed			
	RN*1	RZ	RP	RM
+	+0	+0	+2 ^{E_{min}}	+0
-	-0	-0	-0	-2 ^{E_{min}}

Note 1. If the rounding mode is RN and the FN bit of the FPSR register is set, flushing will occur in the direction of higher accuracy. For details, see **Section 3.2.6.1(9), Flush to Nearest**.

Whether an input operand that is a subnormal number has been flushed or not can be checked by referencing the IF bit of the FPSR register. Whether an operation result that is a subnormal number has been flushed or not can be checked by referencing the U bit of the FPSR register.

CAUTIONS

1. In control systems that require a real-time performance, it is recommended to always set the FS bit of the FPSR register to 1.
2. If the FS bit of the FPSR register is set to 1, an unimplemented operation exception (E) will not occur under any circumstances.
3. Whether the operation result is a subnormal number is judged by using the value before rounding.
4. The IF bit of the FPSR register also accumulates and indicates information about flushing instructions that have caused a floating-point operation exception.

(b) Generate an unimplemented operation exception (E) and execute exception handling

By clearing the FS bit of the FPSR register to 0, an unimplemented operation exception (E) will occur if a subnormal number is input as the operand or obtained as the operation result. When an unimplemented operation exception occurs, software-based progressive underflow processing is performed in the floating-point operation exception handling routine, enabling a more accurate result to be obtained. In this case, however, a real-time processing performance might not be realized due to the software processing load.

(c) Instructions that can handle subnormal numbers

The following instructions can be executed without causing an unimplemented operation exception (E) even if an operand that is a subnormal number is input while the FS bit of the FPSR register is 0.

- Conditional move instruction (CMOV), absolute value (ABS), arithmetic negation (NEG)
- Minimum value (MIN), maximum value (MAX), compare (CMPF)
- Conversion from half-precision to single-precision (CVTF.HS)

(d) Instructions that are not affected by flushing subnormal numbers

For the following instructions, flushing does not occur even an operand that is a subnormal number is input while the FS bit of the FPSR register is 1.

- Conditional move instruction (CMOV), absolute value (ABS), arithmetic negation (NEG)
- Minimum value (MIN), maximum value (MAX), compare (CMPF)
- Conversion from half-precision to single-precision (CVTF.HS)

(9) Flush to Nearest

This CPU provides flush-to-nearest mode, a feature for flushing to the nearest number with higher accuracy when a flushing operation results subnormal number. Flush-to-nearest mode is enabled when the rounding mode is RN and the FN bit of the FPSR register is set (1). When this mode is used, the FPU determines the value to which to flush the subnormal number based on the number of the operation result and not just the sign. This feature has no effect in rounding modes other than RN or on the result of flushing an input operand.

Table 3.118 Rounding Mode and Value to Which Operation Result is Flushed

Value of Subnormal Operation Result	Rounding Mode and Value to Which Operation Result Is Flushed				
	RN		RZ	RP	RM
	FN = 1	FN = 0			
$+2^{E_{min}-1} \leq \text{Operation result} < +2^{E_{min}}$	$+2^{E_{min}}$	+0	+0	$+2^{E_{min}}$	+0
$+0 < \text{Operation result} < +2^{E_{min}-1}$	+0				
$-2^{E_{min}-1} < \text{Operation result} < -0$	-0	-0	-0	-0	$-2^{E_{min}}$
$-2^{E_{min}} < \text{Operation result} \leq -2^{E_{min}-1}$	$-2^{E_{min}}$				

CAUTION

Whether the operation result is a subnormal number is judged by using the value before rounding.

(10) Limitation on the Floating-Point Operation Result

There is a limitation on the floating-point operation result of this CPU.

(a) Operation instruction result when the input data is NaN (not-a-number)

When the input data is NaN (not-a-number), the operation instruction result will be as shown in **Table 3.119**.

Table 3.119 Operation Result when Input data is NaN

Input Data*3	Single-precision Operation Result	Double-precision Operation Result
SNaN and QNaN mixed	7FFF FFFF _H	7FFF FFFF_FFFF FFFF _H
Only SNaN	7FFF FFFF _H	7FFF FFFF_FFFF FFFF _H
Only QNaN	*1	*1
SNaN and real number	7FFF FFFF _H	7FFF FFFF_FFFF FFFF _H
QNaN and real number	*1,*4	*1,*4
Other than above*2	7FFF FFFF _H	7FFF FFFF_FFFF FFFF _H

Note 1. If the input data contains QNaN, QNaN in the input data is selected as the operation result by the priority of reg2, reg1, and reg3.

Note 2. In this case, input data contains real numbers only, and the operation result is invalid.

Note 3. When the operand of an ABSF.S, ABSF.D, NEGF.S, or NEGF.D instruction is NaN, the result of the operation will be the value obtained by the operation on the sign bit alone.

Note 4. When one of the operands of an MAXF.S, MAXF.D, MINF.S, and MINF.D instruction is QNaN, the result of the operation will be whichever operand is a real number.

(b) Results of conversion instructions when the operand is NaN (not-a-number)

When the operand is NaN (not-a-number), the results of conversion are as shown in **Table 3.120**.

Table 3.120 Results of Conversion when the Operand is NaN (not-a-number)

Input Data	Output Format	Results of Conversion
Half-precision SNaN	Single precision	7FFF FFFF _H
Half-precision QNaN	Single precision	The QNaN value with the same sign. Bits 12 to 0 in the fractional part of the output value are 0.
Single-precision SNaN	Word (signed)	8000 0000 _H
	Word (unsigned)	0000 0000 _H
	Long (signed)	8000 0000_0000 0000 _H
	Long (unsigned)	0000 0000_0000 0000 _H
	Half precision	7FFF _H
	Double precision	7FFF FFFF_FFFF FFFF _H
Single-precision QNaN	Word (signed)	8000 0000 _H
	Word (unsigned)	0000 0000 _H
	Long (signed)	8000 0000_0000 0000 _H
	Long (unsigned)	0000 0000_0000 0000 _H
	Half precision	The QNaN value with the same sign. Bits 12 to 0 in the fractional part of the operand are truncated.
	Double precision	The QNaN value with the same sign. Bits 28 to 0 in the fractional part of the output value are 0.
Double-precision SNaN	Word (signed)	8000 0000 _H
	Word (unsigned)	0000 0000 _H
	Long (signed)	8000 0000_0000 0000 _H
	Long (unsigned)	0000 0000_0000 0000 _H
	Single precision	7FFF FFFF _H
	Double-precision QNaN	Word (signed)
	Word (unsigned)	0000 0000 _H
	Long (signed)	8000 0000_0000 0000 _H
	Long (unsigned)	0000 0000_0000 0000 _H
	Single precision	The QNaN value with the same sign. Bits 28 to 0 in the fractional part of the operand are truncated.

(c) When 0 comparison is performed by MAXF and MINF

When both reg1 and reg2 are either +0 or -0, the operation will be as follows.

MINF.S, MINF.D: The value of reg1 is stored in reg3.

MAXF.S, MAXF.D: The value of reg2 is stored in reg3.

(d) In the case of the precision instruction (RECIPF)

The operation results of RECIPF.S and RECIPF.D are the same as the operation results of 1/x by DIVF.S and DIVF.D.

3.2.6.2 Extended Floating-Point Operation

The extended floating-point operation unit (FXU) operates as a coprocessor of the CPU, and executes an extended floating-point operation instruction.

The execution of one extended floating-point operation instruction allows the execution of four single-precision floating-point (vector data) operations. This is an operation generally referred to as an SIMD (Single Instruction Multiple Data) operation.

Also, conversion is possible between a floating-point type and an integer type. In extended floating-point operation by this CPU, the floating-point is handled according to the ANSI/IEEE standard 754-2008 “IEEE Standard for Floating-point Arithmetic”.

The FXU consists of an extended floating-point operation block and vector registers where vector data is stored. There are 32 vector registers and the data width is 128 bits.

An extended floating-point operation instruction is executed by the extended floating-point operation block. The extended floating-point operation block performs 32-bit x 4 vector data operation on 128-bit vector registers. This is an operation method generally referred to as SIMD execution.

(1) Configuration of Extended Floating-Point Operation Function

(a) Not implemented

If the extended floating-point operation function is not implemented, any extended floating-point operation instruction cannot be used, and any attempt to execute an extended floating-point operation instruction will result in a coprocessor unusable exception. Also, any extended floating-point system register cannot be accessed, and any attempt to manipulate an extended floating-point system register by means of the LDSR/STSR instruction will result in a coprocessor unusable exception.

(b) Implemented

If the extended floating-point operation function is implemented, any extended floating-point operation instruction can be used. Also, all vector registers and extended floating-point system registers provide the functions described in **Section 3.2.3.5, FXU Function Registers**.

(2) Data Types

The FXU supports the single-precision floating-point formats specified by IEEE 754. For details of the single-precision floating-point formats, see **Section 3.2.6.1(2), Data Types**.

The FXU does not support the double-precision floating-point formats.

(3) Register Set

For details of the register set, see **Section 3.2.3.5, FXU Function Registers**.

(4) Extended Floating-Point Operation Instructions

Extended floating-point operation instructions consist of extended floating-point vector operation instructions and extended floating-point vector manipulation instructions. For details of the extended floating-point operation instructions, see *the RH850G4MH User's Manual: Software*.

(5) Extended Floating-Point Operation Exceptions

This section describes how the FXU handles extended floating-point operation exceptions.

(a) Types of Exceptions

When it becomes impossible to perform floating-point operation or handle operation results on any operation way during the execution of an extended floating-point operation instruction, an extended floating-point operation exception occurs.

One of the following two operations is performed when an extended floating-point operation exception has occurred.

- When exceptions are enabled

The exception cause of each operation way is set in the FXXC register, and the logical OR of the extended floating-point operation exception that has occurred on each operation way is saved into the cause bits of the extended floating-point configuration/status register FXSR.

Control (software control) is transferred to the exception handler routine.

- When exceptions are disabled

The exception cause of each operation way is set in the FXXP register, and the logical OR of the extended floating-point operation exception that has occurred on each operation way is saved into the preservation bits of the extended floating-point configuration/status register FXSR.

An appropriate value (initial value) is stored in the destination register for an instruction that has caused an exception, and execution is continued.

The FXU uses cause bits, enable bits, and preservation bits (status flags) to support the following five types of IEEE754-defined exception causes.

- Inexact operation (I)
- Overflow (O)
- Underflow (U)
- Division by zero (Z)
- Invalid operation (V)

The sixth type of exception cause is unimplemented operation (E), which causes an exception when an extended floating-point operation cannot be executed. This exception requires processing by software. An unimplemented operation exception (E) occurs when exceptions are always enabled, rather than by using properties, enable bits, or preservation bits.

Figure 3.38 shows the relationship among the enable bits and the cause bits of the FXSR register and the cause bits of the FXXC register; these bits are used to support exceptions. **Figure 3.39** shows the relationship among the enable bits and the preservation bits of the FXSR register and the preservation bits of the FXXP register.

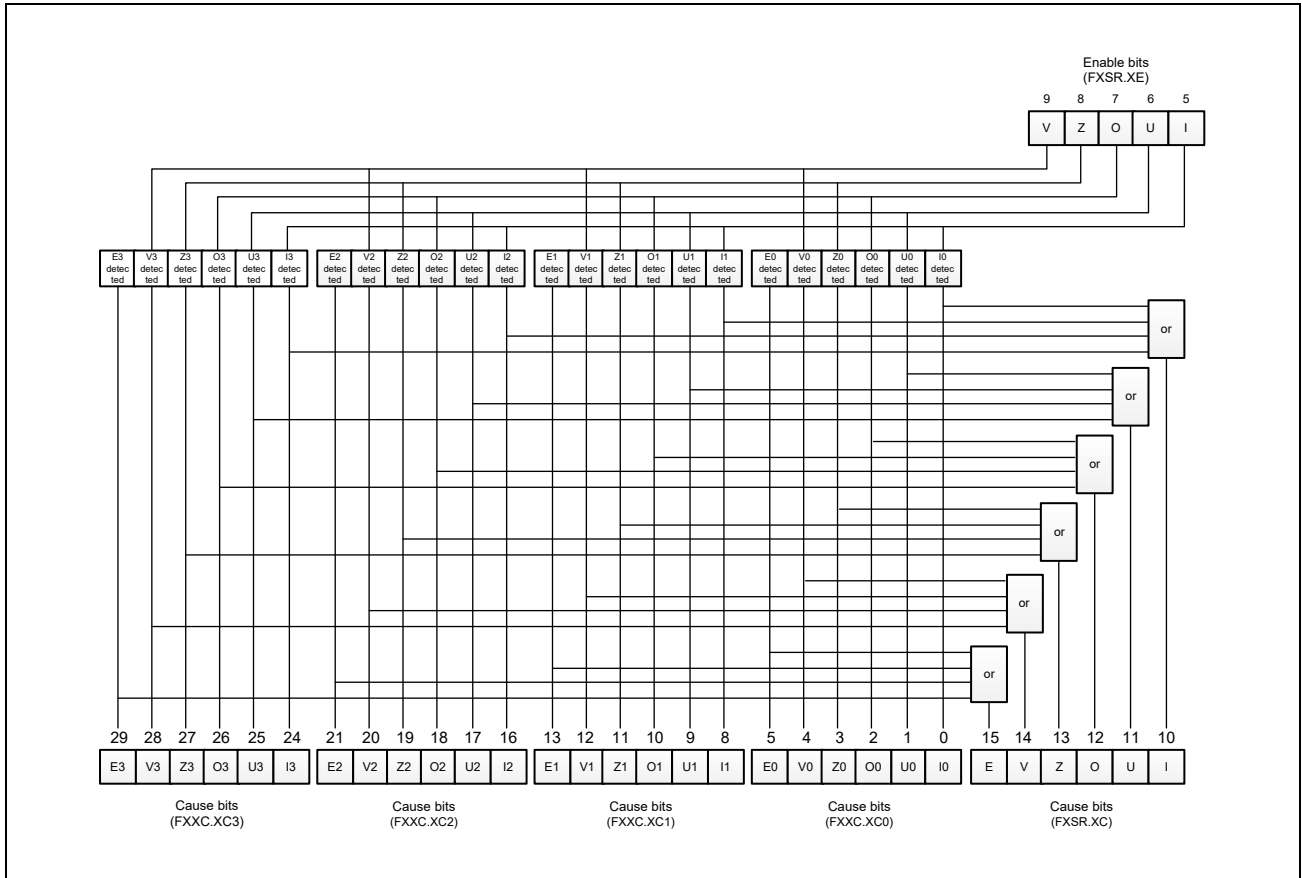


Figure 3.38 Relationship among the Cause and Enable Bits of the FXSR and FXXC Registers

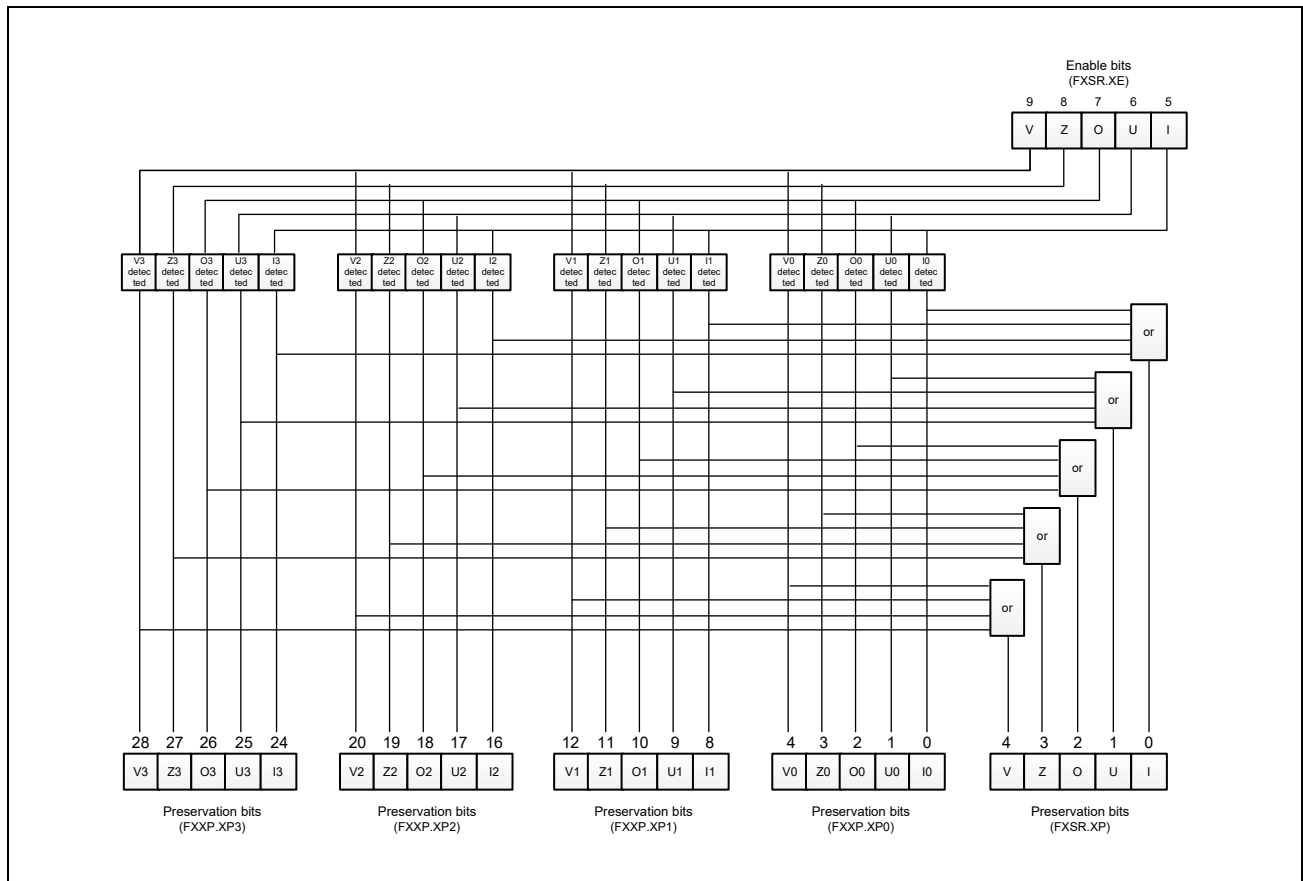


Figure 3.39 Relationship among the Preservation and Enable Bits of the FXSR and FXXP Registers

The five exceptions (V, Z, O, U, and I) defined by IEEE754 are enabled when the corresponding enable bits are set. All operation ways share the enable bits of the FXSR register, but detection of an exception is performed for each operation way. Therefore, different exceptions may be detected on multiple operation ways. In this case, one extended floating-point operation exception occurs.

If an exception has been detected on an operation way with its corresponding enable bit set, the FXU sets the cause bit of the operation way in the FXXC register. Also, it obtains the logical OR of the cause bits of all operation ways and sets the corresponding cause bits in the FXSR register. If the exception can be acknowledged, control is transferred to the exception handler routine.

If exceptions are prohibited, the preservation bit of that operation way in the FXXP register is set. Also, the logical OR of the preservation bits of all operation ways is obtained to set the corresponding preservation bits in the FXSR register. In this case, control is not transferred to the exception handler routine.

(b) Exception handling

When an extended floating-point operation exception occurs, the cause bits of the FXSR register indicate the cause of the extended floating-point operation exception.

3.2.6.2.5.b.1 Status flag

A corresponding preservation bit is available for each IEEE754-defined exception. The preservation bit is set when the corresponding exception is prohibited but the exception condition has been detected. The preservation bit is set or reset whenever new values are written to the FXSR register by the LDSR instruction.

If an exception is prohibited by an enable bit, predetermined processing is performed by the FXU. This processing provides an initial value as the result, rather than a floating-point operation result. This initial value is determined according to the type of exception. For an overflow exception or underflow exception, the initial value also differs depending on the current rounding mode. **Table 3.121** shows the initial values provided for each of the IEEE754-defined exceptions.

Note that initial values are stored only for operation ways where an exception has been detected. If different exceptions have been detected on multiple operation ways, the initial value corresponding to each exception is stored in the destination of each operation way.

Table 3.121 FXU Initial Values for IEEE754 Defined Exceptions

Area	Description	Rounding Mode	Initial Value
V	Invalid operation	—	Quiet not-a-number (Q-NaN)
Z	Division by zero	—	Correctly signed ∞
O	Overflow	RN	∞ with sign of intermediate result
		RZ	Maximum normalized number with sign of intermediate result
		RP	Negative overflow: Maximum negative normalized number Positive overflow: $+\infty$
		RM	Positive overflow: Maximum positive normalized number Negative overflow: $-\infty$
U	Underflow*1	RN*2	0 with sign of intermediate result
		RZ	0 with sign of intermediate result
		RP	Positive underflow: Minimum positive normalized number Negative underflow: 0
		RM	Negative underflow: Minimum negative normalized number Positive underflow: 0
I	Inexact operation	—	Rounded result

Note 1. If the FXSR.FS bit is cleared, an underflow in the rounded result will cause an unimplemented operation exception (E); therefore, in this case, an underflow exception (U) will not occur. If the FXSR.FS bit is set, the flushed result is used as the initial value.

Note 2. If the rounding mode is RN and FXSR.FN is set, flushing will occur in the direction of higher accuracy. For details, see **Section 3.2.6.1(9), Flush to Nearest**.

(6) Exception Details

The following describes the conditions under which each extended floating-point operation exception occurs and how the FXU handles it.

(a) Inexact exception (I)

If any of the following conditions is detected on any operation way, the FXU detects an inexact exception.

- When the precision of the rounded result is dropped
- When the rounded result overflows while overflow exceptions are prohibited
- When the rounded result underflows while underflow exceptions are prohibited
- When the operand that is a subnormal number is flushed, neither an invalid operation exception (V) nor a division by zero exception (Z) is detected, and the other operands are not Q-NaN

CAUTION

If the FXSR.FS bit is cleared and the operation result underflows, an unimplemented operation exception (E) occurs. In such cases, the underflow exception is not detected, so the inexact exception is not detected either.

3.2.6.2.6.a.1 If exception is enabled

The contents of the destination register are not changed, the contents of the source register are saved, and an inexact exception occurs.

3.2.6.2.6.a.2 If exception is not enabled

If no other exception occurs, the rounded result or the result that underflows or overflows is stored in the destination register.

(b) Invalid operation exception (V)

If either or both of the operands are invalid on any operation way, an invalid operation exception is detected.

- Arithmetic operation with S-NaN included in operands. The conditional move instruction (CMOV), absolute value (ABS), and arithmetic negation (NEG) are not handled as arithmetic operations, but minimum value (MIN) and maximum value (MAX) are handled as arithmetic operations.
- Multiplication: $\pm 0 \times \pm \infty$ or $\pm \infty \times \pm 0$
- Fused-multiply-add: $(\pm 0 \times \pm \infty) + c$ or $(\pm \infty \times \pm 0) + c$. But only if c is not Q-NaN.
- Addition/subtraction or multiply-add operation*¹:
Addition of infinite values with different signs or subtraction of infinite values with the same sign.
- Division: $\pm 0 \div \pm 0$ or $\pm \infty \div \pm \infty$
- Square root: When operand is less than 0.
- Conversion to integer when source is outside of integer range.
- Comparison: With condition codes 8 to 15, if the operand is unordered (see *the RH850G4MH User's Manual: Software*)

Note 1. When the multiplication result is infinite or when adding or subtracting between infinities.

3.2.6.2.6.b.1 If exception is enabled

The contents of the destination register are not changed, the contents of the source register are saved, and an invalid operation exception occurs.

3.2.6.2.6.b.2 If exception is not enabled

If no other exception occurs, and the destination is a floating-point format, Q-NaN is stored in the destination register. If the destination has an integer format, see the operation result description of each instruction for the value to be stored in the destination register.

(c) Division by zero exception (Z)

A division by zero exception occurs if, on any operation way, a divisor is 0 and a dividend is a finite number other than 0.

3.2.6.2.6.c.1 If exception is enabled

The contents of the destination register are not changed, the contents of the source register are saved, and a division by zero exception occurs.

3.2.6.2.6.c.2 If exception is not enabled

If no other exception occurs, a correctly signed infinite number ($\pm \infty$) is stored in the destination register.

(d) Overflow exception (O)

An overflow exception is detected if, on any operation way, the exponent range is infinite and if the result of the rounded floating-point is greater than maximum finite number in the destination format.

3.2.6.2.6.d.1 If exception is enabled

The contents of the destination register are not changed, the contents of the source register are saved, and an overflow exception occurs.

3.2.6.2.6.d.2 If exception is not enabled

If no other exception occurs, the initial value that is determined by the rounding mode and the sign of the intermediate result is stored in the destination register (see **Table 3.121, FXU Initial Values for IEEE754 Defined Exceptions**).

(e) Underflow exception (U)

If, on any operation way, the operation result is $-2^{E_{min}}$ to $+2^{E_{min}}$ (but not zero), an underflow exception is detected.

Although IEEE754 defines several methods for detecting an underflow, the same method should be used to detect underflows, regardless of the processing to be performed.

The following two methods can be used to detect an underflow for binary floating-point numbers.

- The result calculated after rounding and using an infinite exponent range is not zero and is within $\pm 2^{E_{min}}$
- The result calculated before rounding and using an infinite exponent range and precision is not zero and is within $\pm 2^{E_{min}}$.

In this CPU, an underflow is detected before rounding.

Also, if the rounded result is one of the following, an inexact result is detected.

- When a given result differs from the result calculated when the exponent range and precision are infinite

In this CPU, the behavior when an inexact result is detected differs as follows depending on whether underflow exceptions are enabled or disabled.

3.2.6.2.6.e.1 If exception is enabled

When the FXSR.FS bit has been set, if underflow exceptions are enabled, an underflow exception (U) occurs. When the FXSR.FS bit has been set, if underflow exceptions are not enabled but inexact exceptions are enabled, an inexact exception (I) occurs.

3.2.6.2.6.e.2 If exception is not enabled

If the FXSR.FS bit has been set, the initial value determined according to the rounding mode and intermediate result value is stored in the destination register (see **Table 3.121 FXU Initial Values for IEEE754 Defined Exceptions**).

CAUTION

If the FXSR.FS bit has not been set, an unimplemented operation exception (E) occurs regardless of whether or not exceptions are enabled. Because an unimplemented operation exception (E) must occur, an underflow exception (U) does not occur.

(f) Unimplemented operation exception (E)

The E bit is set and an unimplemented operation exception (E) occurs when, on any operation way, an abnormal operand or abnormal result that cannot be correctly processed by hardware has been detected. The operand and destination register contents do not change.

If the FXSR.FS bit has been set, an unimplemented operation exception (E) will not occur.

If the FXSR.FS has been cleared, an unimplemented operation exception (E) will occur under the following conditions (except for ABSF.S4, CMOVF.W4, CMPF.S4, CVTF.HS4, MAXF.S4, MAXRF.S4, MINF.S4, MINRF.S4, and NEGF.S4 instructions).

- When the operand is a subnormal number
- When the operation result is a subnormal number, or an underflow has occurred

CAUTION

If the FXSR.FS bit has not been set, an unimplemented operation exception (E) occurs regardless of whether or not exceptions are enabled. Because an unimplemented operation exception (E) must occur, an underflow exception (U) does not occur.

(7) Saving and Restoring Status

When an extended floating-point operation exception occurs, the PC and PSW are saved to the EIPC and EIPSW registers respectively, and the exception cause code is saved to the EIIC register.

The exception cause code of the extended floating-point operation exception is 75_H.

When an EI level exception is acknowledged while processing an extended floating-point operation exception, an EIPC register override occurs, which prevents the returning to the instruction that caused the extended floating-point operation exception to occur. Therefore, when acknowledgment of EI level exceptions is required, the contents of the EIPC, EIPSW, and EIIC registers must be saved in advance into a stack or the like.

When an extended floating-point operation instruction is used in an extended floating-point operation exception handler routine, the FXSR and FXXC registers will be overridden if another floating-point operation exception occurs. In such cases, the FXSR and FXXC registers should be saved at the start of the extended floating-point operation exception handler processing, and should be restored at the end of the handler processing.

The cause bits of the FXSR register hold the results from only one enabled exception. In any case, the previous results are held until the next enabled exception occurs.

(8) Flushing Subnormal Numbers

This CPU can process subnormal numbers—very small numbers that are lower than the minimum normalized number—in one of the following two ways:

- Normalize the operand or operation result and continue executing arithmetic processing
- Generate an unimplemented operation exception (E) and execute exception handling

Executing software-based exception handling will obtain a more accurate result, but the amount of time required to obtain the result will vary depending on the input value. In control systems that require a real-time performance, therefore, this is usually unacceptable. In this case, it is important to obtain the result within a certain amount of time rather than focus on accuracy.

(a) Normalize the subnormal numbers and continue executing arithmetic processing

By setting the FS bit of the FXSR register to 1, this CPU can normalize the operand or operation result to a specific value and continue executing arithmetic processing even if a subnormal number is input as the operand or obtained as the operation result. At this time, extremely small differences in values might not appear in the operation result.

For the operand and operation result, the values to which subnormal numbers are flushed when the FS bit is set to 1 are shown in **Table 3.122** and **Table 3.123** below.

Table 3.122 Rounding Mode and Flush Value of Input Operand

Sign of Subnormal Operand	Rounding Mode and Value to Which Input Operand is Flushed			
	RN	RZ	RN	RM
+			+0	
–			–0	

Table 3.123 Rounding Mode and Flush Value of Operation result

Sign of Subnormal Operation Result	Rounding Mode and Value to Which Operation Result is Flushed			
	RN*1	RZ	RN*1	RM
+	+0	+0	+2 ^{E_{min}}	+0
–	–0	–0	–0	–2 ^{E_{min}}

Note 1. If the rounding mode is RN and FXSR.FN is set, flushing will occur in the direction of higher accuracy. For details, see **Section 3.2.6.1(9), Flush to Nearest**.

Whether an input operand that is a subnormal number has been flushed or not can be checked by referencing the IF bit of the FXSR register. Whether an operation result that is a subnormal number has been flushed or not can be checked by referencing the U bit of the FXSR register.

CAUTIONS

1. In control systems that require a real-time performance, it is recommended to always set the FS bit of the FXSR register to 1.
2. If the FS bit of the FXSR register is set to 1, an unimplemented operation exception (E) will not occur under any circumstances.
3. Whether the operation result is a subnormal number is judged by using the value before rounding.
4. The IF bit of the FXSR register also accumulates and indicates information about flushing instructions that have caused an extended floating-point operation exception.

(b) Generate an unimplemented operation exception (E) and execute exception handling

Clearing the FS bit of the FXSR register to 0 will cause an unimplemented operation exception (E) if a subnormal number is input as the operand or obtained as the operation result. When an unimplemented operation exception occurs, software-based progressive underflow processing is performed in the floating-point operation exception handling routine, enabling a more accurate result to be obtained. In this case, however, a real-time processing performance might not be ensured depending on the software processing load.

(c) Instructions that can handle subnormal numbers

The following instructions can be executed without causing an unimplemented operation exception (E) even if an operand that is a subnormal number is input while the FS bit of the FXSR register is 0.

- Conditional move instruction (CMOV), absolute value (ABS), arithmetic negation (NEG)
- Minimum value (MIN), maximum value (MAX), compare (CMPF)
- Conversion from half-precision to single-precision (CVTF.HS4)

(d) Instructions that are not affected by flushing subnormal numbers

For the following instructions, flushing does not occur even if an operand that is a subnormal number is input while the FS bit of the FXSR register is 1.

- Conditional move instruction (CMOV), absolute value (ABS), arithmetic negation (NEG)
- Minimum value (MIN), maximum value (MAX), compare (CMPF)
- Conversion from half-precision to single-precision (CVTF.HS4)

(9) Flush to Nearest

This CPU provides flush-to-nearest mode, a feature for flushing to the nearest number with higher accuracy when a flushing operation results subnormal number. Flush-to-nearest mode is enabled when the rounding mode is RN and FXSR.FN is set to 1. When this mode is used, the FXU determines the value to which to flush the subnormal number based on the number of the operation result and not just the sign. This feature has no effect in rounding modes other than RN or on the result of flushing an input operand.

Table 3.124 Rounding Mode and Value to Which Operation Result is Flushed

Value of Subnormal Operation Result	Rounding Mode and Value to Which Operation Result is Flushed				
	RN		RZ	RP	RM
	FN = 1	FN = 0			
$+2^{E_{min}-1} \leq \text{Operation result} < +2^{E_{min}}$	$+2^{E_{min}}$	+0	+0	$+2^{E_{min}}$	+0
$+0 < \text{Operation result} < +2^{E_{min}-1}$	+0				
$-2^{E_{min}-1} < \text{Operation result} < -0$	-0	-0	-0	-0	$-2^{E_{min}}$
$-2^{E_{min}} < \text{Operation result} \leq -2^{E_{min}-1}$	$-2^{E_{min}}$				

CAUTION

Whether the operation result is a subnormal number is judged by using the value before rounding.

(10) Limitation on the Floating-Point Operation Result

There is a limitation on the floating-point operation result of this CPU.

(a) Operation instruction result when the input data is NaN (not-a-number)

When the input data is NaN (not-a-number), the operation instruction result will be as shown in **Table 3.125**.

Table 3.125 Operation Result when Input data is NaN

Input Data*3	Operation Result
SNaN and QNaN mixed	7FFF FFFF _H
Only SNaN	7FFF FFFF _H
Only QNaN	*1,*4
SNaN and real number	7FFF FFFF _H
QNaN and real number	*1,*4
Other than above*2	7FFF FFFF _H

Note 1. If the input data contains QNaN, QNaN in the input data is selected as the operation result by the priority of wreg2, wreg1, and wreg3.

Note 2. In this case, input data contains real numbers only, and the operation result is invalid.

Note 3. When the operand of an ABSF.S4 or NEGF.S4 instruction is NaN, the result of the operation will be the value obtained by the operation on the sign bit alone.

Note 4. When one of the operands of a MAXF.S4, MAXRF.S4, MINF.S4, and MINRF.S4 instruction is QNaN, the result of the operation will be whichever operand is a real number.

(b) Results of conversion instructions when the operand is NaN (not-a-number)

When the operand is NaN (not-a-number), the results of conversion are as shown in **Table 3.126**.

Table 3.126 Results of Conversion when the Operand is NaN (not-a-number)

Input Data	Output Format	Results of Conversion
Half-precision SNaN	Single precision	7FFF FFFF _H
Half-precision QNaN	Single precision	The QNaN value with the same sign. Bits 12 to 0 in the fractional part of the output value are 0.
Single-precision SNaN	Word (signed)	8000 0000 _H
	Word (unsigned)	0000 0000 _H
	Half precision	7FFF _H
Single-precision QNaN	Word (signed)	8000 0000 _H
	Word (unsigned)	0000 0000 _H
	Half precision	The QNaN value with the same sign. Bits 12 to 0 in the fractional part of the operand are truncated.

(c) When 0 comparison is performed by MAXF and MINF

When wreg1 and wreg2 elements are either +0 or -0, the operation will be as follows.

MINF.S4: The values of wreg1 elements are stored in wreg3 elements.

MAXF.S4: The values of wreg2 elements are stored in wreg3 elements.

MINRF.S4: The values of wreg1 and wreg2 even elements are stored in wreg3 elements.

MAXRF.S4: The values of wreg1 and wreg2 odd elements are stored in wreg3 elements.

(d) In the case of the precision instruction (RECI PF)

The operation result of RECI PF.S4 is the same as the operation result of 1/x by DIVF.S.

3.2.7 Hazard Control

3.2.7.1 Synchronization Processing

In order to increase the processing performance, this CPU executes subsequent instructions without waiting for completion of the processing of preceding instructions, as long as there is no dependence relationship between a preceding instruction and a subsequent instruction. For example, when a store instruction is to update the value of a control register of a peripheral device which operates more slowly than the CPU, even when updating of the register is not completed while the bus system is still handling processing to store the value, the execution of a subsequent instruction will proceed. For this reason, if a subsequent instruction should certainly be made to wait for completion of processing of a preceding instruction by software, procedures for synchronization processing are required.

Moreover, when a single CPU is executing multiple load and store instructions, the results of execution are guaranteed to be in the order of the instructions of the program. However, in order to guarantee the order of data accesses among multiple bus masters outside a mutual exclusion control section, synchronization processing should be performed before the subsequent data access. For details of synchronization processing, refer to **Section 3.8.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation** and **Section 3.8.2, Synchronization of Load Instruction Completion and Subsequent Instruction Generation**.

This CPU provides four instructions dedicated for synchronization processing.

In addition, since a wait for the result of preceding load processing depends on general-purpose registers and handled automatically by hardware, no synchronization processing is required by software.

(1) SYNCP

The SYNCP instruction is a special instruction for pipeline synchronization processing, which makes pipelines wait for the result of executing a preceding instruction is reflected before starting to execute a subsequent instruction. Though the SYNCP instruction causes a wait for the results of executing all preceding load processing (up to storing of the loaded values in general-purpose registers), it does not cause waiting for the results of executing store processing (updating of the destination memory locations for storage and of memory-mapped control registers). For how to be sure that the results of executing store processing are reflected for subsequent instructions, see **Section 3.2.7.2, Guaranteeing the Completion of Store Instruction**.

(2) SYNCM

The SYNCM instruction is a special instruction for memory-access synchronization processing, and causes a wait for the results of executing all preceding load processing (up to storage of the loaded values in general-purpose registers) and for the results of executing all preceding store processing (updating of the destination memory locations for storage and memory-mapped control registers). However, for those bus systems and peripheral devices for which storage is speculatively completed (resulting in delays in the completion of writing to the memory and control registers), the SYNCM instruction does not guarantee completion of the execution of store instructions. For how to be sure that the results of executing store processing are reflected for subsequent instructions in such cases, see **Section 3.2.7.2, Guaranteeing the Completion of Store Instruction**.

(3) SYNCI

The SYNCI instruction is a special instruction for the synchronization of instruction fetching. It causes subsequent instructions that have already been fetched into the pipeline but not executed to be discarded, and restarts instruction fetching from the earliest of the discarded instructions to have been fetched. This instruction is used when the result of a preceding instruction must be reflected in processing by subsequently fetched instructions. Though the SYNCI instruction causes a wait for the results of executing all preceding load processing (up to storing of the loaded values in

general-purpose registers), it does not cause waiting for the results of executing store processing (updating of the destination memory locations for storage and of memory-mapped control registers). If a cache instruction has been issued, this instruction also causes a wait for the completion of the instruction cache operation.

The SYNCI instruction should be executed after changes to the CPU operating mode or reflection of changes to the memory protection settings if this will be required in subsequent instruction fetching. For details, see **Section 3.2.7.3(1), Updating the Settings Related to Instruction Fetching**.

When reflecting the results of executing store instructions in subsequent instruction fetching is required, due to changes to self-modifying code or switching between areas of the code-flash memory, execute a dummy read before the SYNCI instruction. This causes a wait for the results of executing store instructions, resulting in these being reflected in subsequent instruction fetching. For details, see **Section 3.2.7.2, Guaranteeing the Completion of Store Instruction**.

(4) SYNCE

The SYNCE instruction of this CPU does not perform processing related to synchronization.

The effects of the four synchronization processing instructions are summarized in **Table 3.127**.

Table 3.127 Effects of Synchronization Processing Instructions

Synchronization Processing Instruction	Guaranteed Synchronization Processing				
	Instruction Fetching		Instruction Execution		
	Subsequent Instructions	Cache Operations* ¹	Load Instructions	Store Instructions	Instructions for Other Operations
SYNCP	—	—	Execution completed* ³	—	Execution completed
SYNCM	—	—	Execution completed* ³	Execution completed* ⁴	Execution completed
SYNCI	Re-fetching* ²	Execution completed	Execution completed* ³	—	Execution completed
SYNCE	—	—	—	—	—

Note 1. This indicates the execution of cache instructions. The LDSR instruction to update the function registers for cache operations is included among the instructions for other operations.

Note 2. This CPU speculatively fetches instructions, so even if the next instruction from the SYNCI instruction has already been fetched, it is fetched again.

Note 3. This indicates the processing up to the storing of loaded data in general-purpose registers

Note 4. This indicates the processing up to updating of the destination memory locations and memory-mapped control registers for storage. However, updating may not be guaranteed. This depends on the specification of the destination for writing. For details, see **Section 3.2.7.2, Guaranteeing the Completion of Store Instruction**.

3.2.7.2 Guaranteeing the Completion of Store Instruction

Instructions for storage may appear to be completed faster than those for loading given the same address. However, the data for storage may not actually have been stored when the instruction has been completed. For example, in the synchronization of multiple cores or the manipulation of peripheral devices, processing related to the stored values should only proceed after storage in the target locations is confirmed. This should be checked by software after the execution of instructions for storage.

This CPU provides two approaches to waiting for the progress of storage at different stages.

(1) Using the SYNCM Instruction to Wait for the Completion of Storage

Executing the SYNCM instruction causes operations to wait until the results of preceding store instructions will be correctly read by subsequent reading. This guarantees confirmed execution when an instruction stored in the memory is to be executed or data are to be transferred to the other CPU core or by DMA via the memory.

In the following example, after waiting for the execution of the two store instructions that proceed the SYNCM instruction, the processing for which waiting was required proceeds in “Operations” below. Processing to branch to an address where an instruction code is stored, for notification of the other CPU, and so on, falls into the category of “Operations”.

ST.W	r21, 0[r1]	//	Store data 1
ST.W	r22, 4[r1]	//	Store data 2
SYNCM		//	Wait for completion of the store instructions
Operation		//	Processing after waiting for completion of the store instructions

However, the SYNCM instruction does not guarantee that the results of preceding store instructions will have already been reflected at the destinations. For example, when storing a value in a register of a peripheral device, the register of the peripheral device may not have been updated when waiting due to the SYNCM instruction has finished, and the resulting may be operation that is not as desired. In such cases, use dummy reading to cause the wait. This is described in **Section 3.2.7.2(2), Using Dummy Reading to Wait for the Completion of Storage**.

In addition, since the SYNCM instruction requires hardware operations to wait in response to its execution, it is only effective in access to bus slaves where the CPU can recognize the progress of processing for storage. If a bus slave is capable of arbitrating requests from multiple bus masters, or is connected to the bus via a bus bridge, the CPU is unable to recognize the progress of storage, so using the SYNCM instruction to cause the wait may not be effective.

The bus slaves (address groups) that support using the SYNCM instruction to wait for the completion of storage in this CPU are listed in **Table 3.128**.

Table 3.128 Bus Slaves that Support Using the SYNCM Instruction to Wait for the Completion of Storage in This CPU

Bus Slaves	Description
L1RAM (given CPU)	Waiting for the updating of memory is also possible.*1
L1RAM (other CPU)	A configuration of two CPU cores can be assumed for this device. For configurations with three or more cores, refer to Section 3.8.1.3, Product information of SYNCM .
L2RAM	A configuration of two CPU cores with a single shared L2RAM can be assumed for this device. For configurations with three or more cores or having multiple L2RAMs, refer to Section 3.8.1.3, Product information of SYNCM .
INTC1	This is the dedicated interrupt controller for connection with the CPU.
Others	Depends on the specifications of the connected bus or slave. For details, refer to Section 3.8.1.3, Product information of SYNCM .

Note 1. For details, **Section 3.2.7.2(2), Using Dummy Reading to Wait for the Completion of Storage**.

The instructions for which the SYNCM instruction causes a wait are all preceding instructions for storing and loading. Multiple instructions for storing and loading running at the same time raises the possibility of the wait being longer than is necessary.

One SYNCM instruction per store instruction is not required. Following consecutive storage in a specific bus slave in units of memory type, peripheral device, and so on, a single SYNCM instruction after the last store instruction leads to a wait for the completion of all preceding store instructions. In the above example, one SYNCM instruction is executed after the execution of two store instructions.

(2) Using Dummy Reading to Wait for the Completion of Storage

Executing a load instruction for the same address immediately after a store instruction can also be used to wait for the completion of a store instruction. This approach is unlike using the SYNCM instruction to cause waiting, since it involves actually waiting for the result of the store instruction to be reflected. On the other hand, this approach may require more cycles than using the SYNCM instruction.

Specifying r0 as the destination for storage of the value loaded by the dummy read enables discarding of the loaded values when they are unnecessary, without destroying the other register values. The following is an example of code for this.

ST.W	r21, 0[r1]	//	Store data 1
ST.W	r22, 4[r1]	//	Store data 2
LD.W	4[r1], r0	//	Load data from the target for access (dummy read).
SYNCP		//	Wait for loading to be completed.
Operation		//	Processing after waiting for completion of the store instructions

Unlike using the SYNCM instruction, this approach combines functions of the CPUs to cause the wait. Since it does not require dedicated hardware, it may be used with many bus slaves.

On the other hand, if a cache for temporarily storing data is present in the bus system along the path to the destination, a value to be stored will be read by a subsequent load instruction even if it has not yet reached its destination, so this method cannot guarantee the completion of store instructions in such cases. Also, depending on the specifications of peripheral devices, a predetermined time after a change to a setting by a store instruction may have to elapse before the desired value can be read. In addition to this approach, such cases require processing to wait for the predetermined time.

Though this CPU does not include such cache functions or peripheral functions with limits on the time to elapse before reading, the situation depends on the product. For details, refer to **Section 3.8.1.4, Product information of Waiting for the Completion of Storage**.

As with using the SYNCM instruction, in using dummy reading, following consecutive storage in a particular bus slave, only a single load instruction following the final store instruction is effective in waiting for the execution of all preceding store instructions to be complete.

Table 3.129 lists the bus slaves for which using dummy reading to wait is effective.

Table 3.129 Bus Slaves for which Using Dummy Reading to Wait is Effective

Bus Slaves	Description
L1RAM (other CPU)	
L2RAM	
INTC1	
Others	Depends on the specifications of the connected bus or slave. For details, refer to Section 3.8.1.4, Product information of Waiting for the Completion of Storage .

In the case of access to the L1RAM of a given CPU, for the sake of the efficiency of processing, execution of the load instruction may be completed before the memory is actually updated. On the other hand, it is possible to execute a SYNCM instruction to cause the wait for completion of the updating of memory. Therefore, use the SYNCM instruction to cause the wait for the completion of instructions for storage in the L1RAM of a given CPU.

When reflecting the results of executing store instructions in subsequent instruction fetching is required, due to changes to self-modifying code or switching between areas of the code-flash memory, execute a SYNCI instruction rather than a SYNCP instruction after the dummy read. This causes a wait for the results of executing store instructions, resulting in these being reflected in subsequent instruction fetching. The following shows an example of code for this.

ST.W	r21, 0[r1]	//	A change relating to instruction fetching
LD.W	0[r1], r0	//	Load data from the target for access (dummy read).
SYNCI		//	Synchronization of the wait for the execution of loading and instruction fetching
Operation		//	Processing after the change to instruction fetching operations

3.2.7.3 Hazard Management after System Register Update

If an LDSR instruction is executed to update the setting of a system register before an STSR instruction is executed to read the system register or before a CALLT instruction or the like, which uses the system register, is executed, then the new setting must be reflected in order to perform the desired operation.

This CPU guarantees that if an LDSR instruction is used to update system registers shown in **Table 3.130**, the new register setting will be applied when the subsequent instruction is executed. However, it does not guarantee that the new setting will be applied in instruction fetching. In this case, synchronization process is required. Also, the execution of an EI or DI instruction is treated in the same way as the update of PSW by an LDSR instruction; that is, it is guaranteed that the values updated by an EI or DI instruction is applied to the subsequent instruction.

Table 3.130 System Registers that Guarantee Reflection of their Updates by LDSR Instruction to the Subsequent Instruction

selID*1	System Register*2							
0	EIPC	EIPSW	FEPC	FEPSW	PSW	FPSR	FPEPC	FPST
	FPCC	FPCFG	EIIC	FEIC	CTPC	CTPSW	CTBP	SNZCFG
	EIWR	FEWR						
1	SPID	SPIDLIST	RBASE	EBASE	INTBP*3	MCTL	PID	SVLOCK
	SCCFG	SCBP						
2	PEID	BMID	MEA	MEI	ISPR*3	ICSR	INTCFG*3	PLMR*3
	RBCR0*4	RBCR1*4	RBNR*4	RBIP*4				
5	MPM	MPCFG	MCA	MCS	MCC	MCR	MCI	MPIDX*5
10	FXSR	FXST	FXINFO	FXCFG	FXXC	FXXP		
13	RDBCR							

Note 1. In the representation of LDSR and STSR instructions, selection ID are represented as selID.

Note 2. This table includes a register whose value cannot be updated and a register that has a bit whose value cannot be updated.

Note 3. If these registers, which control the acceptance of interrupts, are updated, interrupts will be accepted with the new register settings if interrupt requests are present at the time of executing the subsequent instruction.

Note 4. If these registers, which control the register bank function, are updated, interrupts will be accepted with the new register settings if interrupt requests are present at the time of executing the subsequent instruction. However, when the RESBANK instruction is executed after updating the RBCR0.MD, a synchronization process is required.

Note 5. If an LDSR instruction is executed to update the MPLA, MPUA, or MPAT register immediately after another LDSR instruction is executed to update the MPIDX register, it is guaranteed that the new setting of MPIDX is reflected in the subsequent instructions. In order to read the MPLA, MPUA, or MPAT register by using a STSR instruction after the update of the MPIDX register, a synchronization process is required.

If the settings related to instruction fetch or a system register that is not shown in **Table 3.130** is updated, the new register setting can be reflected by performing any of the following synchronization processes immediately after the LDSR instruction. The appropriate synchronization process should be decided according to details of the new register setting and the subsequent operation.

For SYNCI or SYNCP instruction, that are used for synchronization process, see **Section 3.2.7.1, Synchronization Processing**.

(1) Updating the Settings Related to Instruction Fetching

To effectively supply instruction codes read from the memory to the instruction execution unit, operations of instruction fetching are independent from those of the instruction execution unit and are speculative. This means that the progress of reading the subsequent instruction code by instruction fetching is not known on completion of the system register update. In other words, the result from which instruction fetching is applied to the new setting of the system register is not known. Therefore, an explicit synchronization process is required for updates of the system registers which are related to instruction fetching. This process is also necessary for updates of the system registers listed in **Table 3.130** in order to ensure that the updates are reflected in instruction fetching.

The system registers related to instruction fetching are listed below:

CPU operating mode:	PSW.UM, PSW.CU1, PSW.CU0
Access protection:	SPID
Memory protection:	MPU function registers
Instruction cache:	Cache operation function registers
Instruction fetch:	Instruction fetch control registers

If the settings of the system registers listed above are updated, by executing the SYNCI, EIRET, or FERET instruction, the new settings are surely reflected when the subsequent instruction is fetched. Here, these instructions are called “Instruction fetch synchronization instructions”. For usage of the EIRET and FERET instructions, see **Section 3.2.7.3(7), Use of EIRET and FERET Instructions in Synchronization Process**.

For fetching the instruction fetch synchronization instructions themselves, whether the updates of the system registers are reflected or not cannot be decided. In these cases, the following cares must be taken.

(a) Updating PSW.UM

When an LDSR instruction is used to update PSW, especially by setting PSW.UM (to 1), which in response enables memory protection function, be sure that fetching of the SYNCI instruction, which is used for synchronizing the instruction fetch, does not cause a memory protection violation. Also note that setting PSW.UM disables execution of some instructions and updating of some system registers.

On the other hand, when an EIRET or FERET instruction is executed, the value in EIPSW or FEPSW is transferred to PSW as the operation of the instruction. This means that fetching of the instruction takes place before updating of PSW.UM by the instruction itself. For these reasons, updating of PSW.UM by an EIRET or FERET instruction is recommended.

(b) Updating PSW.CU1 and PSW.CU0

After PSW.CU1 or PSW.CU0 is updated, if a FXU or FPU instruction is executed without synchronization process, a coprocessor unusable exception (UCPOP) may not be properly detected. To avoid this, execute a SYNCI instruction for synchronization after updating these bits. Unlike updating of PSW.UM, updating of PSW.CU1 or PSW.CU2 and detection of a coprocessor unusable exception can be surely synchronized by a SYNCI instruction.

For updating PSW.CU1 or PSW.CU2 and PSW.UM at the same time, use of an EIRET or FERET instruction described in **Section 3.2.7.3(1)(a), Updating PSW.UM** is recommended.

(c) Updating SPID

SPID is referenced by memory protection function and the bus system outside the CPU. If an ID without access permission to the memory resources or peripheral devices is set to SPID, instruction fetching does not take place correctly. Be sure to set an appropriate ID to SPID. To ensure that the bus system references any updates on SPID, the instruction cache must be disabled once.

(d) Updating MPU Function Registers

Updating the settings of memory protection by manipulating the MPU function registers is detailed in **Section 3.2.7.3(2), Updating the Memory Protection Settings of MPU**.

For reflecting the updated settings in fetching of the subsequent instruction, execute either of the SYNCI, EIRET, and FERET instructions. Note, however, that whether the instruction fetch synchronization instruction itself is fetched with the new protection settings or not is not known. In this case, the following operations are required.

- i) Enable memory protection in user mode only and update the settings in supervisor mode.
- ii) Update the settings of memory protection in the area where execution of instructions are allowed.
- iii) Surely set execution of instructions for the memory area including those for instruction fetch synchronization instruction to be enabled

For updating PSW.UM with operation i) above, use of an EIRET or FERET instruction described in **Section 3.2.7.3(1)(a), Updating PSW.UM** is recommended.

(e) Updating Cache Operation Function Registers

Updating these registers by an instruction does not require certain synchronization process. However, if you want to make sure that the disabled state of the instruction cache is reflected in fetching of the subsequent instruction, execute an instruction fetch synchronization instruction after the update of the cache operation function registers.

(f) Updating Instruction Fetch Control Registers

Updating these registers by an instruction does not require certain synchronization process.

(2) Updating the Memory Protection Settings of MPU

The system registers of MPU that are shown in **Table 3.130** guarantee that their latest settings are reflected in the subsequent instruction, but this does not apply to other system registers (MPIDX, MPLA, MPUA, MPAT, and MPIDn (n = 0 to 7) that configure the settings of protection areas.

In order to read settings updated by an LDSR instruction by using the subsequent STSR instruction, the synchronization process described in **Section 3.2.7.3(4), Updating Register Bank Function-Related System Registers** is required.

For reflecting the updated memory protection settings in the instruction fetch, see **Section 3.2.7.3(1), Updating the Settings Related to Instruction Fetching**.

For reflecting the updated memory protection setting in the instruction that makes operand accesses, execute a SYNCI or SYNCP instruction after the update. This ensures that the new protection setting is reflected in the operand accesses by the subsequent instructions.

The EIRET and FERET instructions can be used instead of the SYNCI and SYNCP instructions. Especially, for the case of updating memory protection settings related to instruction fetch as well as making operand accesses, use of an EIRET or FERET instruction is recommended.

(3) Updating Interrupt-Related System Registers

As shown in **Table 3.130**, when the system registers related to interrupts (ISPR, INTCFG, PLMR) are updated, it is guaranteed that the updates are reflected when the subsequent instruction is executed.

However, when updating the setting of INTCFG to stop automatic update of ISPR, for example, if an interrupt is generated after INTCFG is updated and before an LDSR instruction that updates ISPR is executed, only the update in INTCFG is reflected in the interrupt handling. As the update of ISPR is not reflected, the desired interrupt handling cannot be performed.

To avoid this, when updating the system registers related to interrupts, it is recommended to disable interrupts by setting PSW.ID to 1.

We also recommend that updating of PSW.EBV and EBASE, which specify vector address of an interrupt handler, are executed while interrupts are disabled. For details, see **Section 3.2.7.3(8), Updating PSW.EBV and EBASE**.

(4) Updating Register Bank Function-Related System Registers

As shown in **Table 3.130**, when the system registers related to the register bank function (RBCR0, RBCR1, RBNR, RBIP) are updated, it is guaranteed that the updates are reflected when the subsequent instruction is executed. For example, if an interrupt is generated after an LDSR instruction that updates RBIP is executed, the automatic context saving to the register bank is executed using the updated value of RBIP.

However, when updating the settings of RBCR0 and RBCR1, for example, if an interrupt is generated after RBCR0 is updated and before an LDSR instruction that updates RBCR1 is executed, only the update in RBCR0 is reflected in the automatic context saving to the register bank. As the update of RBCR1 is not reflected, the desired processing cannot be performed.

To avoid this, when updating the system registers related to the register bank function, it is recommended to disable interrupts by setting PSW.ID to 1.

For RBCR0 only, synchronization process by the SYNCI instruction is necessary to surely reflect the new setting of the register to the subsequent RESBANK instruction. The following shows an example of code for this.

DI		// Disable interrupt
LDSR	r20, 15, 2	// Update RBCR0
SYNCI		// Synchronization of instruction fetching
RESBANK		// Execute RESBANK using updated RBCR0

(5) Reading a System Register by Using an STSR Instruction

After executing an LDSR instruction to update a system register, execute a SYNCP instruction before executing an STSR instruction for read operation. Since the SYNCP instruction waits until the execution of the preceding LDSR instruction is completed, it is ensured that the new register setting is read.

For the system registers shown in **Table 3.130**, the new register settings are read without synchronization process by SYNCP instruction.

(6) Referencing a System Register by the Subsequent Instruction

After executing an LDSR instruction to update a system register, execute a SYNCP instruction before executing an instruction to refer to the system register. Since the SYNCP instruction waits until the execution of the preceding LDSR instruction is completed, it is ensured that the new register setting is reflected.

(7) Use of EIRET and FERET Instructions in Synchronization Process

The EIRET and FERET instructions are defined as return instructions from the exception handler of their corresponding exception levels. These instructions can set any PC and PSW at the same time and therefore usable as instructions for synchronization process in this CPU.

To use an EIRET instruction for synchronization process, set appropriate values to EIPC and EIPSW and execute the instruction. To use an FERET instruction for synchronization process, set appropriate values to FEPC and FEPSW and execute the instruction.

Set values to EIPC and FEPC that is the PCs after these instructions are executed. Set values to EIPSW and FEPSW that is the PSW after these instructions are executed.

Note that these instructions are supervisor privileged instructions. That means, the state of PSW.UM can be changed only from 0 (clear) to 1 (set), supervisor mode to user mode.

For details on the EIRET and FERET instructions, see *the RH850G4MH User's Manual: Software*.

(8) Updating PSW.EBV and EBASE

EBASE is a register that indicates the vector address of the exception handler. This is enabled when PSW.EBV is set (to 1). The recommended updating procedure is as follows.

- 1) Set PSW.ID to 1
- 2) Clear PSW.EBV to 0 (updating at the same time with PSW.ID is possible)
- 3) Update EBASE
- 4) Set PSW.EBV to 1
- 5) Clear PSW.ID to 0 (updating at the same time with PSW.EBV is possible)

For updating PSW.UM in the procedure above as well, use of an EIRET or FERET instruction described in **Section 3.2.7.3(1)(a), Updating PSW.UM** is recommended.

3.2.8 Reset

3.2.8.1 Status of Registers After Reset

If a reset signal is input by a method defined by the hardware specifications, the program registers and system registers are placed in the status shown by the value after reset of each register in **Section 3.2.3, Register Set**, and program execution is started. Set the contents of each register to an appropriate value in the program.

The CPU executes a reset to start execution of a program from the reset address specified by **Section 3.2.4.4, Exception Handler Address**.

Note that because the PSW.ID bit is set to 1 immediately after a reset, conditional EI level exceptions will not be acknowledged. To acknowledge conditional EI level exceptions, clear the PSW.ID bit to 0.

3.2.9 Ensuring Coherency after Code Flash Programming

The CPU has an efficient instruction cache and data buffer for the code flash area.

Therefore, after using self-programming to program the code flash memory, clear the instruction cache and data buffer to ensure coherency. The instruction cache and data buffer can be cleared by using the ICCTRL register and the RDBCR register, respectively. The entire CPU's instruction cache and data buffer can also be cleared at the same time by using the TM_CC register. For details on the ICCTRL and RDBCR register, see **Section 3.2.3.7, Cache Operation Function Registers** and **Section 3.2.3.9, Hardware Function Registers**. For details on the TM_CC register, see **Section 3.2.9.1, Registers**.

3.2.9.1 Registers

(1) Register Base Address

The GCFU base address is listed in the following table. The GCFU register addresses are given as offsets from the base addresses.

Table 3.131 Register Base Address

Base Address Name	Base Address	Bus Group
<GCFU_base>	FFFB 1400 _H	Peripheral Group 0

(2) Clock Supply

Clock supply by and to GCFU is listed in the following table.

Table 3.132 Clock Supply

Unit Name	Unit Clock Name	Clock Supply Name
GCFU	PCLK	CLK_HBUS

(3) List of Registers

Table 3.133 List of Registers

Module Name	Register Name	Symbol	Address	Access
GCFU	Cache Clear Operation Register	TM_CC	<GCFU_base> + 08 _H	32

(4) TM_CC — Cache Clear Operation Register

The TM_CC register is used to issue requests for clearing of the unit (instruction cache or data buffer) that holds data from the code flash.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICCLR	DCCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 3.134 TM_CC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 2	—	Reserved These bits are always read as 0. The write value should always be 0.
1	ICCLR	<p>Instruction Cache Clear Request</p> <p>Writing 1 to this bit issues an instruction cache clear request. Writing 0 is ignored. The read value is 1 when this bit is read before completion of the cache clear request, and 0 when it is read after completion of the cache clear request.</p> <p>Note: If 1 has been written to this bit and then 1 is written again before completion of the cache clear request, the second write is ignored. (The cache clear request is not executed by the second write.)</p> <p>If 1 has been written to this bit and then 0 is written before completion of the cache clear request, the cache clear request is not canceled.</p> <p>To execute cache clear requests continuously, be sure to check that execution of the preceding cache clear request has been completed (by reading this bit and checking that the readout value is 0) before executing the next cache clear request.</p> <p>0: Indicates that an instruction cache clear request is not currently being processed, or processing of a cache clear request has been completed (when read).</p> <p>1: Issues an instruction cache clear request (when written). Indicates that an instruction cache clear request is currently being processed (when read).</p>

Table 3.134 TM_CC Register Contents (2/2)

Bit Position	Bit Name	Function
0	DCCLR	<p>Data Buffer Clear Request</p> <p>Writing 1 to this bit issues a data buffer clear request. Writing 0 is ignored.</p> <p>The read value is 1 when this bit is read before completion of the data buffer clear request, and 0 when it is read after completion of the data buffer clear request.</p> <p>Note: If 1 has been written to this bit and then 1 is written again before completion of the data buffer clear request, the second write is ignored. (The data buffer clear request is not executed by the second write.)</p> <p>If 1 has been written to this bit and then 0 is written before completion of the data buffer clear request, the data buffer clear request is not canceled.</p> <p>To execute data buffer clear requests continuously, be sure to check that execution of the preceding data buffer clear request has been completed (by reading this bit and checking that the readout value is 0) before executing the next data buffer clear request.</p> <p>0: Indicates that a data buffer clear request is not currently being processed, or processing of a data buffer clear request has been completed (when read).</p> <p>1: Issues a data buffer clear request (when written). Indicates that a data buffer clear request is currently being processed (when read).</p>

3.3 Inter-CPU Overview

This subsection contains a Barrier-Synchronization (BARR), Inter-Processor Interrupts (IPIR), Mutual Exclusion (MEV) and Time Protection Timer (TPTM). The first part of this subsection describes this product's specific properties, such as the number of units, register base addresses, etc. The remainder of the subsection shows the functions and registers.

3.3.1 Features

3.3.1.1 Number of Units

Table 3.135 Number of Units

Product Name		RH850/E2x-FCC1	
		373 pins	292 pins
BARR	Number of units	1	1
	Name of unit	BARR	
IPIR	Number of units	1	1
	Number of channels for each CPU	4	4
	Name of unit	IPIR	
MEV	Number of units	1	1
	Number of registers in a unit	32	32
	Name of unit	MEV	
TPTM	Number of units	1	1
	Number of channels for each CPU	4 interval timers 1 free-run timer	4 interval timers 1 free-run timer
	Name of unit	TPTM	

Product Name		RH850/E2M	
		373 pins	292 pins
BARR	Number of units	1	1
	Name of unit	BARR	
IPIR	Number of units	1	1
	Number of channels for each CPU	4	4
	Name of unit	IPIR	
MEV	Number of units	1	1
	Number of register in a unit	32	32
	Name of unit	MEV	
TPTM	Number of units	1	1
	Number of channels for each CPU	4 interval timers 1 free-run timer	4 interval timers 1 free-run timer
	Name of unit	TPTM	

3.3.1.2 Register Base Addresses

Table 3.136 Register Base Addresses

Base Address Name	Base Address	Bus Group
<BARR_base>	FFFB 8000 _H	I-Bus Group2
<IPIR_base>	FFFB 9000 _H	I-Bus Group1
<MEV_base>	FFFB A000 _H	I-Bus Group0
<TPTM_base>	FFFB B000 _H	I-Bus Group3

3.3.1.3 Clock Supplies

The clock supplies are listed in the following table.

Table 3.137 Clock Supplies

Unit Name	Unit Clock Name	Supply Clock Name
BARR	cpu_clk	CLK_CPU
IPIR	cpu_clk	CLK_CPU
MEV	cpu_clk	CLK_CPU
TPTM	cpu_clk	CLK_CPU

3.3.1.4 Interrupt Requests

The interrupt requests are listed in the following table.

Table 3.138 Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Description	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
INTIPIR0	ipir_int_ch0	IPIR CH0 interrupt	EIINT0	—	—
INTIPIR1	ipir_int_ch1	IPIR CH1 interrupt	EIINT1	—	—
INTIPIR2	ipir_int_ch2	IPIR CH2 interrupt	EIINT2	—	—
INTIPIR3	ipir_int_ch3	IPIR CH3 interrupt	EIINT3	—	—
FEINT1 or INTTPTM*1	TPTM_IRQ	TPTM interval interrupt	FEINT1 or EIINT31*1	—	—

Note 1. This interrupt can be selected for use as FEINT1 or EIINT31. See **Section 6.3.13, TPTMSEL — TPTM Interrupt FE EI Select Register**.

3.3.1.5 Reset Sources

The reset sources are shown below. These modules are initialized by the following reset sources.

Table 3.139 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
BARR	All registers	✓	✓	✓	✓	—	—
IPIR	All registers	✓	✓	✓	✓	—	—
MEV	All registers	✓	✓	✓	✓	—	—
TPTM	All registers	✓	✓	✓	✓	—	—

3.3.1.6 External Input/Output Signals

This module has no external input/output signals.

3.3.2 Processor Element Identifier

The PEID, the ID number of a processor element, can be read from the PEID register. Which CPU core performs a specific program can be known by referring to the PEID. The following shows the PEIDs of this product.

For the PEID register, see **Section 3.2.3.2, Basic System Registers**.

CPU Core	PEID
CPU0 (PE0)	000 _B
CPU1 (PE1)	001 _B

3.4 Inter-Processor Interrupts

3.4.1 Inter-Processor Interrupts Overview

Inter Processor Interrupt Register (IPIR) is a function that controls fast interrupt requests between PEs. Use of IPIR achieves faster processing of inter-PE interrupts than setting a request flag for the INTC2 interrupt channel by using software.

IPIR has the following features.

- Support of inter-PE interrupt function of 4 channels
- Level detection of interrupts. Edge detection is not supported.
- Accessible from all clusters and all PEs
- Identification of interrupt request source PE is possible.
- Unintended inter-PE interrupts can be prevented by masking interrupt requests.
- SET1, CLR1, and NOT1 can be executed as atomic operation instructions to IPIR.
 - Read-modify-write operations using the LD instruction and the ST instruction are also possible, but not as atomic operations.

3.4.2 Inter-Processor Interrupts Registers

3.4.2.1 List of Registers

Table 3.140 List of Registers

Module Name	Register Name	Register Symbol	Address	Access	Access Protection
IPIR	IPIRn*2 Inter-PE Interrupt Enable Self-Register*1	IPIInENS	<IPIR_base> + 000 _H + 020 _H × n	8	—
	IPIRn*2 Inter-PE Interrupt Flag Self-Register*1	IPIInFLGS	<IPIR_base> + 004 _H + 020 _H × n	8	—
	IPIRn*2 Inter-PE Interrupt Clear Self-Register*1	IPIInFCLRS	<IPIR_base> + 008 _H + 020 _H × n	8	—
	IPIRn*2 Inter-PE Interrupt Request Self-Register*1	IPIInREQS	<IPIR_base> + 010 _H + 020 _H × n	8	—
	IPIRn*2 Inter-PE Interrupt Request Clear Self-Register*1	IPIInRCLRS	<IPIR_base> + 014 _H + 020 _H × n	8	—
	IPIRn*2 Inter-PE Interrupt Enable Register m*3	IPIInENm	<IPIR_base> + 800 _H + 020 _H × n + 100 _H × m	8	—
	IPIRn*2 Inter-PE Interrupt Flag Register m*3	IPIInFLGm	<IPIR_base> + 804 _H + 020 _H × n + 100 _H × m	8	—
	IPIRn*2 Inter-PE Interrupt Clear Register m*3	IPIInFCLRm	<IPIR_base> + 808 _H + 020 _H × n + 100 _H × m	8	—
	IPIRn*2 Inter-PE Interrupt Request Register m*3	IPIInREQm	<IPIR_base> + 810 _H + 020 _H × n + 100 _H × m	8	—
	IPIRn*2 Inter-PE Interrupt Request Clear Register m*3	IPIInRCLRm	<IPIR_base> + 814 _H + 020 _H × n + 100 _H × m	8	—

Note 1. When PE accesses the self register, PE can access the corresponding register for each PE. For example, when PE1 accesses the IPIOREQS register, PE1 can also access the IPIOREQ1 register.

Note 2. n = 0 to 3. n is the channel number of IPIR.

Note 3. m = 0 to 1. m is the PEID.

3.4.2.2 Self Region

The self region contains the following five types of registers.

- IPInENS — IPIRn Inter-PE Interrupt Enable Self-Register
- IPInFLGS — IPIRn Inter-PE Interrupt Flag Self-Register
- IPInFCLRS — IPIRn Inter-PE Interrupt Clear Self-Register
- IPInREQS — IPIRn Inter-PE Interrupt Request Self-Register
- IPInRCLRS — IPIRn Inter-PE Interrupt Request Clear Self-Register

The self-registers are virtual registers that do not physically exist. Access from each PE to self-registers is routed to the actual register corresponding to the access source PE. **Table 3.141** lists the access source PEs and routing destination registers. For the functions of each register bit, refer to the specifications of the routing destination register. When the masters except PEx access the self-registers, the registers return 0_B, and write access is ignored.

It is basically assumed that the IPIR registers are accessed from PEs via self region when PEs use the IPIR function. This allows PEs to use the same code because it is not necessary to specify different register addresses for each PE.

It is also possible to directly access registers by specifying the address of the actual register without using the self-registers. In this case, the purpose is assumed to be to check the status of registers for another PE, or to reference the individual registers by a debugging tool.

Table 3.141 Self Region Register Routing List

Self Register	Access Source PE	
	PE0	PE1
IPInENS	IPInEN0	IPInEN1
IPInFLGS	IPInFLG0	IPInFLG1
IPInFCLRS	IPInFCLR0	IPInFCLR1
IPInREQS	IPInREQ0	IPInREQ1
IPInRCLRS	IPInRCLR0	IPInRCLR1

3.4.2.3 IPIInENm — IPIRn Inter-PE Interrupt Enable Register m (n = 0 to 3)

The IPIRn Inter-PE Interrupt Enable Register m (IPIInENm) is an 8-bit read/write register that sets the transmitting PE allowed to issue inter-PE interrupt requests to PEm.

This register is used to enable inter-PE interrupt requests of PE (PE_x) by the receiving PE (PE_m) itself.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	EN	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 3.142 IPIInENm Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	EN	Inter-PE Interrupt Enable. Write 1 to the x-th bit to enable the issuance of inter-PE interrupt requests from PE _x to PE _m . Write 0 to the x-th bit to disable the issuance of inter-PE interrupt requests from PE _x to PE _m . Bit 0: PE0 Bit 1: PE1

3.4.2.4 IPIInFLGm — IPIRn Inter-PE Interrupt Flag Register m (n = 0 to 3)

The IPIRn Inter-PE Interrupt Flag Register m (IPIInFLGm) is an 8-bit read only register that indicates the transmitting PE that issued an inter-PE interrupt request to PEm.

This register is used to distinguish the requesting PE by PEm when PEm has received an inter-PE interrupt request.

Value after reset: 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	FLG	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 3.143 IPIInFLGm Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	FLG	Inter-PE Interrupt Request Flag. This register indicates the status of Inter-PE Interrupt request from other PEs. The x-th bit is automatically updated according to the IPIInREQx[m] change while the value of the IPIInENm[x] bit is 1. The x-th bit is not set if IPIInENm[x] is 0, even if IPIInREQx[m] is set. The x-th bit is also cleared when the IPIInFCLRm[x] bit is set. Bit 0: PE0 Bit 1: PE1

3.4.2.5 IPInFCLRm — IPIRn Inter-PE Interrupt Clear Register m (n = 0 to 3)

The Inter-PE Interrupt Clear Register m (IPInFCLRn) is an 8-bit write only register that clears inter-PE interrupt requests to PEm.

This register is used to clear the request flag by PEm after receiving the Inter-PE interrupt request from another PE.

Value after reset: 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	FCLR	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W

Table 3.144 IPInFCLRm Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	FCLR	Inter-PE Interrupt Request Flag Clear IPInFLGm[x] can be cleared by writing 1 to this bit. Writing 0 is ignored. Bit 0: PE0 Bit 1: PE1

3.4.2.6 IPIInREQm — IPIRn Inter-PE Interrupt Request Register m (n = 0 to 3)

The Inter-PE Interrupt Request Register m is an 8-bit read/write register that controls Inter-PE interrupt request from PEm to other PEs.

Value after reset: 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	REQ	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 3.145 IPIInREQm Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	REQ	When 1 is written to the x-th bit while the value of the IPIInENx[m] bit is 1: The value of the x-th bit becomes 1. The high level is output to the inter-PE interrupt request for PEx, and IPIInFLGx[m] is automatically set to 1. When 1 is written to the x-th bit while the value of the IPIInENx[m] bit is 0: The value of the x-th bit becomes 1. There are no other operations. When 0 is written to the x-th bit: Writing 0 is ignored. When read: The register value is read out. Bit 0: PE0 Bit 1: PE1

3.4.2.7 IPIInRCLRm — IPIInR Inter-PE Interrupt Request Clear Register m (n = 0 to 3)

The Inter-PE Interrupt Request Clear Register m (IPIInRCLRm) is an 8-bit write only register that clears inter-PE interrupt requests for other PEs by PEm.

This register is assumed to be used to clear inter-PE interrupt requests that are set unintentionally while IPIInENm is 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RCLR	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W

Table 3.146 IPIInRCLRm Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	RCLR	Inter-PE Interrupt Request Clear Writing 1 to the x-th bit clears the IPIInREQm[x] bit. If the value of the IPIInENx[m] bit is 1, the IPIInFLGx[m] bit is also cleared at the same time. The transmitting PE allowed to issue inter-PE interrupt requests by the IPIInENm register can also cancel an inter-PE interrupt request by using this bit, but the receiving PE may in some cases accept the inter-PE interrupt request while the cancellation processing takes place. Writing 0 is ignored. This bit always returns 0 when read. Bit 0: PE0 Bit 1: PE1

3.4.3 Inter-Processor Interrupts Function

This product has only PE0 and PE1, but this subsection includes examples of PE0 to PE7.

3.4.3.1 Initial Setting

Initial setting of authorized PEs by the IPIInENm register must be made before IPIR is used. An example of initial setting is given below.

(1) Initial Setting by Receiving PE

Figure 3.40 shows an example of the initial setting of IPIR channel 0 by PE1, which is a receiving PE. In this figure, bits PE4 to 7 of each register are omitted.

First, if IPIR has already been used and the values of bits IPI0REQ0 to 7[1] and the value of the IPI0FLG1 register have changed from the initial values, PE1 writes FF_H*1 to the IPI0FCLRS (= IPI0FCLR1) register to clear bits IPI0REQ0 to 7[1] and the IPI0FLG1 register. If the values of bits IPI0REQ0 to 7[1] and the IPI0FLG1 register are the initial values, for example after hardware reset, this clearing operation is not required.

Next, PE1 writes 01_H to the IPI0ENS (= IPI0EN1) register to accept interrupt requests from PE0 to PE1. Like for the clear register, upon completion of setting of the enable register, PE1 can accept interrupt requests from PE0.

Note 1. As this product has only PE0 and PE1, PE1 writes 03_H to IPI0FCLRS register to clear.

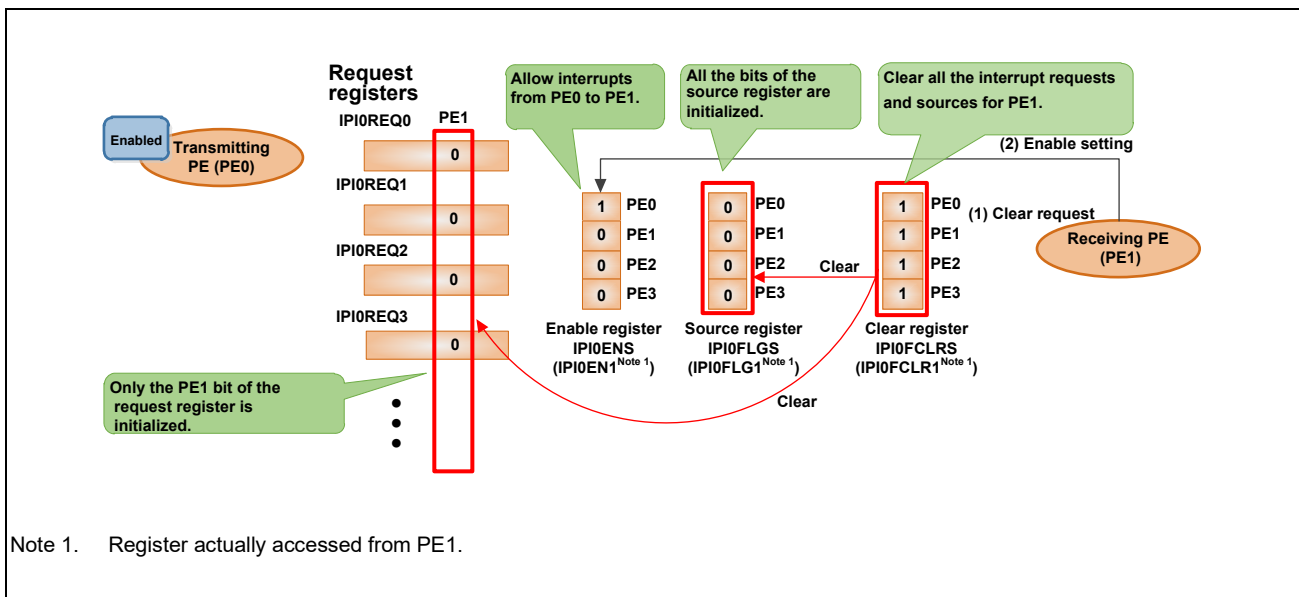


Figure 3.40 Example of Initial Setting of IPIR Channel 0 by Receiving PE (PE1)

(2) Initial Setting by Control PE

An example of initial setting from a PE other than a receiving PE is shown below. In this section, the PE that performs the initial setting is called the control PE for the sake of convenience. **Figure 3.41** shows an example of the initial setting of IPIR channel 0 by PE3, which is the control PE. In this figure, bits PE4 to 7 of each register are omitted.

First, if IPIR has already been used and the values of bits IPIOREQ0 to 7[1] and the value of the IPIOFLG1 register have changed from the initial values, PE3 writes FF_H to the IPIOFCLR1 register to clear bits IPIOREQ0 to 7[1] and the IPIOFLG1 register. If the values of bits IPIOREQ0 to 7[1] and the IPIOFLG1 register are the initial values, for example after hardware reset, this clearing operation is not required.

Next, PE3 writes 01_H to the IPIOEN1 register to accept interrupt requests from PE0 to PE1. Like for the clear register, upon completion of setting of the enable register, PE1 can accept interrupt requests from PE0.

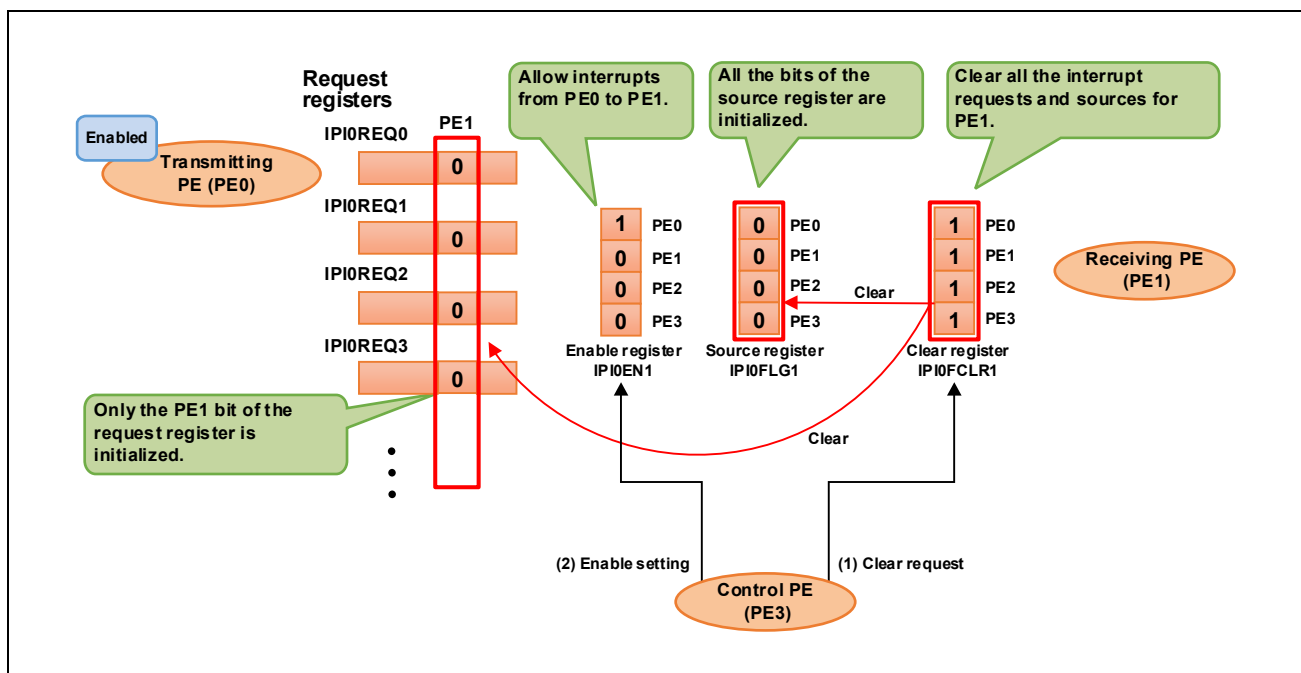


Figure 3.41 Example of Initial Configuration of IPIR Channel 0 by Control PE (PE3)

3.4.3.2 Inter-PE Interrupt Request

Figure 3.42 shows an example of the operation when PE0 sends an inter-PE interrupt request to PE1 using IPIR channel 0. In this figure, bits PE4 to 7 of each register are omitted. In this example, the initial setting is set to IPIOEN1 in advance to enable inter-PE interrupt requests from PE0. For an operation example of the initial setting, see **Section 3.4.3.1, Initial Setting**. The following describes an example of accessing the registers through the self-register.

First, PE0, which is the transmitting PE, reads the IPIOREQS (= IPIOREQ0) register and checks that the value of the IPIOREQS[1] (= IPIOREQ0[1]) bit is 0_B. If the value of the IPIOREQS[1] (= IPIOREQ0[1]) bit is 1_B, this means that the previous inter-PE interrupt request from PE0 to PE1 has not been accepted by PE1, and thus a new inter-PE interrupt request cannot be issued. If it is confirmed that the value of the IPIOREQS[1] (= IPIOREQ0[1]) bit is 0_B and thus an inter-PE interrupt request from PE0 to PE1 is enabled, PE0 sets the IPIOREQS[1] (= IPIOREQ0[1]) bit to 1_B to issue an inter-PE interrupt request from PE1. Because inter-PE interrupts from PE0 to PE1 are enabled by the IPIOEN1 register, when PE0 sets the IPIOREQS[1] (= IPIOREQ0[1]) bit to 1_B, the IPIOFLG1[1] bit is automatically set to 1_B, and an inter-PE interrupt request signal is output to PE1 from IPIR.

Figure 3.43 shows an example of the operation when PE1 receives an inter-PE interrupt request from PE0 using IPIR channel 0. When PE1, which is the receiving PE, receives an inter-PE interrupt signal, it reads the IPI0FLGS (= IPI0FLG1) register, and because the value of the IPI0FLGS[0] (= IPI0FLG1[0]) bit is 1_B, it recognizes that the inter-PE interrupt request has been sent from PE0. But if the request side clears the request, the value of it is 0_B. After verifying the source of the inter-PE interrupt, PE1 sets the IPI0FCLRS[0] (= IPI0FCLR1[0]) bit to 1_B, and the processing transitions to the inter-PE interrupt processing. The IPI0REQS[1] (= IPI0REQ0[1]) bit and the IPI0FLGS (= IPI0FLG1) bit are automatically cleared when the IPI0FCLRS[0] (= IPI0FCLR1[0]) bit is set to 1_B by PE1.

Figure 3.44 shows the operation flow from initial setting of IPIR by the receiving PE (PE1), to sending of an inter-PE interrupt request by the transmitting PE (PE1), and completion of reception of that inter-PE interrupt request by the receiving PE (PE1).

Note that if a new inter-PE interrupt request is generated to the same receiving PE before the source register is cleared by the receiving PE with the clear register, the bit corresponding to the new transmitting PE in the source register will be set to 1_B, but the inter-PE interrupt request signal being retained high, the second and subsequent inter-PE interrupt requests will not be output. Therefore, the processing when multiple sources occur when the receiving PE has received an inter-PE interrupt request must be controlled by software.

The transmitting PE can detect whether the preceding inter-PE interrupt request was successfully received by the receiving PE, by checking the value of the request register. To monitor the request register with a polling loop, it is recommended to curb the bus load by executing the snooze instruction within the polling loop in order to avoid the bus system occupation for extended periods of time.

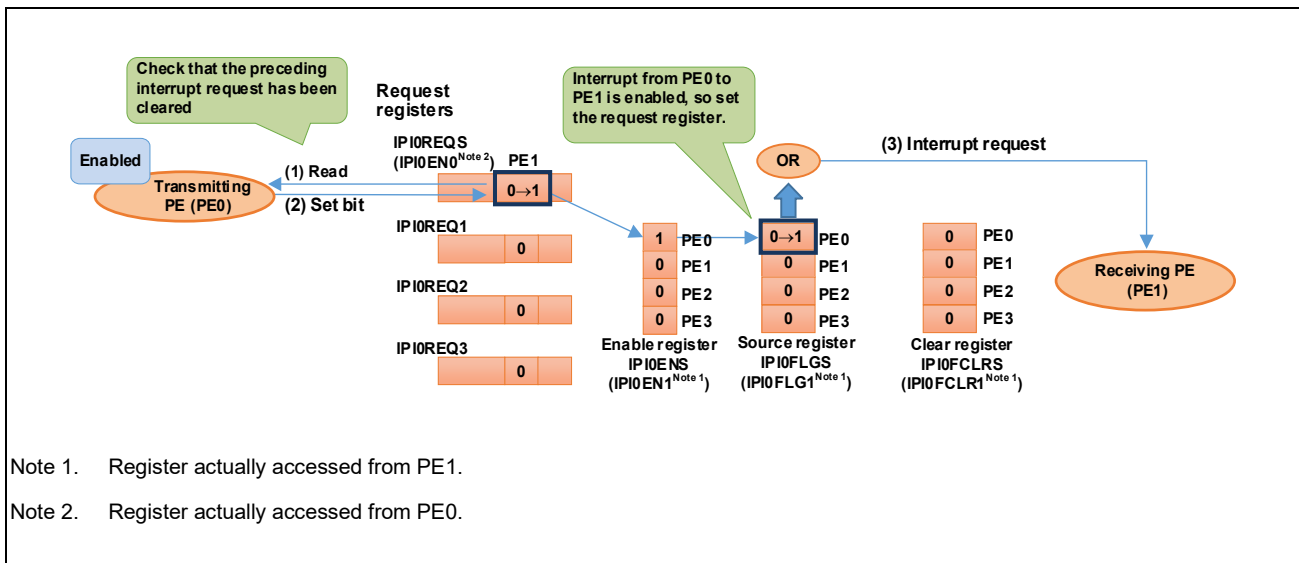
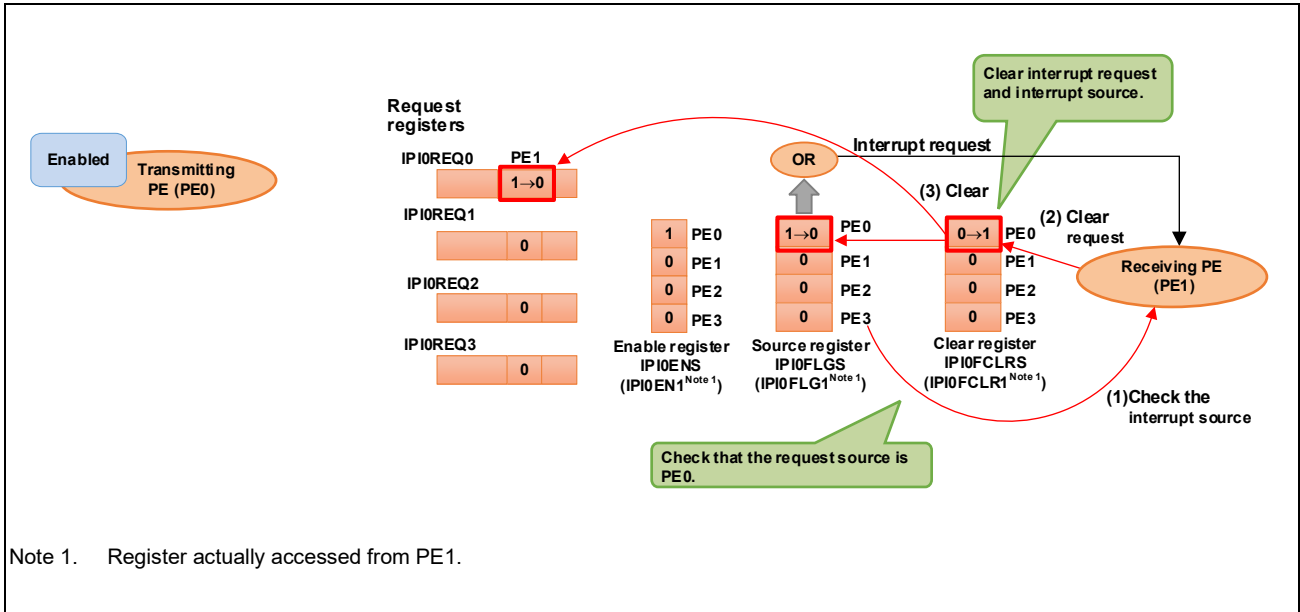


Figure 3.42 Example of Inter-PE Interrupt Transmission Operation



Note 1. Register actually accessed from PE1.

Figure 3.43 Example of Inter-PE Interrupt Reception Operation

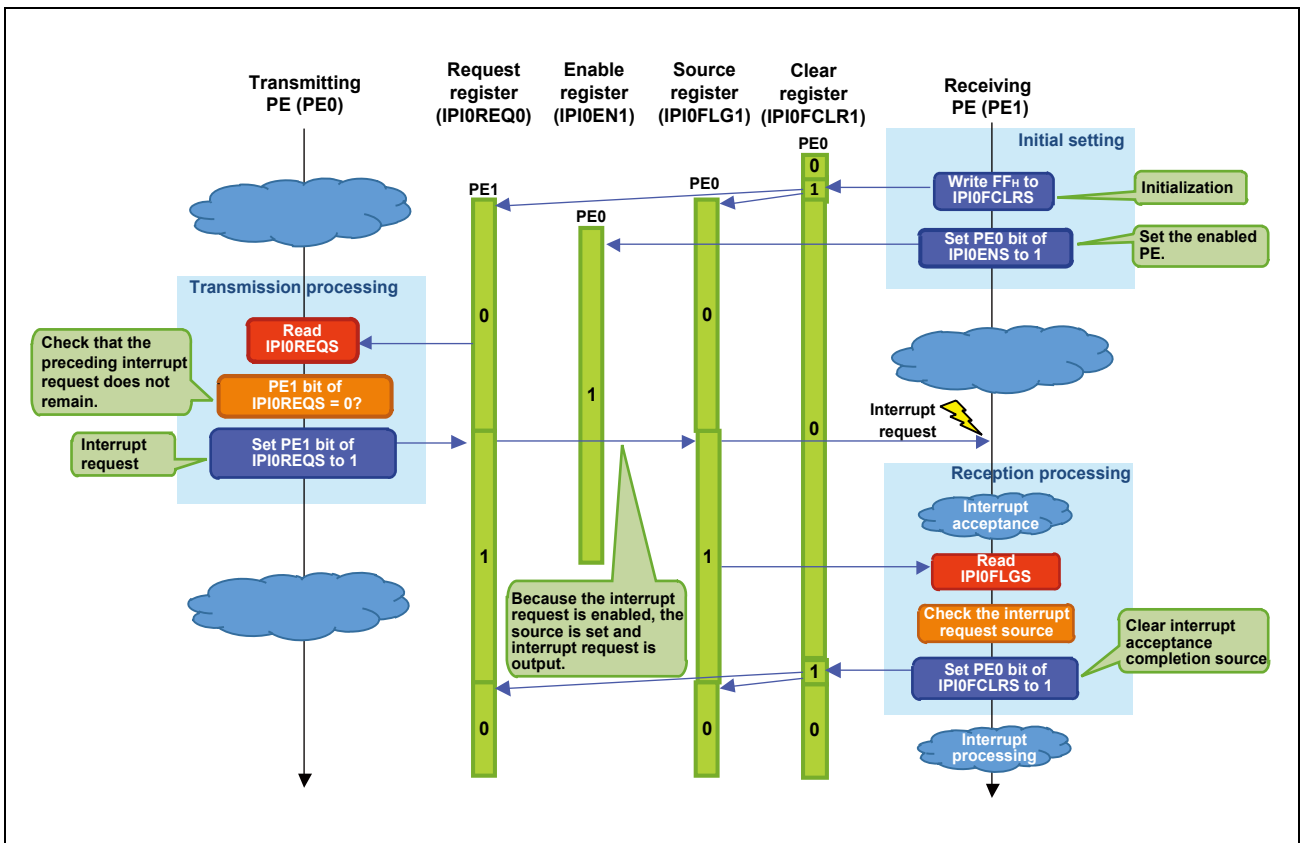


Figure 3.44 Operation Flow of Inter-PE Interrupt

(1) Sample Code**(a) Transmission Processing**

Figure 3.45 shows a sample flowchart of the inter-PE interrupt request transmission processing using IPIR channel 0.

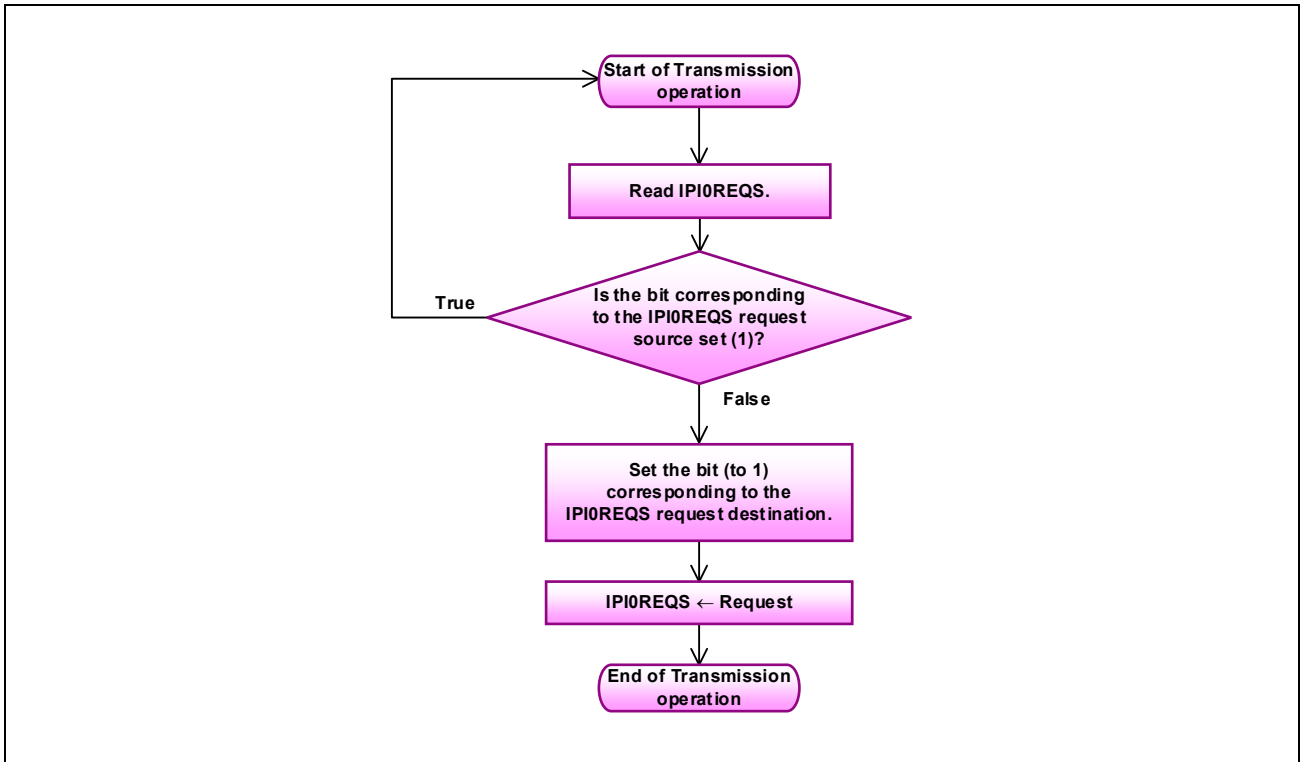


Figure 3.45 Flowchart of Inter-PE Interrupt Request Transmission Processing

(b) Reception Processing

Figure 3.46 shows a sample flowchart of the inter-PE interrupt request reception processing using IPIR channel 0. To prevent source flag loss, the source flag & source clear register configuration is used.

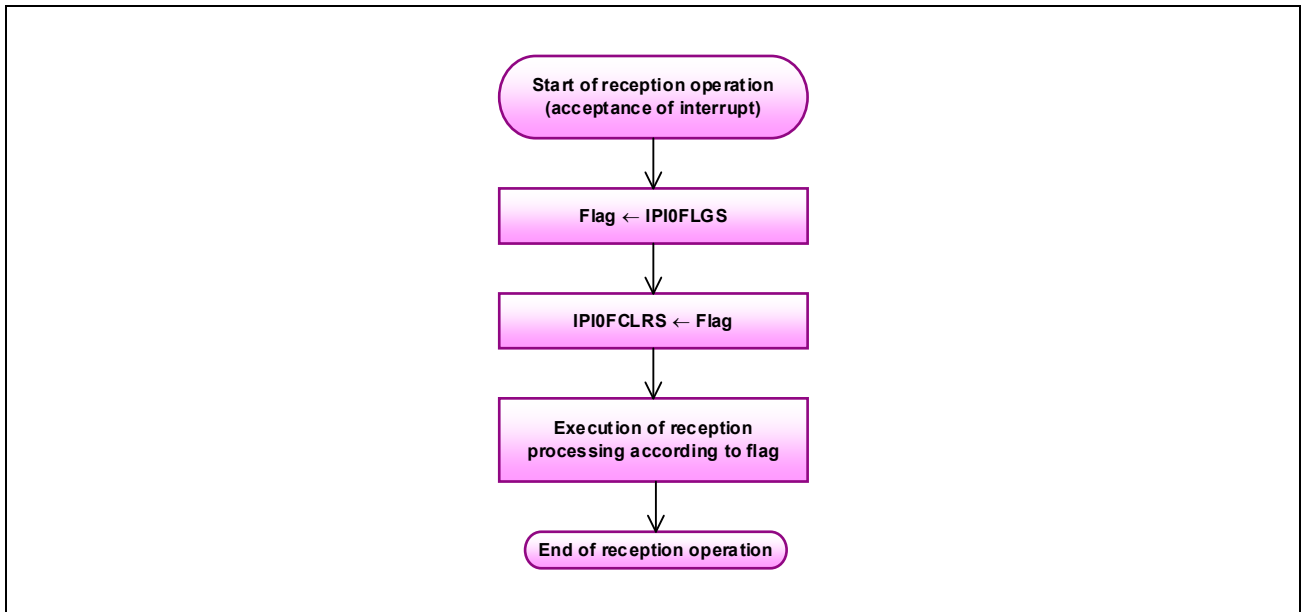


Figure 3.46 Flowchart of Inter-PE Interrupt Request Transmission Processing

3.4.3.3 Request Mask Function

Inter-PE interrupt requests disabled by the IPInENm register setting are ignored. **Figure 3.47** shows an example of the operation when an inter-PE interrupt is requested from a disabled PE. In this figure, bits PE4 to 7 of each register are omitted. In this example, the initial setting is set to IPI0EN1 in advance to prohibit inter-PE interrupt requests from PE2 to PE1. For an operation example of the initial setting, see **Section 3.4.3.1, Initial Setting**.

Even if PE2 sets IPI0REQS[1] (= IPI0REQ2[1]) to 1_B to request PE1 to issue an inter-PE interrupt, inter-PE interrupts from PE2 to PE1 are prohibited, so the IPInFLG1[2] bit remains 0_B and no interrupt request signal is output.

If the transmitting PE disabled by the IPInENm register setting unintentionally writes 1_B to the IPI0REQm register, the IPI0REQm register can be cleared by using the IPInRCLRm register. For details, see **Section 3.4.3.5, Inter-PE Interrupt Request Clear Function**.

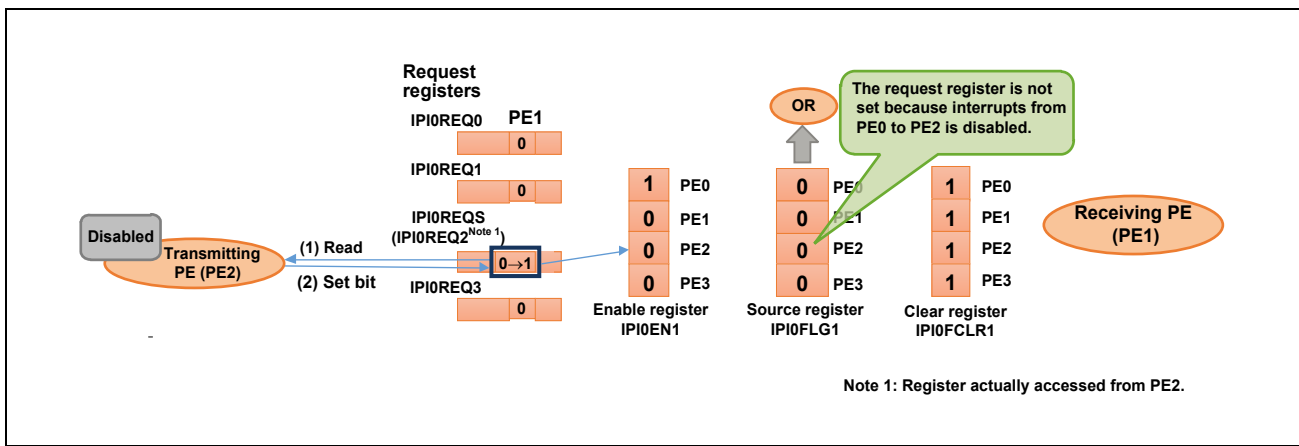


Figure 3.47 Request Mask Operation Example

3.4.3.4 Inter-PE Interrupt Requests of Multiple Systems

If the combination of the transmitting PE and receiving PE differs, multiple inter-PE interrupt requests can be output simultaneously on one channel of IPIR. For example it is possible to output inter-PE interrupt request A (PE0 to PE1) and inter-PE interrupt request B (PE1 to PE0) on channel 1 of IPIR simultaneously. Each register operates independently when the combination of the transmitting PE and receiving PE differs, so multiple inter-PE interrupt requests can be output simultaneously.

3.4.3.5 Inter-PE Interrupt Request Clear Function

The $IPInREQm[x]$ bit can be cleared by setting the $IPInRCLRm[x]$ bit to 1_B , allowing you to cancel inter-PE interrupt requests from the transmitting PE. If inter-PE interrupt requests from the transmitting PE are enabled by the $IPInENx[m]$ bit, the $IPInFLGx[m]$ bit is also cleared at the same time as the $IPInREQm[x]$ bit is cleared. However, the receiving PE may in some cases accept the inter-PE interrupt while the cancellation processing takes place.

Figure 3.48 shows an example of the operation to clear inter-PE interrupt requests from PE0 and PE2 to PE1. When PE0 writes 1_B to the $IPIORCLRS[1]$ (= $IPIORCLR0[1]$) bit, the $IPIOREQ0[1]$ bit is cleared. Moreover, because inter-PE interrupt requests from PE0 to PE1 are enabled by the $IPIOEN1$ register setting, $IPIOFLG1[0]$ is also cleared.

When PE2 writes 1_B to the $IPIORCLRS[1]$ (= $IPIORCLR2[1]$) bit, the $IPIOREQ2[1]$ bit is cleared. Here, inter-PE interrupt requests from PE2 to PE1 are disabled by the $IPIOEN1$ register setting, so the $IPIORCLR2$ register does not affect the value of the $IPIOFLG1$ register.

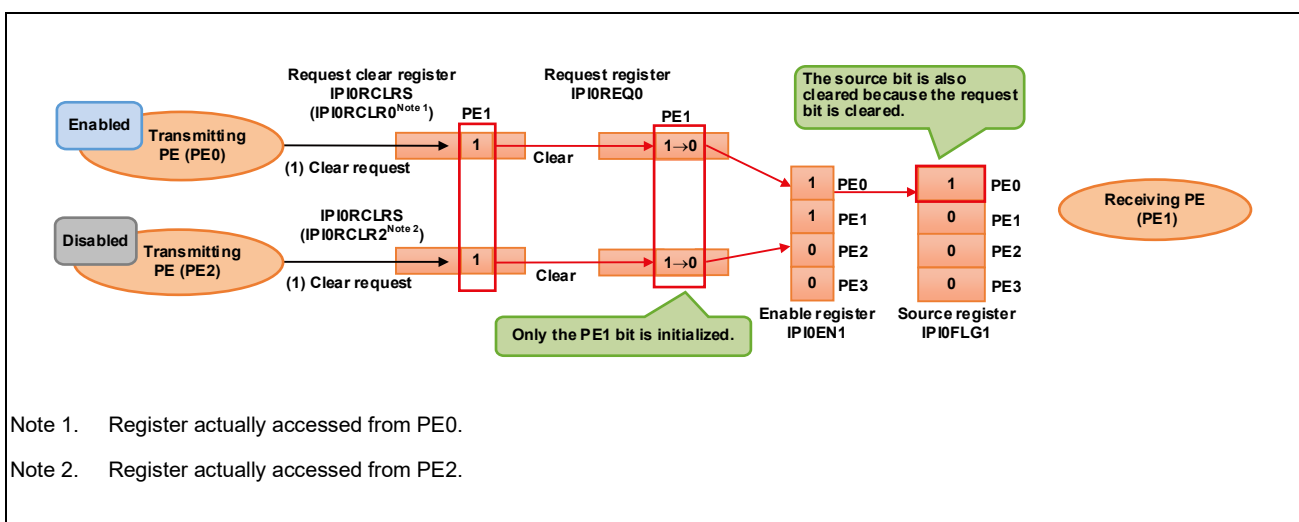


Figure 3.48 Example of Inter-PE Interrupt Request Clear Function Operation

3.5 Mutual Exclusion

This CPU provides instructions that enable shared resources to be controlled mutually exclusively from multiple programs when the system is operating in a multi- processing environment.

When using mutual exclusion, mutual exclusion variables have to be defined in the memory and all programs must operate in accordance with the appropriate instruction flow.

CAUTION

Embedded CPUs in a single-processor configuration use a programming model in which data coherence is maintained by disabling the acknowledgment of maskable interrupts. This is a very easy and sure method of maintaining data coherence, but naturally in a multi-processor, multiple programs might be executing and attempting to use the data at the same time. In this case it is not possible to maintain data coherence simply by disabling maskable interrupt acknowledgment.

3.5.1 Mutual Exclusion Overview

The local RAM, cluster RAM, and the mutual exclusion variable register (MEV) are available as resources for exclusive control. As atomic operation instructions, the instructions of LDL/STC, CAXI, SET1, CLR1, and NOT1 can be performed for the local RAM and cluster RAM, and the instructions of CAXI, SET1, CLR1, and NOT1 can be performed for the mutual exclusion variable register (MEV). These registers are also accessible by the LD and ST instructions but are not regarded as atomic operation.

These MEV registers support exclusive control for variables shared between PEs (common resources).

- Thirty-two 32-bit MEV registers are included.
- 1-, 8-, 16-, and 32-bit accesses are available for each MEV.
- Accesses from all CPU can be made.
- Atomic operation instructions of CAXI, SET1, CLR1, and NOT1 can be performed.

3.5.2 Mutual Exclusion Registers

3.5.2.1 List of Registers

Table 3.147 List of Registers

Module Name	Symbol	Register Name	Address	Access	Access Protection
MEV	G0MEV0	Mutual Exclusion Variable Register 0	<MEV_base> + 00 _H	8, 16, 32	—
	G0MEV1	Mutual Exclusion Variable Register 1	<MEV_base> + 04 _H	8, 16, 32	—
	G0MEV2	Mutual Exclusion Variable Register 2	<MEV_base> + 08 _H	8, 16, 32	—
	G0MEV3	Mutual Exclusion Variable Register 3	<MEV_base> + 0C _H	8, 16, 32	—
	G0MEV4	Mutual Exclusion Variable Register 4	<MEV_base> + 10 _H	8, 16, 32	—
	G0MEV5	Mutual Exclusion Variable Register 5	<MEV_base> + 14 _H	8, 16, 32	—
	G0MEV6	Mutual Exclusion Variable Register 6	<MEV_base> + 18 _H	8, 16, 32	—
	G0MEV7	Mutual Exclusion Variable Register 7	<MEV_base> + 1C _H	8, 16, 32	—
	G0MEV8	Mutual Exclusion Variable Register 8	<MEV_base> + 20 _H	8, 16, 32	—
	G0MEV9	Mutual Exclusion Variable Register 9	<MEV_base> + 24 _H	8, 16, 32	—
	G0MEV10	Mutual Exclusion Variable Register 10	<MEV_base> + 28 _H	8, 16, 32	—
	G0MEV11	Mutual Exclusion Variable Register 11	<MEV_base> + 2C _H	8, 16, 32	—
	G0MEV12	Mutual Exclusion Variable Register 12	<MEV_base> + 30 _H	8, 16, 32	—
	G0MEV13	Mutual Exclusion Variable Register 13	<MEV_base> + 34 _H	8, 16, 32	—
	G0MEV14	Mutual Exclusion Variable Register 14	<MEV_base> + 38 _H	8, 16, 32	—
	G0MEV15	Mutual Exclusion Variable Register 15	<MEV_base> + 3C _H	8, 16, 32	—
	G0MEV16	Mutual Exclusion Variable Register 16	<MEV_base> + 40 _H	8, 16, 32	—
	G0MEV17	Mutual Exclusion Variable Register 17	<MEV_base> + 44 _H	8, 16, 32	—
	G0MEV18	Mutual Exclusion Variable Register 18	<MEV_base> + 48 _H	8, 16, 32	—
	G0MEV19	Mutual Exclusion Variable Register 19	<MEV_base> + 4C _H	8, 16, 32	—
	G0MEV20	Mutual Exclusion Variable Register 20	<MEV_base> + 50 _H	8, 16, 32	—
	G0MEV21	Mutual Exclusion Variable Register 21	<MEV_base> + 54 _H	8, 16, 32	—
	G0MEV22	Mutual Exclusion Variable Register 22	<MEV_base> + 58 _H	8, 16, 32	—
	G0MEV23	Mutual Exclusion Variable Register 23	<MEV_base> + 5C _H	8, 16, 32	—
	G0MEV24	Mutual Exclusion Variable Register 24	<MEV_base> + 60 _H	8, 16, 32	—
	G0MEV25	Mutual Exclusion Variable Register 25	<MEV_base> + 64 _H	8, 16, 32	—
	G0MEV26	Mutual Exclusion Variable Register 26	<MEV_base> + 68 _H	8, 16, 32	—
	G0MEV27	Mutual Exclusion Variable Register 27	<MEV_base> + 6C _H	8, 16, 32	—
	G0MEV28	Mutual Exclusion Variable Register 28	<MEV_base> + 70 _H	8, 16, 32	—
	G0MEV29	Mutual Exclusion Variable Register 29	<MEV_base> + 74 _H	8, 16, 32	—
	G0MEV30	Mutual Exclusion Variable Register 30	<MEV_base> + 78 _H	8, 16, 32	—
	G0MEV31	Mutual Exclusion Variable Register 31	<MEV_base> + 7C _H	8, 16, 32	—

3.5.2.2 G0MEVn — Mutual Exclusion Variable Register n (n = 0 to 31)

The Mutual Exclusive variable Register n (G0MEVn) is a 32-bit read/write register that can be used as a mutual exclusive resource.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	G0MEVn[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	G0MEVn[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.148 G0MEVn Register Contents

Bit Position	Bit Name	Function
31 to 0	G0MEVn[31:0]	Mutual Exclusive Variable. These bits can be read and written from all PEs.

3.5.3 Mutual Exclusion Function

3.5.3.1 Operation Example Using Mutual Exclusion Variable Register (MEV)

An example of a flow chart that does mutual exclusion of a shared resource by spin lock with MEV is given in the following figure. When the value of MEV is 1B in this example, a shared resource shows that it is lock status. It is recommended that reading or writing to MEV be carried out by atomic operation because MEV itself is also a shared resource.

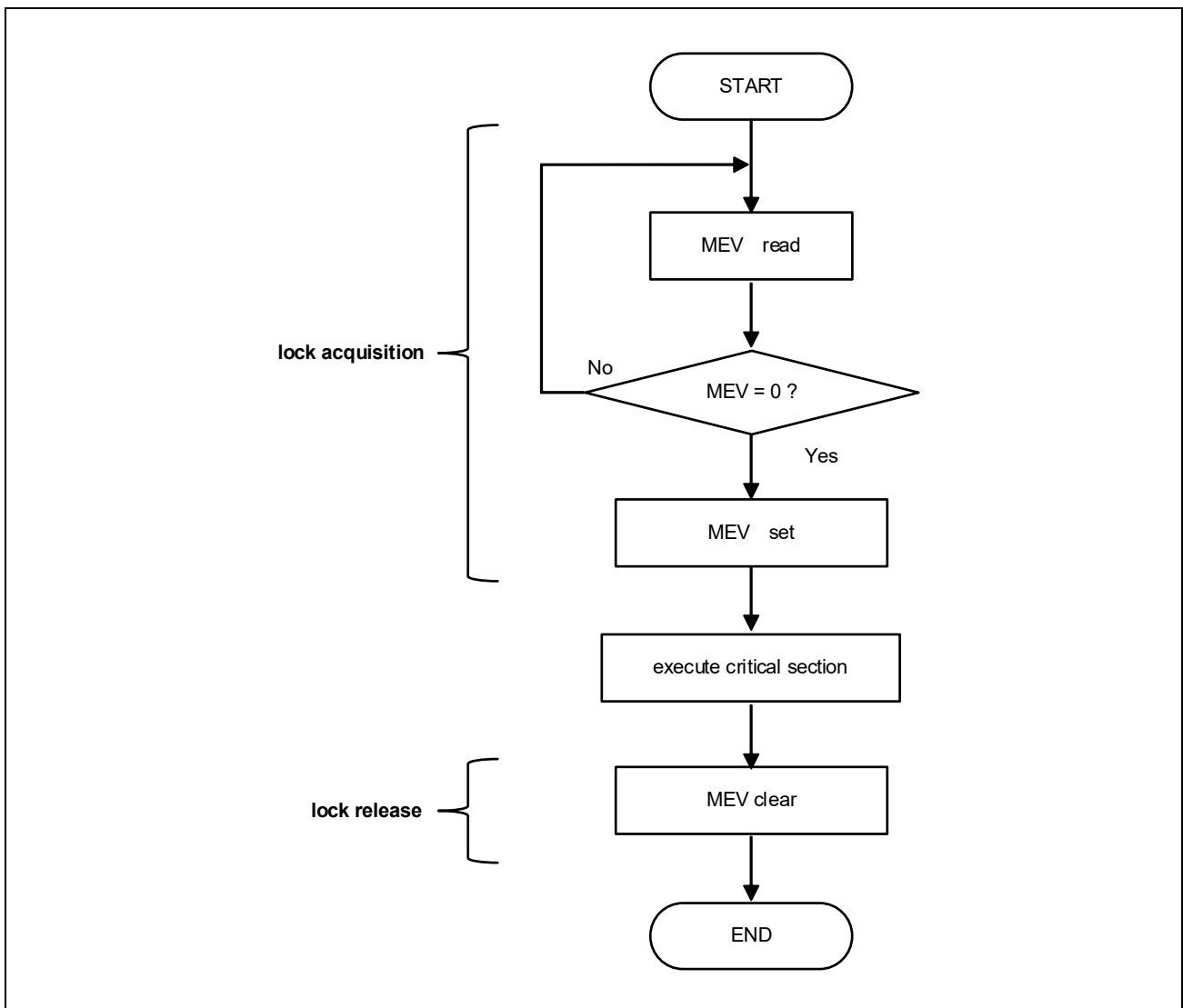


Figure 3.49 Spin Lock Examples of Lock Step Using MEV

A CAXI operation can be used for atomic operation which extends over more than one data column in G4MH. The sample code which does lock acquisition/release using MEV below is indicated. “lock_adr” is equivalent to the address of MEV by this example.

Lock Acquisition

```

MOV      lock_adr, r20
Lock:    MOV      1, r21
         CAXI    [r20], r0, r21
         BZ     Lock_succes
         SNOOZE
         BR     Lock
Lock_succes:

```

Lock Release

```

MOV      lock_adr, r20
ST.W    r0, 0[r20]

```

Figure 3.50 shows an operation example that does mutual exclusion of a shared resource between PE0 and PE1. The lock status of a shared resource is indicated by the value of MEV that corresponds to each PE, and MEV has lock status at 1_B by this example.

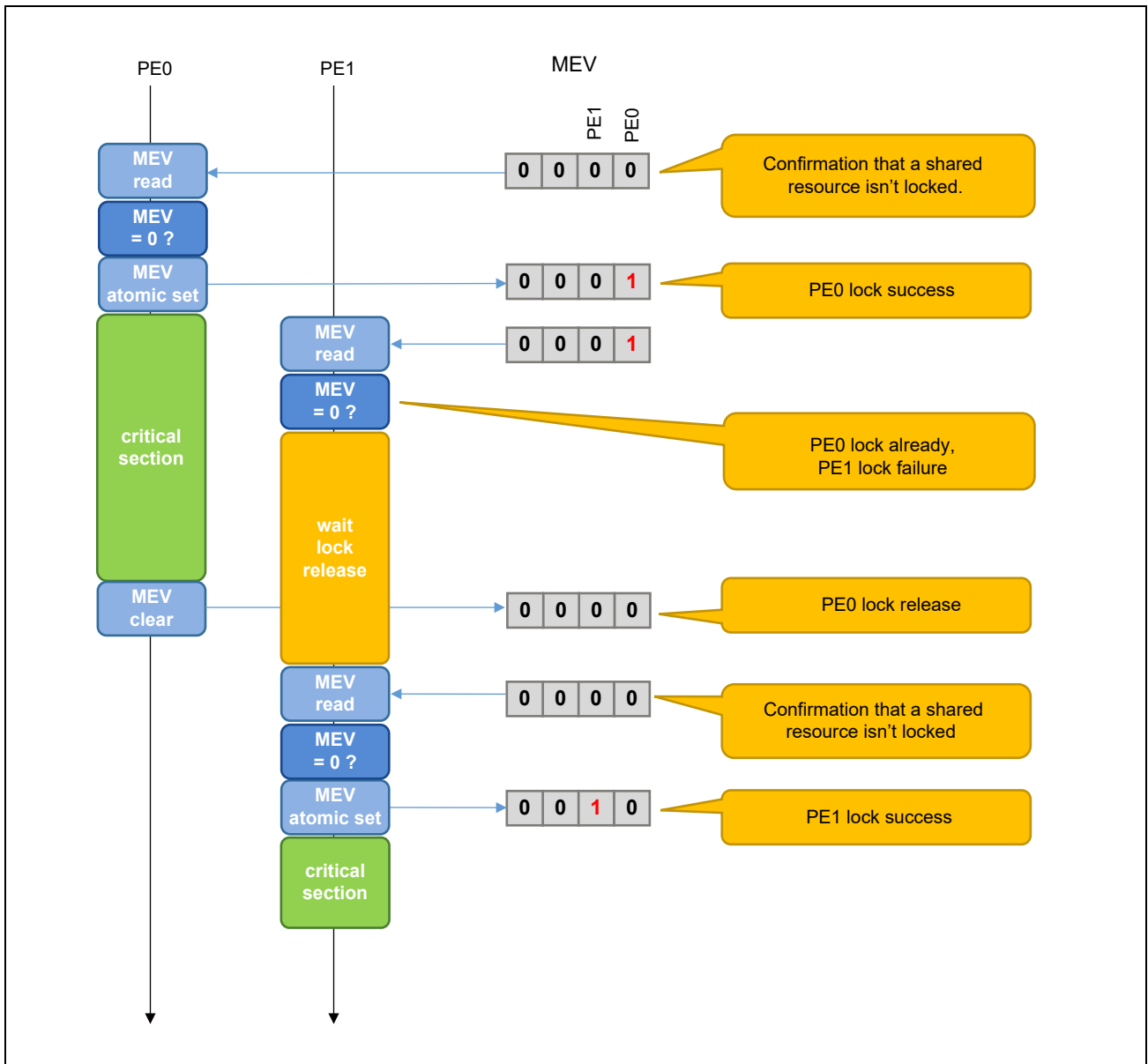


Figure 3.50 MEV Usage Sample

3.5.3.2 Shared Data that does not Require Mutual Exclusion Processing

This CPU maintains data access coherence for the following types of data access even in a multi-processor environment.

- Access in which the data is aligned to the size that matches the data type (aligned access)
 - LD, ST, SLD, SST, LDL, STC, and bit manipulation instructions (TST1)
- Access by using a bit manipulation instruction (SET1, CLR1, or NOT1) (read-modify-write)
- Access by using the CAXI instruction (read-modify-write)

NOTE

Whether an instruction is atomically executed or not depends on the data format. Please refer to **Section 3.2.2.6, Data Types**.

With some exceptions, mutual exclusion is achieved by using these types of data access. In other words, it is guaranteed that while one CPU is executing the instructions that perform the above data accesses, another CPU is not accessing the data. This is known as an instruction being executed atomically or an instruction guaranteeing the atomicity.

Note that the atomic execution of an instruction means that a data access bus transaction completes with no disruption; it does not necessarily mean that a series of transactions has been completed

CAUTION

The extent to which coherency is guaranteed might be limited, depending on the hardware specifications. For example, for some memories, coherency might not be guaranteed even if aligned access is used. For details, see **Section 3.5.1, Mutual Exclusion Overview** and **Section 3.8.3, Accesses to Registers by Bit-Manipulation Instructions**.

3.5.3.3 Operation of the LDL and STC Instructions

The LDL and STC instructions can be used to obtain atomic read-modify-write operations for accurate processing in the updating of memory by multicore systems. The LDL and STC instructions operate as follows. For the operation of the LDL and STC instructions, refer to the RH850G4MH User's Manual: Software.

(1) Link

Only one link (LLbit) can be created. The link includes information on the address for which it was created, and control is applied according to whether an STC instruction succeeds or fails at this address and whether the link is lost. The link also includes the data size information when it the link is created and therefore any STC instruction which has a data size different from that of the LDL instruction that created a link always fails, and the link is deleted.

(2) Link generation

The CPU is capable of generating a link to the local RAM and the cluster RAM. Executing the LDL instruction on the target RAM for the operation leads to the link address being registered, the link flag being set, and a link being generated in response to reading by the instruction.

- (a) The local RAM for the given processor
- (b) The cluster RAM

The CPU is capable of generating a link to either (a) or (b).

(3) Success in storing

After a link has been generated, storing will only proceed in response to executing an STC instruction corresponding to the generated link.

(4) Failure in storing

If a link is lost, storing does not proceed even when an STC instruction for the corresponding address is processed. Storing also does not proceed when an STC instruction that does not correspond to the link is processed.

(5) Condition for successful storing

If the following condition is met, the STC instruction is judged to be for the address corresponding to the link.

- The address and size for the LDL instruction that generated the link matches that for the STC instruction.

(6) Loss of the link

A link is lost when certain event or address conditions are satisfied. **Table 3.149** shows the link loss conditions. A link is lost if any of the conditions shown in this table satisfied.

Table 3.149 Link Loss Conditions

Event Condition	Remark
Store operation to a 32-byte-aligned address range that includes the address of the existing link*1	The following operations by both the CPU for which the link was generated and another bus master correspond. <ul style="list-style-type: none"> • ST, SST, STC, and STV instructions • SET1, NOT1, CLR1, and CAXI instructions • PREPARE and PUSHSP instructions
Execution of STC instruction for a location corresponding to the existing link	The corresponding link (for (1) or (2) above) will be lost whether the result of the instruction was success or fail.
Execution of LDL instruction for a location corresponding to the existing link	The link generated in response to the preceding LDL instruction will be lost, and the link in response to the following LDL instruction is generated.
Execution of CLL instruction	
Exception acknowledgment	
Execution of EIRET instruction	
Execution of FERET instruction	
Bus access error of LDL	

Note 1. In the local RAM, if the store instruction except STC/CAXI instruction is executed, the Link is not always lost. Therefore, use the program flow which does not need that these instructions cause the Link Loss. For example, in the sample code (7), after reading the Lock variable using the LDL instruction, by executing the STC instruction only if nobody has the Lock, the program flow does not need the Link Loss caused by the store instruction for the Lock Release.

(7) Sample Code

The sample code of a spinlock executed by using the LDL.W and STC.W instructions is shown below.

Lock Acquisition

```

        MOV        lock_adr, r20
Lock:   LDL.W     [r20], r21
        CMP        r0, r21
        BNZ        Lock_wait
        MOV        1, r21
        STC.W     r21, [r20]
        CMP        r0, r21
        BNZ        Lock_success
Lock_wait:
        SNOOZE
        BR         Lock
Lock_success:

```

Lock Release

```

        ST.W     r0, 0[r20]

```

3.5.3.4 Performing Mutual Exclusion by Using the SET1 Instruction

The SET1 instruction can be used to perform mutual exclusion over multiple data arrays. By executing the SET1 instruction on the same bit in the memory and then checking the PSW.Z flag, which indicates the execution result, it can be determined whether lock acquisition succeeded or failed.

CAUTIONS

1. Depending on the hardware specifications, the system performance might drop if exclusive control is executed frequently by using the SET1 instruction, because this causes the bus to be occupied for a long time. It is therefore recommended to execute exclusive control by using the LDL/STC instructions as much as possible.
2. When performing mutual exclusion by using the SET1 instruction, to prevent the problem of excessive bus occupancy described in Caution 1 above, execute the SNOOZE instruction before attempting to acquire a lock again after lock acquisition has failed, and adjust the lock acquisition loop execution interval.

(1) Sample Code

The sample code of a spinlock executed by using the SET1 instruction is shown below.

Lock Acquisition

```

MOV      lock_adr, r20
Lock:    SET1    0, 0[r20]
        BZ      Lock_success
        SNOOZE
        BR      Lock
Lock_success:

```

Lock Release

```

CLR1    0, 0[r20]

```

3.5.3.5 Performing Mutual Exclusion by Using the CAXI Instruction

The CAXI instruction can be used to perform mutual exclusion over multiple data arrays. By executing the CAXI instruction on the same word in the memory and then checking the destination register, it can be determined whether lock acquisition succeeded or failed.

CAUTIONS

1. Depending on the hardware specifications, the system performance might drop if exclusive control is executed frequently by using the CAXI instruction, because this causes the bus to be occupied for a long time. It is therefore recommended to execute exclusive control by using the LDL/STC instructions as much as possible.
2. When performing mutual exclusion by using the CAXI instruction, to prevent the problem of excessive bus occupancy described in Caution 1 above, execute the SNOOZE instruction before attempting to acquire a lock again after lock acquisition has failed, and adjust the lock acquisition loop execution interval.

(1) Sample Code

The sample code of a spinlock executed by using the CAXI instruction is shown below.

Lock Acquisition

```

        MOV        lock_adr, r20
Lock:   MOV        1, r21
        CAXI       [r20], r0, r21
        BZ         Lock_success
        SNOOZE
        BR         Lock
Lock_success:

```

Lock Release

```

        ST.W       r0, 0[r20]

```

3.6 Barrier-Synchronization

3.6.1 Barrier-Synchronization Overview

When parallel processing is performed using a multicore processor, it may be necessary to make the cores wait until the data required for the next processing is ready, at which point the processing can proceed to the next processing.

Typically, control that sets up a barrier that prevents the next process from starting before all the applicable cores complete the predetermined processing is called barrier synchronization. It is possible to implement barrier synchronization just using software, but this may degrade system processing performance due to the concentration of barrier arrival notifications from the various cores and memory access from the various cores for verification purposes.

Barrier synchronization registers enable easy barrier synchronization by detecting whether all the applicable cores have completed the required processing through the reception of completion notifications from the various PEs, while also providing a barrier synchronization support function that automatically clears completion notifications from the various PEs in preparation for the next barrier synchronization. This function makes it possible to reduce the number of memory accesses for barrier synchronization, thereby minimizing the negative impact of barrier synchronization on processing performance. This function can also be used for barrier synchronization between PEs forming a cluster.

The barrier synchronization register has the following features.

- The 16-ch barrier synchronization registers are provided.
- Barrier synchronization can be implemented using the same code for all the cores.
- Accessible from all clusters and all PEs within the system

3.6.2 Barrier-Synchronization Registers

3.6.2.1 List of Registers

Table 3.150 List of Registers

Module Name	Register Name	Register Symbol	Address	Access	Access Protection
BARR	Barrier-Synchronization n*2 Barrier Participating PE Setting Register	BRnEN	<BARR_base> + 004 _H + 010 _H × n	8	—
	Barrier-Synchronization n*2 Initialization Register	BRnINIT	<BARR_base> + 000 _H + 010 _H × n	8	—
	Barrier-Synchronization n*2 Barrier Check Register Self*1	BRnCHKS	<BARR_base> + 100 _H + 010 _H × n	8	—
	Barrier-Synchronization n*2 Barrier Synchronization Completion Register Self*1	BRnSYNCS	<BARR_base> + 104 _H + 010 _H × n	8	—
	Barrier-Synchronization n*2 Barrier Check Register m*3	BRnCHKm	<BARR_base> + 800 _H + 010 _H × n + 100 _H × m	8	—
	Barrier-Synchronization n*2 Barrier Synchronization Completion Register m*3	BRnSYNCSm	<BARR_base> + 804 _H + 010 _H × n + 100 _H × m	8	—

Note 1. When PE accesses the self register, PE can access the corresponding register for each PE. For example, when PE1 accesses BR0CHKS register, PE1 can access BR0CHK1 register.

Note 2. n = 0 to 15. n is the Barrier-Synchronization channel number.

Note 3. m = 0 to 1. m is the value of PEID.

3.6.2.2 Self Region

The self region contains the following two types of registers.

- BRnCHKS — Barrier-Synchronization n - Barrier Check Self-Register
- BRnSYNCS — Barrier-Synchronization n – Barrier Synchronization Completion Self-Register

The self registers are virtual registers that do not physically exist. Access from each PE to self registers is routed to the actual register corresponding to the access source PE. **Table 3.151** lists the access source PEs and routing destination registers. For the functions of each register bit, refer to the specifications of the routing destination register. When masters except PEx access the self registers, the register returns 0, and write access is ignored.

It is basically assumed that the barrier synchronization registers are accessed from PEs via self region when PEs use the barrier synchronization function. This allows PEs to use same code because it is not necessary to specify different register address for each PE.

It is also possible to directly access registers by specifying the address of the actual register without using the self register. In this case, the purpose is assumed to check the status of registers for another PE, or referencing of individual registers by a debugging tool.

Table 3.151 Self Region Register Routing List

Self Register	Access Source PE	
	PE0	PE1
BRnCHKS	BRnCHK0	BRnCHK1
BRnSYNCS	BRnSYNCS0	BRnSYNCS1

3.6.2.3 BRnINIT — Barrier-Synchronization n Initialization Register

The Barrier-Synchronization n Initialization Register (BRnINIT) is an 8-bit write only register that initializes the Barrier-Synchronization channel n ($n = 0$ to 15). This register clears the BRnCHKm register and BRnSYNCm register of channel n.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	BRINIT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 3.152 BRnINIT Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	BRINIT	Barrier-Synchronization n initialize Writing 1 to this bit clears registers BRnCHK0 to 1 and registers BRnSYNC0 to 1 of the same channel. This bit always returns 0 when read.

3.6.2.4 BRnEN — Barrier-Synchronization n Barrier Participating PE Setting Register

The Barrier-Synchronization n Barrier Participating PE Setting Register (BRnEN) is an 8-bit read/write register that enables Barrier-Synchronization Participation in channel n for PEs.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	BREN[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 3.153 BRnEN Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1 to 0	BREN[1:0]	Barrier-Synchronization Participation enable for channel n Writing 1 to the x-th bit enables the barrier synchronization support function for PEx. Writing 0 to the x-th bit disables the barrier synchronization support function for PEx. Bit 0: PE0 Bit 1: PE1

3.6.2.5 BRnCHKm — Barrier-Synchronization n Barrier Check Register m

The Barrier-Synchronization n Barrier Check Register (BRnCHKm) is an 8-bit read/write register that is used for PEm to notify that PEm has arrived at the barrier of channel n. The PE that has arrived at the barrier is to write the BRCHK bit of this register.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	BRCHK
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 3.154 BRnCHKm Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	BRCHK	Barrier Check Bit of channel n This bit can be updated by both software and hardware. Software: When the BRnCHKm register is written, the value of this bit becoming 1 regardless of the writing value. This bit can be set even if the BRnEN.BRENm bit is 0, but it does not affect the barrier - synchronization of channel n. Hardware: This bit is cleared by the hardware when the barrier-synchronization is established at the condition that all the BRnCHKm.BRCHK bits of participating PEs, which are enabled by the BRnEN register, are set. At the same time, the BRnSYNCm.BRSYNC bits of all participating PEs are set. This bit is cleared by writing 1 to the BRnINIT.BRINIT bit.

NOTE

If all bits of the BRnEN register are 0 (no PE participates the barrier synchronization), BRCHK bit cannot be set.

3.6.2.6 BRnSYNCm — Barrier-Synchronization n Barrier Synchronization Completion Register m

The Barrier-Synchronization n Barrier Synchronization Completion Register is an 8-bit read/write register that indicates that all the PEs that participate in the barrier of channel n have arrived at the barrier of channel n.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	BRSYNC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 3.155 BRnSYNCm Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	BRSYNC	Barrier synchronization completion This bit can be updated by both software and hardware. Software: When the BRnSYNCm register is written, this bit is updated with the write data. Hardware: If the BRnEN.BRENm bit is 1, this bit is automatically set when the barrier-synchronization is established at the condition that all the BRnCHKm.BRCHK bits of participating PEs, which are enabled by the BRnEN register, are set. If the BRnEN.BRENm bit is 0, this bit is not updated by the hardware. This bit is cleared by writing 1 to the BRnINIT.BRINIT bit.

3.6.3 Barrier-Synchronization Function

This product has only PE0 and PE1, but this subsection includes examples of PE0 to PE3.

3.6.3.1 Initial Setting

Before the barrier synchronization register is used, initial settings must be made in the BRnEN register. **Figure 3.51** shows an example of the initial setting of synchronization register channel 0 by PE0.

First, to clear the setting of the previous operation, the BR0EN register is cleared by write operation. After clearing of the BR0EN register, write 07H to the BR0EN register to participate PE0, PE1 and PE2 in barrier synchronization. Lastly, PE0 writes 1B to the BR0INIT register to clear registers BROCHK0 to 3 and registers BR0SYNC0 to 3. Upon completion of setting of the participating PE setting register and initialization, barrier synchronization using the barrier synchronization register is possible.

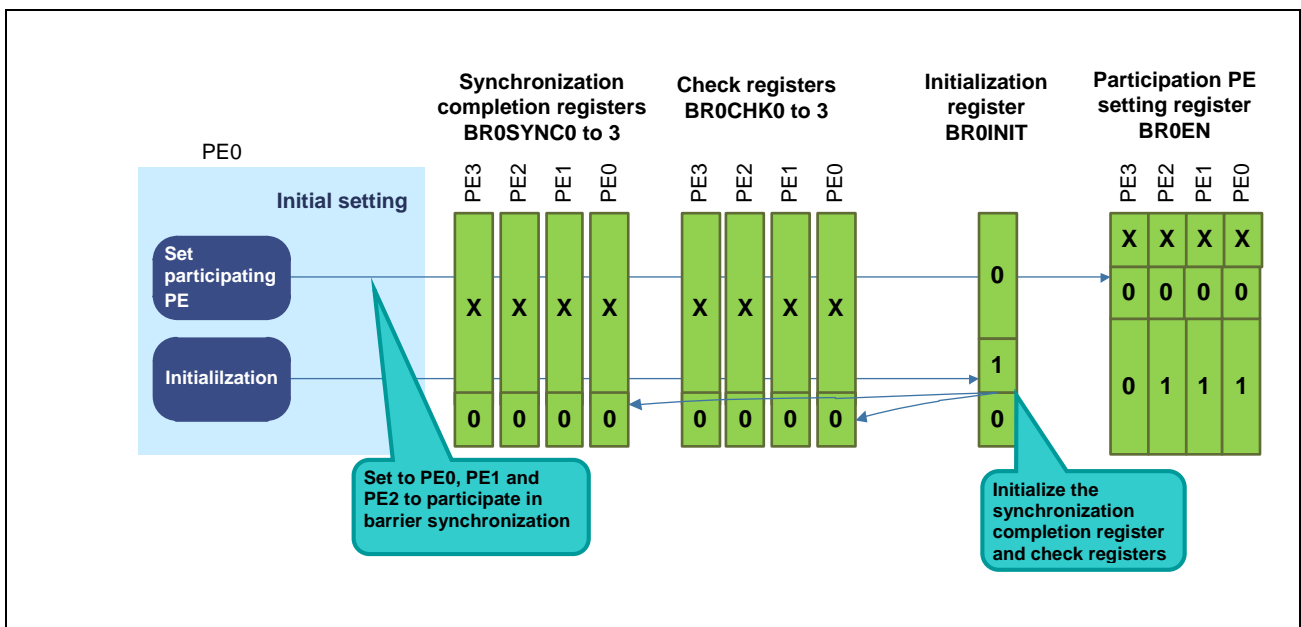


Figure 3.51 Example of Initial Setting of Barrier Synchronization Register

3.6.3.2 Barrier Synchronization

Figure 3.52 shows an example of the operation executing barrier synchronization of PE0, PE1, and PE2 using barrier synchronization register channel 0. In this example, PE0, PE1, and PE2 are set by the BR0EN register in advance to participate in barrier synchronization by the initial setting processing described in **Section 3.4.3.1, Initial Setting**.

When PE0, PE1, and PE2, which are set to participate in barrier synchronization by the BR0EN register, arrive at the barrier, they each write data to the BR0CHKS register, setting it to 1_B. Note that the BR0CHKS register is set to 1_B when it is written to, regardless of the write value. Each of the PEs, upon arriving at the barrier, polls the BR0SYNCS register and waits for all the other PEs participating in barrier synchronization to arrive at the barrier. When all the PEs participating in barrier synchronization arrive at the barrier and the values of registers BR0CHK0 to 2 have all become 1_B, registers BR0CHK0 to 3 are automatically cleared, and 1_B is automatically set to registers BR0SYNCS0 to 2 (= BR0SYNCS). When a PE that arrives at the barrier detects that the BR0SYNCS register has been set to 1_B, it considers that all the PEs participating in barrier synchronization have arrived at the barrier, and it clears the respective BR0SYNCS registers to 0_B to allow transition to the next processing. When the values of registers BR0CHK0 to 2 have all become 1_B, registers BR0CHK0 to 3 are automatically cleared, so each PE is able to move on to the next cycle as soon as the BR0SYNCS register is cleared.

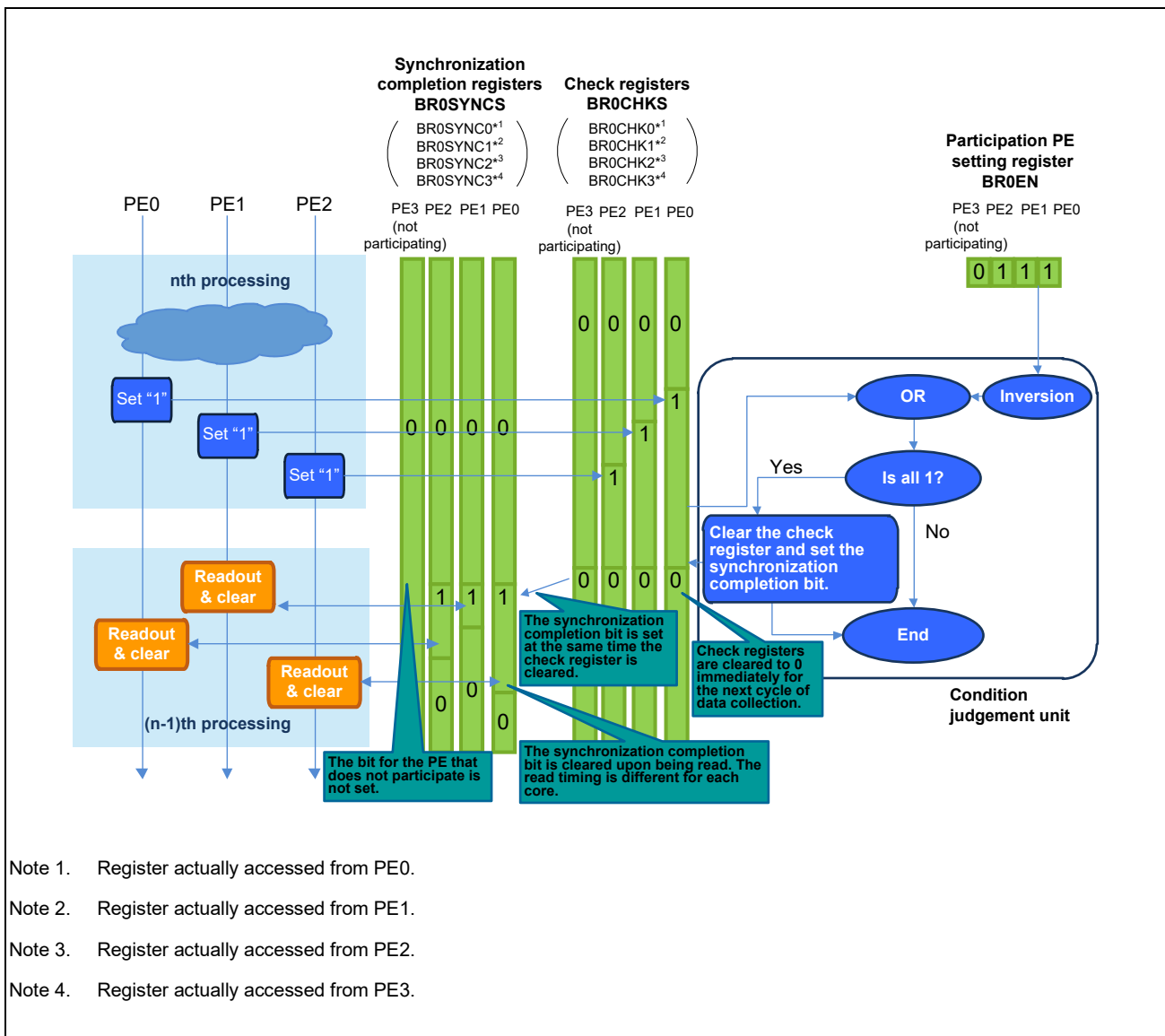


Figure 3.52 Example of Operation of Barrier Synchronization Using Barrier Synchronization Register

3.7 TPTM

3.7.1 TPTM Overview

Time Protection Timer (TPTM) is sets of CPU dedicated timers used to achieve high performance and functionality of timing protection. One timer set is assigned to each CPU respectively.

TPTM has the following features.

- Interval timer × 4ch (down counter)
- Free-run timer × 1ch (up counter)
- Start, Stop and Restart of counter for both of the interval timers and the free-run timer.
- Divided counter of the timers can be configured for the interval timers and the free-run timer respectively.
- Simultaneous count control for the interval timers.
- Underflow interrupt for the interval timers.

3.7.2 TPTM Registers

3.7.2.1 List of Registers

Table 3.156 List of Registers (1/2)

Module Name	Register Name	Register Symbol	Address	Access	Access Protection
TPTM self*1	Counter Start Register of Interval Timer Self	TPTMSIRUN	<TPTM_base> + 000 _H	8, 16, 32	—
	Counter Restart Register of Interval Timer Self	TPTMSIRRUN	<TPTM_base> + 004 _H	8, 16, 32	—
	Counter Stop Register of Interval Timer Self	TPTMSISTP	<TPTM_base> + 008 _H	8, 16, 32	—
	Counter Status Register of Interval Timer Self	TPTMSISTR	<TPTM_base> + 00C _H	8, 16, 32	—
	Interrupt Enable Register of Interval Timer Self	TPTMSIEN	<TPTM_base> + 010 _H	8, 16, 32	—
	Underflow Status Register of Interval Timer Self	TPTMSIISTR	<TPTM_base> + 014 _H	8, 16, 32	—
	Divider Register of Interval Timer Self	TPTMSIDIV	<TPTM_base> + 018 _H	8, 16, 32	—
	Counter Start Register of Free-run Timer Self	TPTMSFRUN	<TPTM_base> + 020 _H	8, 16, 32	—
	Counter Restart Register of Free-run Timer Self	TPTMSFRUN	<TPTM_base> + 024 _H	8, 16, 32	—
	Counter Stop Register of Free-run Timer Self	TPTMSFSTP	<TPTM_base> + 028 _H	8, 16, 32	—
	Counter Status Register of Free-run Timer Self	TPTMSFSTR	<TPTM_base> + 02C _H	8, 16, 32	—
	Divider Register of Free-run Timer Self	TPTMSFDIV	<TPTM_base> + 030 _H	8, 16, 32	—
	Counter Channel 0 Register of Interval Timer Self	TPTMSICNT0	<TPTM_base> + 080 _H	32	—
	Load Channel 0 Register of Interval Timer Self	TPTMSILD0	<TPTM_base> + 084 _H	32	—
	Counter Channel 1 Register of Interval Timer Self	TPTMSICNT1	<TPTM_base> + 088 _H	32	—
	Load Channel 1 Register of Interval Timer Self	TPTMSILD1	<TPTM_base> + 08C _H	32	—
	Counter Channel 2 Register of Interval Timer Self	TPTMSICNT2	<TPTM_base> + 090 _H	32	—
	Load Channel 2 Register of Interval Timer Self	TPTMSILD2	<TPTM_base> + 094 _H	32	—
	Counter Channel 3 Register of Interval Timer Self	TPTMSICNT3	<TPTM_base> + 098 _H	32	—
	Load Channel 3 Register of Interval Timer Self	TPTMSILD3	<TPTM_base> + 09C _H	32	—
Counter Register of Free-run Timer Self	TPTMSFCNT	<TPTM_base> + 0A0 _H	32	—	
TPTM PEn*2	Counter Start Register of Interval Timer for PEn	TPTMnIRUN	<TPTM_base> + 100 _H + (100 _H × n)	8, 16, 32	—
	Counter Restart Register of Interval Timer for PEn	TPTMnIRRUN	<TPTM_base> + 104 _H + (100 _H × n)	8, 16, 32	—
	Counter Stop Register of Interval Timer for PEn	TPTMnISTP	<TPTM_base> + 108 _H + (100 _H × n)	8, 16, 32	—
	Counter Status Register of Interval Timer for PEn	TPTMnISTR	<TPTM_base> + 10C _H + (100 _H × n)	8, 16, 32	—
	Interrupt Enable Register of Interval Timer for PEn	TPTMnIEN	<TPTM_base> + 110 _H + (100 _H × n)	8, 16, 32	—
	Underflow Status Register of Interval Timer for PEn	TPTMnIISTR	<TPTM_base> + 114 _H + (100 _H × n)	8, 16, 32	—

Table 3.156 List of Registers (2/2)

Module Name	Register Name	Register Symbol	Address	Access	Access Protection
TPTM PEn*2	Divider Register of Interval Timer for PEn	TPTMnIDIV	<TPTM_base> + 118 _H + (100 _H × n)	8, 16, 32	—
	Counter Start Register of Free-run Timer for PEn	TPTMnFRUN	<TPTM_base> + 120 _H + (100 _H × n)	8, 16, 32	—
	Counter Restart Register of Free-run Timer for PEn	TPTMnFRRUN	<TPTM_base> + 124 _H + (100 _H × n)	8, 16, 32	—
	Counter Stop Register of Free-run Timer for PEn	TPTMnFSTP	<TPTM_base> + 128 _H + (100 _H × n)	8, 16, 32	—
	Counter Status Register of Free-run Timer for PEn	TPTMnFSTR	<TPTM_base> + 12C _H + (100 _H × n)	8, 16, 32	—
	Divider Register of Free-run Timer for PEn	TPTMnFDIV	<TPTM_base> + 130 _H + (100 _H × n)	8, 16, 32	—
	Counter Channel 0 Register of Interval Timer for PEn	TPTMnCINT0	<TPTM_base> + 180 _H + (100 _H × n)	32	—
	Load Channel 0 Register of Interval Timer for PEn	TPTMnILD0	<TPTM_base> + 184 _H + (100 _H × n)	32	—
	Counter Channel 1 Register of Interval Timer for PEn	TPTMnCINT1	<TPTM_base> + 188 _H + (100 _H × n)	32	—
	Load Channel 1 Register of Interval Timer for PEn	TPTMnILD1	<TPTM_base> + 18C _H + (100 _H × n)	32	—
	Counter Channel 2 Register of Interval Timer for PEn	TPTMnCINT2	<TPTM_base> + 190 _H + (100 _H × n)	32	—
	Load Channel 2 Register of Interval Timer for PEn	TPTMnILD2	<TPTM_base> + 194 _H + (100 _H × n)	32	—
	Counter Channel 3 Register of Interval Timer for PEn	TPTMnCINT3	<TPTM_base> + 198 _H + (100 _H × n)	32	—
	Load Channel 3 Register of Interval Timer for PEn	TPTMnILD3	<TPTM_base> + 19C _H + (100 _H × n)	32	—
	Counter Register of Free-run Timer for PEn	TPTMnFCNT	<TPTM_base> + 1A0 _H + (100 _H × n)	32	—

Note 1. When PE accesses the self register, PE can access the corresponding register for each PE. For example, when PE1 accesses TPTMSIRUN register, PE1 can access the TPTM1IRUN register.

Note 2. n = 0 to 1. n is the PEID number.

3.7.2.2 Self Region

The self region contains the following registers.

- TPTMSIRUN — Counter Start Register of Interval Timer
- TPTMSIRRUN — Counter Restart Register of Interval Timer
- TPTMSISTP — Counter Stop Register of Interval Timer
- TPTMSISTR — Counter Status Register of Interval Timer
- TPTMSIEN — Interrupt Enable Register of Interval Timer
- TPTMSIISTR — Underflow Status Register of Interval Timer
- TPTMSIDIV — Divider Register of Interval Timer
- TPTMSFRUN — Counter Start Register of Free-run Timer
- TPTMSFRRUN — Counter Restart Register of Free-run Timer
- TPTMSFSTP — Counter Stop Register of Free-run Timer
- TPTMSFSTR — Counter Status Register of Interval Timer
- TPTMSFDIV — Divider Register of Free-run Timer
- TPTMSICNT0 — Counter Channel 0 Register of Interval Timer
- TPTMSILD0 — Load Channel 0 Register of Interval Timer
- TPTMSICNT1 — Counter Channel 1 Register of Interval Timer
- TPTMSILD1 — Load Channel 1 Register of Interval Timer
- TPTMSICNT2 — Counter Channel 2 Register of Interval Timer
- TPTMSILD2 — Load Channel 2 Register of Interval Timer
- TPTMSICNT3 — Counter Channel 3 Register of Interval Timer
- TPTMSILD3 — Load Channel 3 Register of Interval Timer
- TPTMSFCNT — Counter Register of Free-run Timer

Table 3.157 Self Region Register Routing List

Self Region Register	Access source PE	
	PE0	PE1
TPTMSIRUN	TPTM0IRUN	TPTM1IRUN
TPTMSIRRUN	TPTM0IRRUN	TPTM1IRRUN
TPTMSISTP	TPTM0ISTP	TPTM1ISTP
TPTMSISTR	TPTM0ISTR	TPTM1ISTR
TPTMSIEN	TPTM0IEN	TPTM1IEN
TPTMSIUSTR	TPTM0IUSTR	TPTM1IUSTR
TPTMSIDIV	TPTM0IDIV	TPTM1IDIV
TPTMSFRUN	TPTM0FRUN	TPTM1FRUN
TPTMSFRRUN	TPTM0FRRUN	TPTM1FRRUN
TPTMSFSTP	TPTM0FSTP	TPTM1FSTP
TPTMSFSTR	TPTM0FSTR	TPTM1FSTR
TPTMSFDIV	TPTM0FDIV	TPTM1FDIV
TPTMSICNT0	TPTM0ICNT0	TPTM1ICNT0
TPTMSILD0	TPTM0ILD0	TPTM1ILD0
TPTMSICNT1	TPTM0ICNT1	TPTM1ICNT1
TPTMSILD1	TPTM0ILD1	TPTM1ILD1
TPTMSICNT2	TPTM0ICNT2	TPTM1ICNT2
TPTMSILD2	TPTM0ILD2	TPTM1ILD2
TPTMSICNT3	TPTM0ICNT3	TPTM1ICNT3
TPTMSILD3	TPTM0ILD3	TPTM1ILD3
TPTMSFCNT	TPTM0FCNT	TPTM1FCNT

3.7.2.3 TPTMnIRUN — Counter Start Register of Interval Timer for PEn (n = 0 to 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	IRUN3	IRUN2	IRUN1	IRUN0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 3.158 TPTMnIRUN Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved. When read, the value after reset is returned. When writing, write the value after reset.
3	IRUN3	Start command bit of interval timer channel 3 0: No action 1: Start counter When writing 1 to the bit, the value of TPTMnILD3 is loaded in TPTMnCINT3. This bit is always read as 0.
2	IRUN2	Start command bit of interval timer channel 2 0: No action 1: Start counter When writing 1 to the bit, the value of TPTMnILD2 is loaded in TPTMnCINT2. This bit is always read as 0.
1	IRUN1	Start command bit of interval timer channel 1 0: No action 1: Start counter When writing 1 to the bit, the value of TPTMnILD1 is loaded in TPTMnCINT1. This bit is always read as 0.
0	IRUN0	Start command bit of interval timer channel 0 0: No action 1: Start counter When writing 1 to the bit, the value of TPTMnILD0 is loaded in TPTMnCINT0. This bit is always read as 0.

3.7.2.4 TPTMnIRRUN — Counter Restart Register of Interval Timer for PEn (n = 0 to 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	IRRUN3	IRRUN2	IRRUN1	IRRUN0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 3.159 TPTMnIRRUN Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved. When read, the value after reset is returned. When writing, write the value after reset.
3	IRRUN3	Restart command bit of interval timer channel 3 0: No action 1: Restart counter When writing 1 to the bit, the counter is restarted from current value of TPTMnICNT3. This bit is always read as 0.
2	IRRUN2	Restart command bit of interval timer channel 2 0: No action 1: Restart counter When writing 1 to the bit, the counter is restarted from current value of TPTMnICNT2. This bit is always read as 0.
1	IRRUN1	Restart command bit of interval timer channel 1 0: No action 1: Restart counter When writing 1 to the bit, the counter is restarted from current value of TPTMnICNT1. This bit is always read as 0.
0	IRRUN0	Restart command bit of interval timer channel 0 0: No action 1: Restart counter When writing 1 to the bit, the counter is restarted from current value of TPTMnICNT0. This bit is always read as 0.

3.7.2.5 TPTMnISTP — Counter Stop Register of Interval Timer for PEn (n = 0 to 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ISTP3	ISTP2	ISTP1	ISTP0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 3.160 TPTMnISTP Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved. When read, the value after reset is returned. When writing, write the value after reset.
3	ISTP3	Stop command bit of interval timer channel 3 0: No action 1: Stop counter When writing 1 to the bit, the counter is stopped at the value of TPTMnICNT3. This bit is always read as 0.
2	ISTP2	Stop command bit of interval timer channel 2 0: No action 1: Stop counter When writing 1 to the bit, the counter is stopped at the value of TPTMnICNT2. This bit is always read as 0.
1	ISTP1	Stop command bit of interval timer channel 1 0: No action 1: Stop counter When writing 1 to the bit, the counter is stopped at the value of TPTMnICNT1. This bit is always read as 0.
0	ISTP0	Stop command bit of interval timer channel 0 0: No action 1: Stop counter When writing 1 to the bit, the counter is stopped at the value of TPTMnICNT0. This bit is always read as 0.

3.7.2.6 TPTMnISTR — Counter Status Register of Interval Timer for PEn (n = 0 to 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ISTR3	ISTR2	ISTR1	ISTR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.161 TPTMnISTR Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved. When read, the value after reset is returned. When writing, write the value after reset.
3	ISTR3	Status bit of interval timer channel 3 0: Counter is stopped. 1: Counter is running.
2	ISTR2	Status bit of interval timer channel 2 0: Counter is stopped. 1: Counter is running.
1	ISTR1	Status bit of interval timer channel 1 0: Counter is stopped. 1: Counter is running.
0	ISTR0	Status bit of interval timer channel 0 0: Counter is stopped. 1: Counter is running.

3.7.2.7 TPTMnIIEN — Interrupt Enable Register of Interval Timer for PEn (n = 0 to 1)

This register controls enabling/disabling of interrupt request by TPTM_IRQ[n].

When underflow occurs in interval timer channel m in TPTMn, TPTMnIUSTR.IUSTRm bit is set whether IIENm bit is set or not. If IIENm bit is set, TPTM_IRQ[n] is asserted.

Note that TPTM_IRQ[n] will be asserted immediately after IIENm bit is set if TPTMnIUSTR.IUSTRm holds 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	IIEN3	IIEN2	IIEN1	IIEN0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 3.162 TPTMnIIEN Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved. When read, the value after reset is returned. When writing, write the value after reset.
3	IIEN3	Interrupt enable bit of interval timer channel 3 0: Disable. 1: Enable.
2	IIEN2	Interrupt enable bit of interval timer channel 2 0: Disable. 1: Enable.
1	IIEN1	Interrupt enable bit of interval timer channel 1 0: Disable. 1: Enable.
0	IIEN0	Interrupt enable bit of interval timer channel 0 0: Disable. 1: Enable.

3.7.2.8 TPTMnIUSTR — Underflow Status Register of Interval Timer for PEn (n = 0 to 1)

This register indicates the status of underflow occurrence of interval timers in TPTMn.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	IUSTR3	IUSTR2	IUSTR1	IUSTR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 3.163 TPTMnIUSTR Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved. When read, the value after reset is returned. When writing, write the value after reset.
3	IUSTR3	Underflow flag bit of interval timer channel 3 0: Underflow did not occur. 1: Underflow occurred. When writing 0 to the bit, IUSTR3 is cleared. Writing 1 to the bit is ignored.
2	IUSTR2	Underflow flag bit of interval timer channel 2 0: Underflow did not occur. 1: Underflow occurred. When writing 0 to the bit, IUSTR2 is cleared. Writing 1 to the bit is ignored.
1	IUSTR1	Underflow flag bit of interval timer channel 1 0: Underflow did not occur. 1: Underflow occurred. When writing 0 to the bit, IUSTR1 is cleared. Writing 1 to the bit is ignored.
0	IUSTR0	Underflow flag bit of interval timer channel 0 0: Underflow did not occur. 1: Underflow occurred. When writing 0 to the bit, IUSTR0 is cleared. Writing 1 to the bit is ignored.

3.7.2.9 TPTMnIDIV — Divider Register of Interval Timer for PEn (n = 0 to 1)

Value after reset: 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	IDIV[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.164 TPTMnIDIV Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved. When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	IDIV[7:0]	Clock dividing setting of interval timer for all channels Counter clock ratio is specified by the following expression. $count_clock = clk_cpu / (IDIV[7:0] + 1)$

CAUTIONS

- To avoid unintended timer behavior, the TPTMnIDIV register should be written while all channels of TPTMnICNTm are in stop state.
- It is prohibited to access this register by bit-manipulation instruction.

3.7.2.10 TPTMnFRUN — Counter Start Register of Free-run Timer for PEn (n = 0 to 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRUN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 3.165 TPTMnFRUN Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved. When read, the value after reset is returned. When writing, write the value after reset.
0	FRUN	Start command bit of free-run timer 0: No action 1: Start counter When writing 1 to the bit during the counter stopping, the value of 0000 0000 _H is loaded in TPTMnFCNT and the counter is started. When writing 1 to the bit during the counter working, the value of 0000 0000 _H is loaded in TPTMnFCNT and the counter is re-started. This bit is always read as 0.

3.7.2.11 TPTMnFRRUN — Counter Restart Register of Free-run Timer for PEn (n = 0 to 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRRUN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 3.166 TPTMnFRRUN Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved. When read, the value after reset is returned. When writing, write the value after reset.
0	FRRUN	Restart command bit of free-run timer 0: No action 1: Restart counter When writing 1 to the bit, the counter is restarted from current value of TPTMnFCNT. This bit is always read as 0.

3.7.2.12 TPTMnFSTP — Counter Stop Register of Free-run Timer for PEn (n = 0 to 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FSTP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 3.167 TPTMnFSTP Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved. When read, the value after reset is returned. When writing, write the value after reset.
0	FSTP	Stop command bit of free-run timer 0: No action 1: Stop counter When writing 1 to the bit, the counter is stopped at the value of TPTMnFCNT. This bit is always read as 0.

3.7.2.13 TPTMnFSTR — Counter Status Register of Free-run Timer for PEn (n = 0 to 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FSTR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.168 TPTMnFSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved. When read, the value after reset is returned. When writing, write the value after reset.
0	FSTR	Status bit of Free- run timer 0: Counter is stopped. 1: Counter is running.

3.7.2.14 TPTMnFDIV — Divider Register of Free-run Timer for PEn (n = 0 to 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	FDIV[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.169 TPTMnFDIV Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved. When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	FDIV[7:0]	Clock dividing setting of free-run timer Counter clock ratio is specified by the following expression. $count_clock = clk_cpu / (FDIV[7:0] + 1)$

CAUTIONS

- To avoid unintended timer behavior, the TPTMnFDIV register should be written while TPTMnFCNT is in stop state.
- It is prohibited to access this register by bit-manipulation instruction.

3.7.2.15 TPTMnICNTm — Counter Channel m Register of Interval Timer for PEn (n = 0 to 1, m = 0 to 3)

Value after reset: Undefined

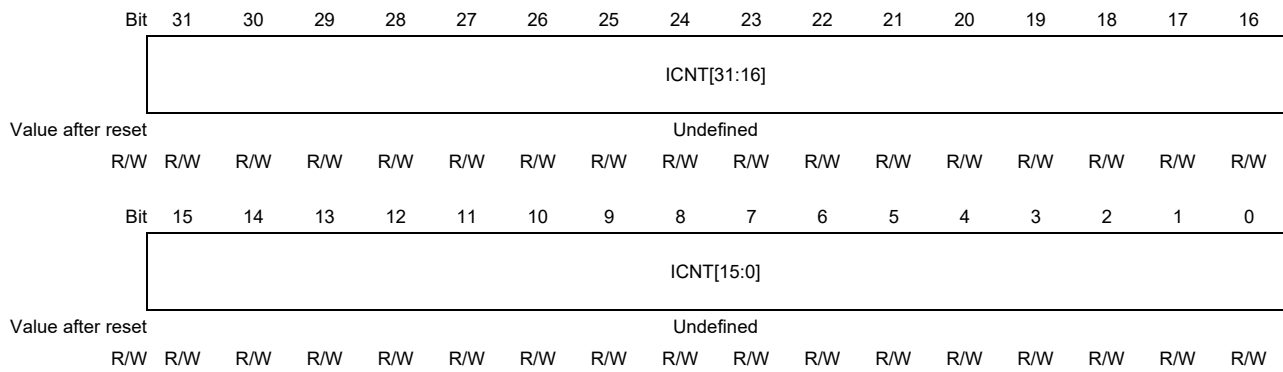


Table 3.170 TPTMnICNTm Register Contents

Bit Position	Bit Name	Function
31 to 0	ICNT[31:0]	Counter value of interval timer channel m for PEn When writing 1 _B to TPTMnIRUN.IRUNm, the value of TPTMnILDm is loaded in this register and this register starts to count down from this value. Also, when an underflow of this register occurs, the value of TPTMnILDm is reloaded in this register and this register continues to count down from this value.

CAUTION

To avoid unintended timer behavior, the TPTMnICNTm register should be written while in stop state.

3.7.2.16 TPTMnILDm — Load Channel m Register of Interval Timer for PEn (n = 0 to 1, m = 0 to 3)

Value after reset: Undefined

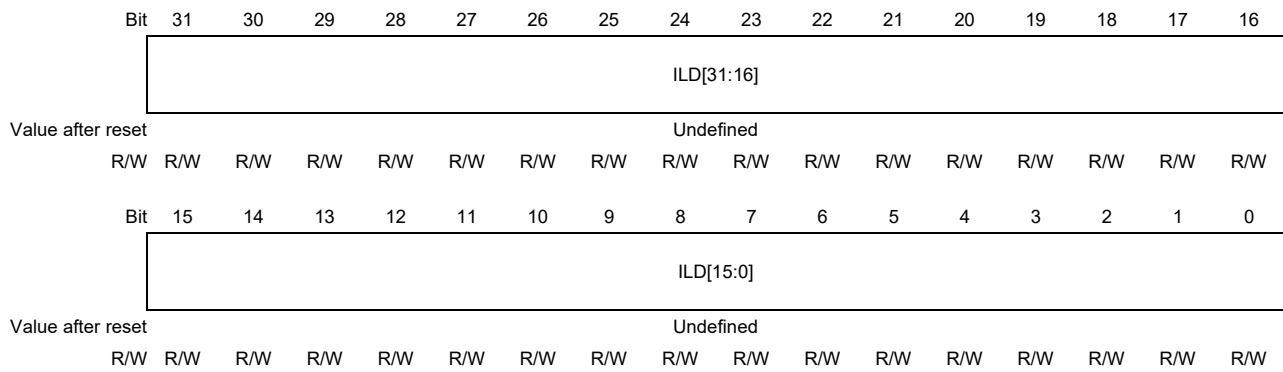


Table 3.171 TPTMnILDm Register Contents

Bit Position	Bit Name	Function
31 to 0	ILD[31:0]	Load data value of interval timer channel m for PEn At the timing of writing 1 to TPTMnIRUNm, the value of this register is loaded in TPTMnICNTm. Also, when an underflow of TPTMnICNTm occurs, the value of this register is reloaded in TPTMnICNTm.

CAUTION

To avoid unintended timer behavior, the TPTMnILDm register should be written while TPTMnCNTm of corresponding channel is in stop state.

3.7.2.17 TPTMnFCNT — Counter Register of Free-run Timer for PEn (n = 0 to 1)

Value after reset: Undefined

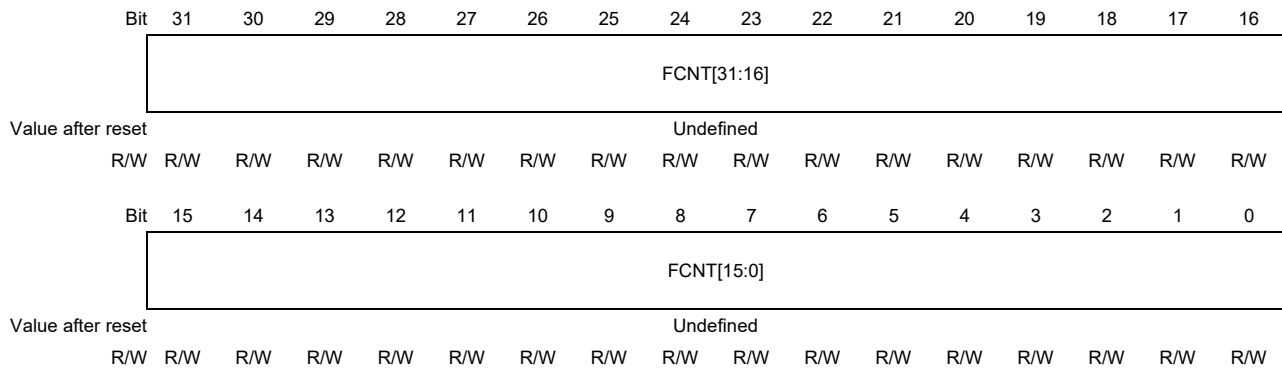


Table 3.172 TPTMnFCNT Register Contents

Bit Position	Bit Name	Function
31 to 0	FCNT[31:0]	Counter value of free-run timer for PEn When writing 1 _B to TPTMnFRUN.FRUN, the value of 0000_0000 _H is loaded in this register and this register starts to count up from this value. Also, when an overflow of this register occurs, the value of 0000_0000 _H is reloaded in this register and this register continues to count up from this value.

CAUTION

To avoid unintended timer behavior, the TPTMnFCNT register should be written while in stop state.

3.7.3 TPTM Function

CAUTION

If interruption is used as EI level instead of FE level, set the TPTMSEL register before TPTM starts.
It is prohibited to change the setting value of the TPTMSEL register after TPTM start.

3.7.3.1 Normal Operation

The normal operation flow is shown in **Figure 3.53**.

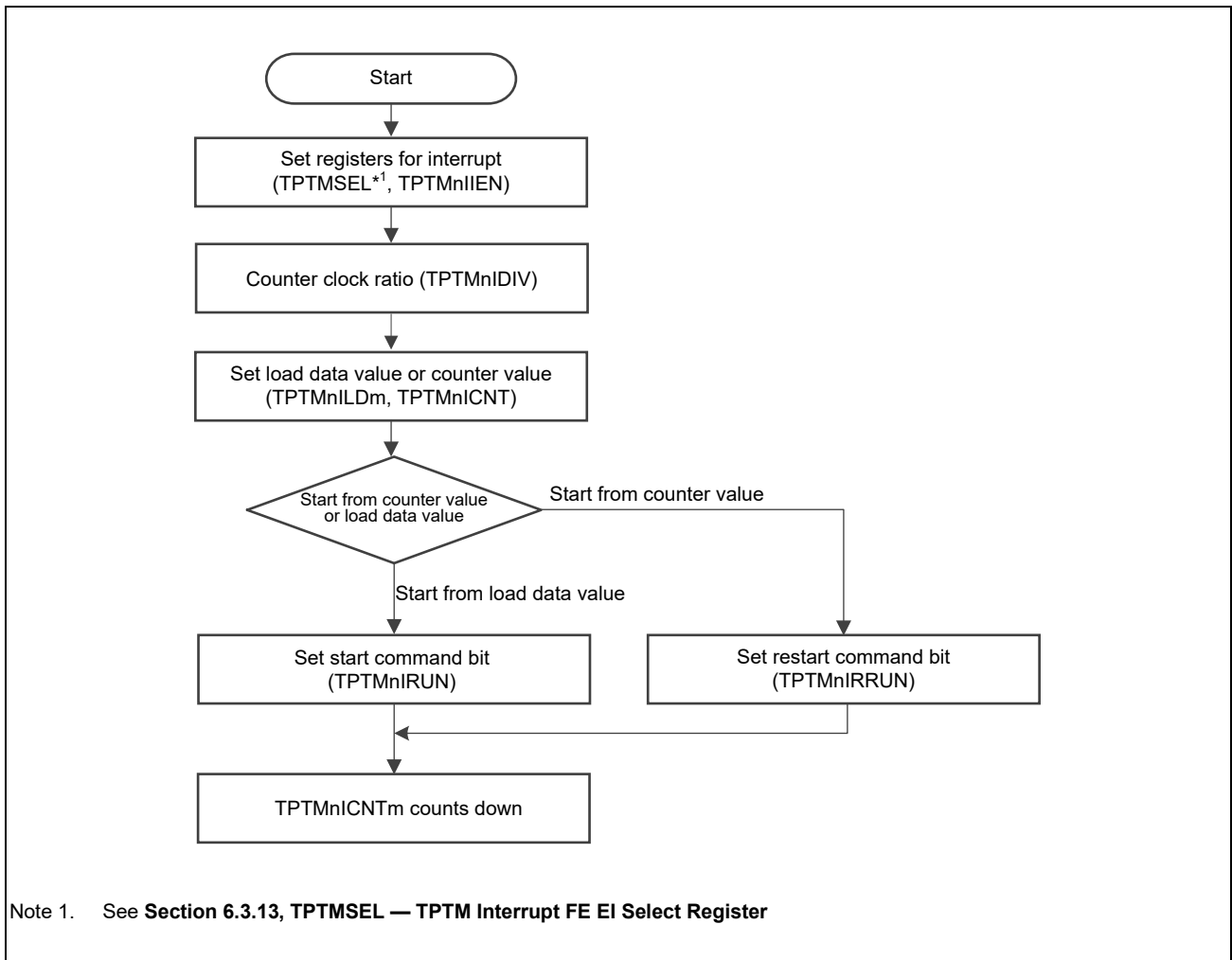


Figure 3.53 Flow of Normal Operation

It shows the timing chart of normal operation of interval timer in case that counter clock ratio is a 1/2 period of CPU_CLK and load data value is 0000 0004_H in **Figure 3.54**.

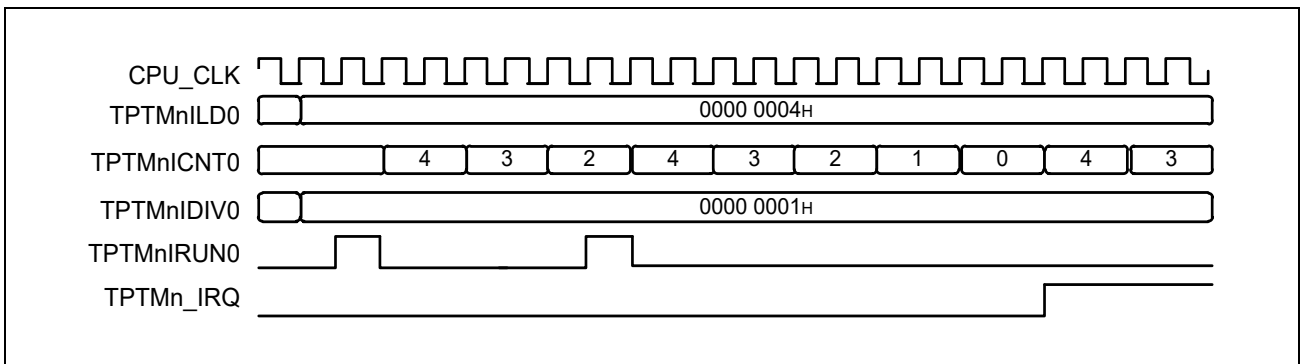


Figure 3.54 Normal Operation (TPTMnIDIV = 0000 0001_H, TPTMnILD = 0000 0004_H)

3.7.3.2 Operation during a Restart

The flow of operation during a restart is shown in **Figure 3.55**.

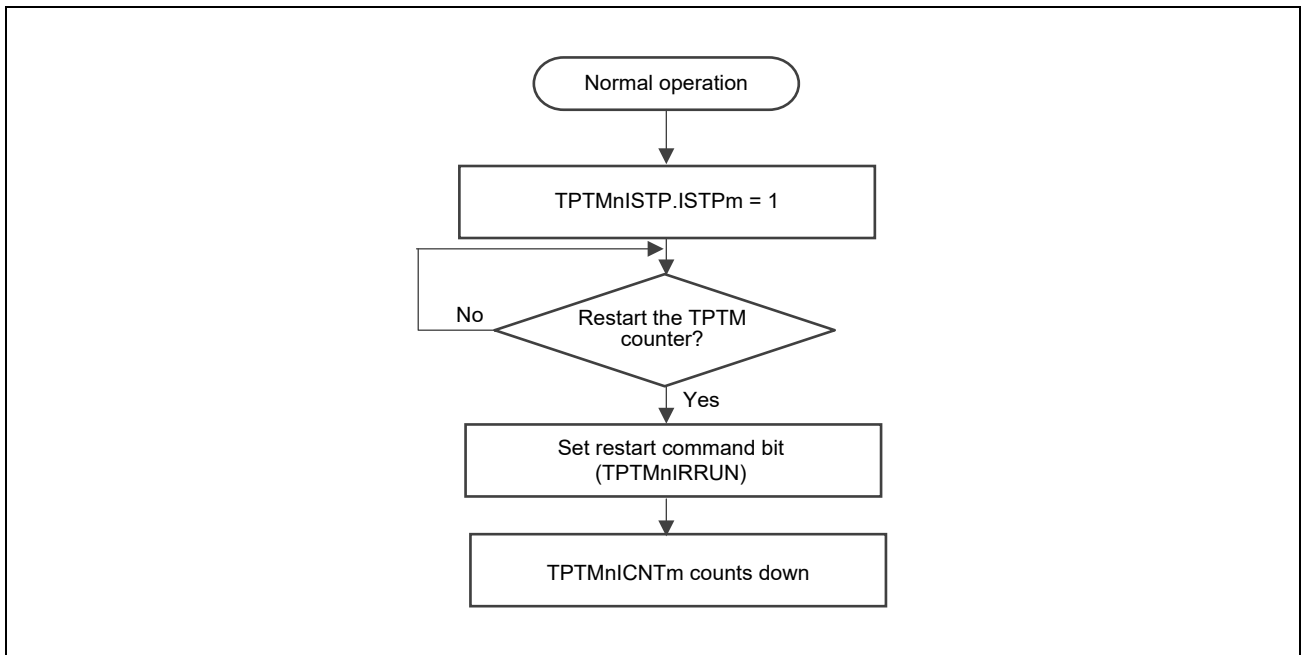


Figure 3.55 Flow of Operation during a Restart

TPTMnCNTm starts countdown at the next cycle after setting TPTMnIRUN.

TPTMnCNTm stops countdown at the next cycle after setting TPTMnISTP during countdown.

TPTMnCNTm restarts countdown from the current counter value at the next cycle after setting TPTMnIRRUN during countdown stop.

The timing chart of interval timer operation at the time of restart when the counter clock ratio is 1/2 period of CPU_CLK and load data value is 0000 0004_H is shown in **Figure 3.54**.

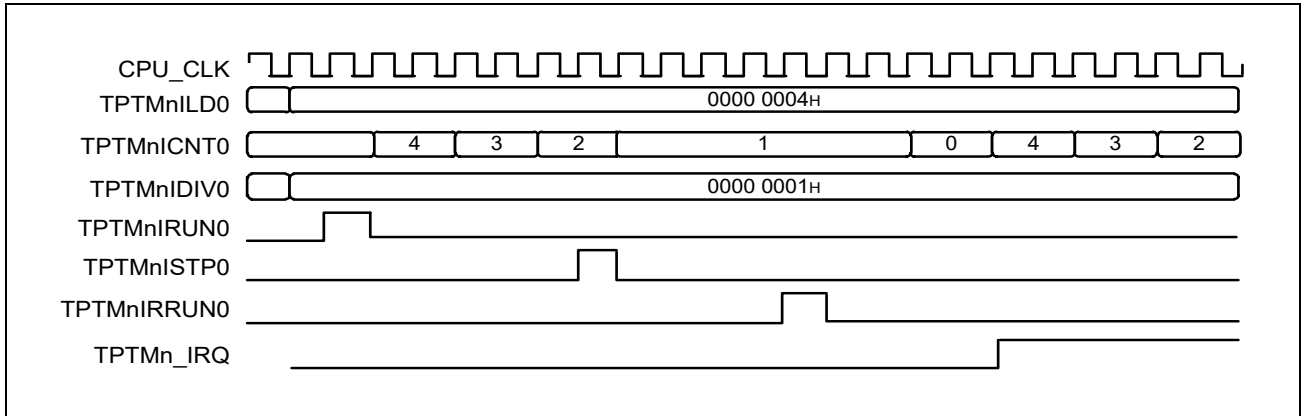


Figure 3.56 Operation during a Restart (TPTMnDIV = 0000 0001H, TPTMnILD = 0000 0004H)

3.7.3.3 Operation during an Underflow

When TPTMnIEN.IENm is 1B, an interrupt is enabled, TPTMnIUSTR.IUSTRm is set and TPTMn_IRQ is asserted at the next cycle after TPTMnCNTm becomes 0000 0000H.

When CPU has finished interrupt processing of TPTMn, it needs to clear TPTMnIUSTR.IUSTRm.

Cycles of TPTMn Interrupt

The cycles of TPTMn Interrupt are as follows.

- TPTMn Interrupt generation cycle = TPTMnCNTm × Counter-Clock cycle*

$$*Counter-Clock\ cycle = CPU_CLK \times (TPTMnDIV + 1)$$

When TPTMnIEN.IENm is 0, an interrupt is disabled, TPTMnIUSTR.IUSTRm is set but TPTMn_IRQ is not asserted at the next cycle after TPTMnCNTm becomes 0000 0000H. In this case, TPTMn_IRQ is asserted if TPTMnIEN.IENm is set before TPTMnIUSTR.IUSTRm is cleared.

3.8 Usage Notes

3.8.1 Synchronization of Store Instruction Completion and Subsequent Instruction Generation

When a control register is updated by a store instruction, there is a time lag from execution of the store instruction by the CPU to actual updating of the control register. Therefore, appropriate processing for synchronization is required to ensure that control registers reflect updated values before execution of subsequent instructions. The processing for synchronization is shown below.

For details on the procedure regarding updating of the system registers by the LDSR instruction and synchronization with subsequent instructions, refer to **Section 3.2.7.3, Hazard Management after System Register Update**.

3.8.1.1 When Updated Results in the Control Registers or Memory are Reflected in the Implementation of a Subsequent Instruction

Example 1) An interrupt may be enabled by implementation of an EI instruction after an interrupt request is cleared by access from the control register in the INTC2 and the peripheral circuits. Proceed as follows in this case.

- (1) Store instruction to update a control register (ST.W, etc.)
- (2) Dummy reading of the above-mentioned control register (LD.W, etc.)*¹
- (3) SYNCP
- (4) Subsequent instruction (EI, etc.)

Example 2) Implement the same processing even when the access required after waiting to secure the updating of a given control register (register A) is to another control register (register B). This includes the following cases: the interlinked operation of different peripheral modules and when releasing the interrupt mask in INTC after making peripheral module settings.

However, this processing is unnecessary if control registers A and B are in the same peripheral group.

For the correspondence between the peripheral groups and the peripheral modules/registers, see each section.

- (1) Store instruction to update control register A (ST.W, etc.)
- (2) Dummy reading of the above-mentioned control register (LD.W, etc.)*¹
- (3) SYNCP
- (4) Store/Load instruction to access control register B (ST.W, LD.W, etc.)

The same processing is also required when access to control registers and memory within the scope of protection starts after waiting for the completion of settings for safety functions such as memory protection, ECC checking, and so on.

Moreover, the same processing is also required when guarantee the completion of store accesses to memory.

Note 1. Dummy reading of any register of the same peripheral group can be used instead.

3.8.1.2 When the Updated Results in the Control Registers and Memories are Reflected in the Instruction Fetch of a Subsequent Instruction

1. If you wish to write an instruction to the RAM and then branch to the RAM to execute the written instruction, do this as follows.
 - (1) Store instruction to update a memory (ST.W, etc.)
 - (2) Dummy reading of the above-mentioned memory (LD.W, etc.)
 - (3) SYNCI
 - (4) Subsequent instruction (branch instruction, etc.)

2. When branching to a target memory after waiting for the completion of updating the control registers for memory protection and ECC, do this as follows.
 - (1) Store instruction to update a control register (ST.W, etc.)
 - (2) Dummy reading of the control register (LD.W, etc.)
 - (3) SYNCI
 - (4) Subsequent instruction (branch instruction, etc.)

3.8.1.3 Product information of SYNCM

Table 3.173 shows the correspondence between the module names given in *the RH850G4MH User's Manual: Software* and the module names in this product information document.

Table 3.173 Module Name Correspondence in this Product Information Document

Module Name	Module Name of This Product Information Document
L1RAM (given CPU)	Local RAM (own core)
L1RAM (other CPU)	Local RAM (other core)
L2RAM	Cluster0/1 RAM
INTC1	INTC1
Others	There is no module corresponding to "Others" in this product

3.8.1.4 Product information of Waiting for the Completion of Storage

Table 3.174 Bus Slaves for which Using Dummy Reading to Wait is Effective

Bus Slaves	Description
L1RAM (given CPU)	Local RAM (own core)
L1RAM (other CPU)	Local RAM (other core)
L2RAM	Cluster0/1 RAM
INTC1	INTC1
Others	All peripheral module

3.8.2 Synchronization of Load Instruction Completion and Subsequent Instruction Generation

When a control register and memory is accessed by a load instruction, the order of data accesses does not guarantee the same order of program. Therefore, appropriate processing for synchronization is required to ensure the completion of a load instruction for the subsequent data access instructions. However, the synchronization processing is not necessary for accesses to some target data space. For details on target data space, refer to **Section 4.2.2, Data Space Accessible by CPU0, CPU1**. The processing for load synchronization is shown below.

Example) It is necessary to wait for the completion of a load access to an address A before an access to other address B, proceed as follows in this case.

- (1) Load instruction to refer Address A (LD.W, etc.)
- (2) SYNC
- (3) Store/Load instruction to access Address B (ST.W, LD.W, etc.)

3.8.3 Accesses to Registers by Bit-Manipulation Instructions

Processing of a bit-manipulation instruction takes the form of atomic reading, modification, and writing of an eight-bit unit. Thus, access by a bit-manipulation instruction is only possible for registers for which reading and writing in 8-bit units is possible. However, take care in the cases of registers that contain multiple flag bits, since the read-modify-write cycle may also clear flags other than those which were for clearing.

Write access to H-Bus group registers by using bit-manipulation instructions is not atomic. Access by other masters may interrupt the read-modify-write processing of these instructions.

3.8.4 Ensuring Coherency after Code Flash Programming

For PE0 and PE1, see **Section 3.2.9, Ensuring Coherency after Code Flash Programming**.

For ICUMD, see *the E2x ICUMD User's Manual*.

3.8.5 Overwriting Context when Acknowledging Multiple Exceptions

Exceptions may be acknowledged regardless of the states of the ID or NP bits of the PSW register. This depends on the type of exception. When multiple exceptions occur, the contents of the system registers which hold the context information are overwritten. Regarding the conditions for acknowledging exceptions from each source and the possibility of return and recovery, see **Section 3.2.4, Exceptions and Interrupts**.

3.8.6 Usage Notes on Prefetching

CPU executes speculative instruction fetching from locations after the current value of the program counter to maintain the throughput of instruction fetches. Reading from memory due to such prefetching may proceed even from locations to which instruction codes have not been assigned (Note 1. in **Figure 3.57**). Note the following and keep in mind that the CPU does not execute values read in such cases.

The following notes apply to instruction fetching from memory in general.

- Occurrence of ECC errors due to values in memory being undefined

This prefetching may lead to an ECC error in case of reading from the code flash memory after it has been erased or from the Local RAM or Cluster RAM before initialization. When instruction codes are assigned to memory, initialize said area with values as desired (Note 1. in **Figure 3.57**).

- Detection of illegal access by the CRG

The CRG may detect such prefetching as illegal access. To prevent prefetching being detected as an illegal access, do not allow any region of overlap area with said areas (Note 1. in **Figure 3.57**) and areas to which access is prohibited by the CRG. Reading from an area protected by the MPU does not cause a memory protection exception.

- Access to Access Prohibited Area

Assign instruction codes to memory without allowing any overlap between said area (Note 1. in **Figure 3.57**) and an access-prohibited area.

- Speculative read operation by Prefetch

Depending on the decode result of prefetch instruction, the CPU may issue speculative read request to Local RAM (own core), Local RAM (other core), Cluster RAM or Code Flash. The lockstep error described in **Section 38.4.4, Usage Note** may be occurred by speculative read request. If it is unnecessary as a result of branch instruction, the data is discarded.

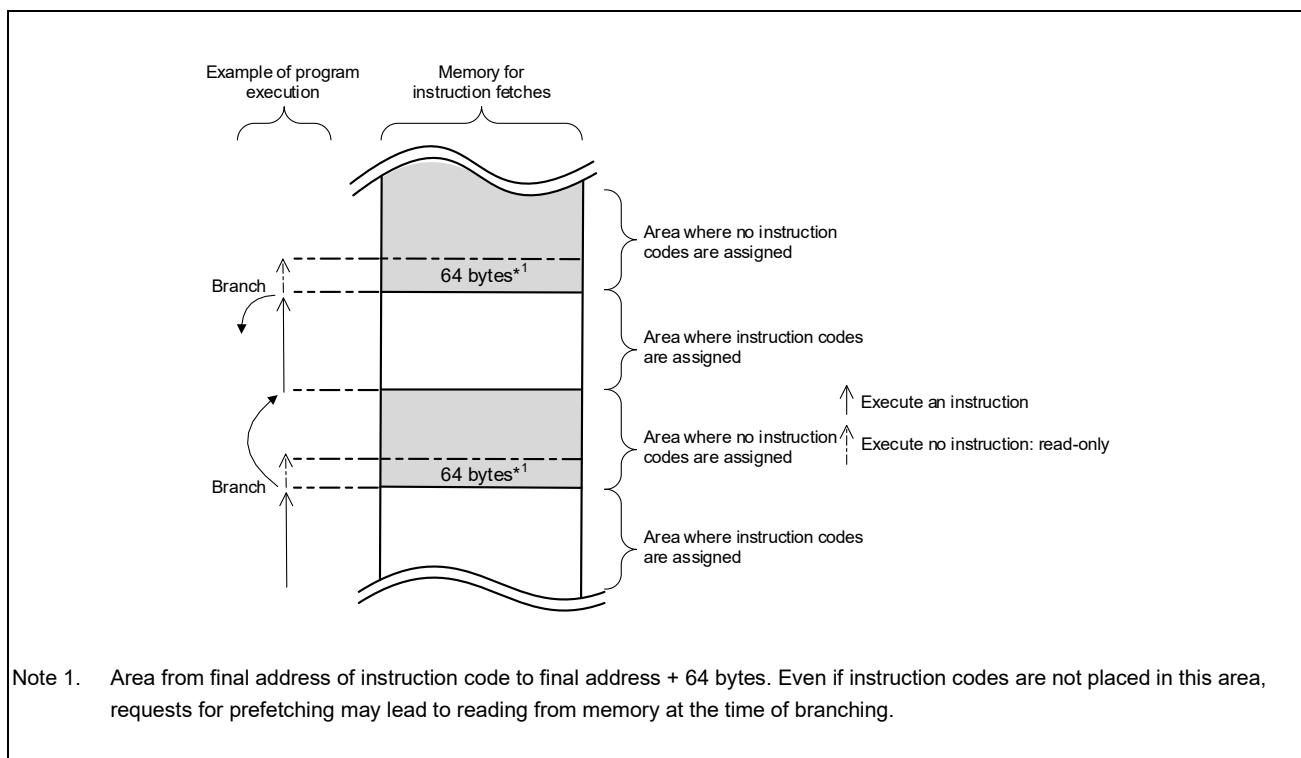


Figure 3.57 Areas that Require Attention Regarding Prefetching

3.8.7 Product Information of Initial Value for G4MH Register

The G4MH register's initial values of the product dependent are shown below.

Register No.	Symbol	Initial value
SR0,1	SPID	*1
SR1,1	SPIDLIST	FFFF FFFF _H
SR2,1	RBASE	*2
SR6,1	PID	*3
SR0,2	PEID	*4
SR1,2	BMID	*5
SR24,4	ICCTRL	0001 0003 _H
SR26,4	ICCFG	*6

Note 1. The initial values of SPID register are listed below.

CPU0: 0000 0000_H

CPU1: 0000 0001_H

Note 2. See **Section 4, Address MAP** about RBASE register's bit to 31 from 9. Initial value of RBASE.DV bit is 0 and Initial value of RBASE.RINT bit is 0.

Note 3. The initial values of PID register are listed below.

E2x-FCC1 0604 0F00_H(CPU0/CPU1)

E2M 0604 0F00_H(CPU0/CPU1)

Note 4. The initial values of PEID register are listed below.

CPU0: 0000 0000_H

CPU1: 0000 0001_H

Note 5. The initial values of BMID register are listed below.

CPU0: 0000 0000_H

CPU1: 0000 0001_H

Note 6. The initial values of ICCFG register are listed below.

E2x-FCC1 0000 1044_H

E2M 0000 1044_H

3.8.8 Product Information of SYSERR Factor

The product-dependent SYSERR causes are shown below. Refer to **Section 3.2.4, Exceptions and Interrupts** for details on SYSERR.

Exception Cause Code* ¹	Cause
11 _H	When any of the following errors occurs in instruction fetch <ul style="list-style-type: none"> • ADDRESS ECC error in Local RAM, Cluster RAM and Code Flash • Guard error in PEG, CRG and security guard.*² • Address Feedback error in LRAM and Cluster RAM • An unimplemented area is fetch accessed
13 _H	When any of the following errors occurs in instruction fetch <ul style="list-style-type: none"> • Uncorrectable DATA ECC error in Local RAM, Cluster RAM and Code Flash (including cases where 1-bit correction is prohibited) • Address parity error in Code Flash.

Note 1. Refer to **Section 3.2.4, Exceptions and Interrupts** about another exception cause code (excluding 11_H and 13_H).

Note 2. Guard error caused by security setting. Refer to *the E2x ICUMD User's Manual* for details on Security guard.

3.8.9 Product Information of Cache Structure

The product-dependent cache structure are shown below.

Cache structure	E2x-FCC1	E2M
Total Size	16 KB	16 KB
Number of Way	4-way	4-way
Number of Line	128 line	128 line

3.8.10 Product Information of Fetch Size

These product's CPU(G4MH) fetch size is 64-Bit.

3.8.11 Register Initialization

The CPU in this product has general purpose registers and system registers which value after reset are undefined. In case of lockstep core, initial value of the master core and checker core after reset may be different. Therefore, these registers must be initialized at first. The registers which need the initialization are shown in **Table 3.175**.

For Usage of Lockstep Error, see **Section 38.4.4, Usage Note**.

Table 3.175 Registers needed to be initialized

Type	Registers
General purpose register	r1 to r31
Basic system register	EIPC, FEPC, CTPC, EIWR, FEWR, EBASE, INTBP, MEA, MEI, RBIP
FPU system register * ¹	FPSR, FPEPC, FPST, FPCC
MPU function register	MCA, MCS, MCR, MCI, MPLA, MPUA, MPAT, MPID0 to MPID7
FXU Function Registers * ²	wr0 to wr31, FXSR, FXST, FXXC, FXXP
Cache operation function register	ICTAGL, ICTAGH, ICTATL, ICTATH, ICERR

Note 1. When the PSW.CU0 is set to 1, initialize the above registers.

Note 2. When the PSW.CU1 is set to 1, initialize the above registers.

Section 4 Address MAP

4.1 Address MAP

Table 4.1 shows the address space of the RH850/E2x-FCC1 and E2M.

When making an access to register space, access the addresses shown in each peripheral IPs section. Do not access an address that is not specified in **Table 4.1** and an area where any access is prohibited. If an unspecified address or reserved area is accessed, operation is not guaranteed.

Table 4.1 Address MAP of RH850/E2x-FCC1 and E2M

Address	Address Space Type	Size
0000 0000 _H to 007F FFFF _H	Code flash (User area)	8 MB
0080 0000 _H to 07FF FFFF _H	Access prohibited area	
0800 0000 _H to 0800 FFFF _H	Code flash (User boot area)	64 KB
0801 0000 _H to 0802 FFFF _H	Access prohibited area	
0803 0000 _H to 0803 3FFF _H	Code flash (Product Info Area)	16 KB
0803 4000 _H to 0803 7FFF _H	Code flash (ECC test Area)	16 KB
0803 8000 _H to 0FFF FFFF _H	Access prohibited area	
1000 0000 _H to 1FFF FFFF _H	H-Bus area	256 MB
2000 0000 _H to FD9F FFFF _H	Access prohibited area	
FDA0 0000 _H to FDA0 7FFF _H	Local RAM (CPU1)	32 KB
FDA0 8000 _H to FDBF FFFF _H	Access prohibited area	
FDC0 0000 _H to FDC0 7FFF _H	Local RAM (CPU0)	32 KB
FDC0 8000 _H to FDDF FFFF _H	Access prohibited area	
FDE0 0000 _H to FDE0 7FFF _H	Local RAM (self)	32 KB
FDE0 8000 _H to FDFE FFFF _H	Access prohibited area	
FE00 0000 _H to FE0A FFFF _H	Cluster RAM	704 KB
FE0B 0000 _H to FEFF FFFF _H	Access prohibited area	
FF00 0000 _H to FFFB 7FFF _H (FF20 0000 _H to FF23 7FFF _H) (FF30 0000 _H to FF3F FFFF _H)	P-Bus area (Data Flash) (Flash Extra Area)	15.72 MB 224 KB 1 MB
FFFB 8000 _H to FFFB EFFF _H	I-Bus area	28 KB
FFFB F000 _H to FFFB FFFF _H	Access prohibited area	
FFFC 0000 _H to FFFC 3FFF _H	CPU peripheral area (self)	16 KB
FFFC 4000 _H to FFFC 7FFF _H	CPU peripheral area (CPU0)	16 KB
FFFC 8000 _H to FFFC BFFF _H	CPU peripheral area (CPU1)	16 KB
FFFC C000 _H to FFFF FFFF _H	Access prohibited area	

See **Section 3, CPU System** for details about each area.

See **Section 41.2.3, Mapping of Flash Extra Area** for details about Flash Extra Area.

See **Section 41.2.4, Examples of the Chip Writing** for details about OTP setting.

See *RH850/E2x Safety Application Note* for details about ECC test Area in Code flash.

4.2 Address Space Viewed from Each Bus Master

Figure 4.1 shows address spaces of RH850/E2x-FCC1 and E2M viewed from each bus master.

4.2.1 Space in which Instructions can be Fetched

1. Instructions of CPU0, CPU1 and ICUM can be fetched from the code flash, local RAM and Cluster RAM.
2. The reset vector (RBASE initial value) of CPU0 and CPU1:
 - When starting up from the user boot area, its head address is 0800 0000_H.
 - When starting up from the user area of RH850/E2x-FCC1 and E2M, its head address is selectable from 0000 0000 to 007F FFFF_H by Option byte. For details, see **Section 41, Flash Memory**.
3. The reset vector (RBASE initial value) of ICUM:
 - See *the E2x ICUMD User's Manual*.

4.2.2 Data Space Accessible by CPU0, CPU1

All spaces are accessible.

NOTE

The data accesses inside the following target data space are operated in the same order as the program.

- Code Flash
- H-Bus area Group 0, 1, 2
- P-Bus area Group 0, 1, 2H/2L, 3, 4, 5, 6, 7, 8, 9
- I-Bus area 0, 1, 2, 3
- CPU peripheral area (self, CPU0, CPU1)
- Cluster RAM Cluster 0, 1*1

Note 1. The data accesses between Cluster0 RAM and Cluster1 RAM are also operated in the same order as the program.

The load accesses inside the following target data space may be operated before the completion of the preceding access (load or store). In order to guarantee the order of accesses, synchronization processing by software for the preceding access is necessary.

- Local RAM (self, CPU0, CPU1)

For the data accesses between different target spaces, the order of data accesses is not guaranteed as the program order.

In order to guarantee the order of accesses, synchronization processing by software for the preceding access is necessary.

For details of synchronization processing, refer to **Section 3.8.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation** and **Section 3.8.2, Synchronization of Load Instruction Completion and Subsequent Instruction Generation**.

4.2.3 Data Space Accessible by ICUM

See the *E2x ICUMD User's Manual*.

4.2.4 Data Space Accessible by DMA (sDMAC, DTS)

See **Figure 4.1** for the accessible spaces from the DMA.

4.2.5 Data Space Accessible by H-Bus Master

See **Figure 4.1** for the accessible spaces from the H-Bus Master.

NOTE

H-Bus Master can't access to each H-Bus Slave (except RHSIF). See table below about detail of HBUS access.

H-Bus Master \ H-Bus Slave	RHSIF	FLXA	ETNC
RHSIF	✓	✓	✓
FLXA	—	—	—
HS-SPI	—	—	—
ETNC	—	—	—

	Access from CPU0/1	Access from DMA	Access from H-Bus Master
0000 0000 _H	Code flash (user area)	Code flash (user area)	Code flash (user area)
007F FFFF _H			
0080 0000 _H	Access prohibited	Access prohibited	Access prohibited
07FF FFFF _H			
0800 0000 _H	Code flash (User boot area)	Code flash (User boot area)	Code flash (User boot area)
0800 FFFF _H			
0801 0000 _H	Access prohibited	Access prohibited	Access prohibited
0802 FFFF _H			
0803 0000 _H	Code flash (Product Info Area)	Code flash (Product Info Area)	Code flash (Product Info Area)
0803 3FFF _H			
0803 4000 _H	Code flash (ECC test area)	Code flash (ECC test area)	Code flash (ECC test area)
0803 7FFF _H			
0803 8000 _H	Access prohibited	Access prohibited	Access prohibited
0FFF FFFF _H			
1000 0000 _H	H-Bus area	H-Bus area	H-Bus area
1FFF FFFF _H			
2000 0000 _H	Access prohibited	Access prohibited	Access prohibited
FD9F FFFF _H			
FDA0 0000 _H	Local RAM(CPU1)	Local RAM(CPU1)	Local RAM(CPU1)
FDA0 7FFF _H			
FDA0 8000 _H	Access prohibited	Access prohibited	Access prohibited
FDBF FFFF _H			
FDC0 0000 _H	Local RAM(CPU0)	Local RAM(CPU0)	Local RAM(CPU0)
FDC0 7FFF _H			
FDC0 8000 _H	Access prohibited	Access prohibited	Access prohibited
FDDF FFFF _H			
FDE0 0000 _H	Local RAM(self)		
FDE0 7FFF _H			
FDE0 8000 _H	Access prohibited		
FDFE FFFF _H			
FE00 0000 _H	Cluster RAM	Cluster RAM	Cluster RAM
FE0A FFFF _H			
FE0B 0000 _H	Access prohibited	Access prohibited	Access prohibited
FEFF FFFF _H			
FF00 0000 _H	P-Bus area (Data flash) (Flash Extra Area)	P-Bus area (Data flash) (Flash Extra Area)	P-Bus area (Data flash) (Flash Extra Area)
FFFB 7FFF _H			
FFFB 8000 _H	I-Bus area	I-Bus area	I-Bus area
FFFB EFFF _H			
FFFB F000 _H	Access prohibited	Access prohibited	Access prohibited
FFFB FFFF _H			
FFFC 0000 _H	Cpu peripheral area (self)		
FFFC 3FFF _H			
FFFC 4000 _H	Cpu peripheral area (CPU0)	Cpu peripheral area (CPU0)	Cpu peripheral area (CPU0)
FFFC 7FFF _H			
FFFC 8000 _H	Cpu peripheral area (CPU1)	Cpu peripheral area (CPU1)	Cpu peripheral area (CPU1)
FFFC BFFF _H			
FFFC C000 _H	Access prohibited	Access prohibited	Access prohibited
FFFF FFFF _H			

Note: The following color coding is used in the map above.

Fetch and data access available
Data access available.
Access prohibited

Note: For usage assignment of instruction codes, see **Section 3.8.6, Usage Notes on Prefetching.**

Figure 4.1 Address Space of RH850/E2x-FCC1 and E2M Viewed from Each Bus Master

4.3 Cluster RAM and Retention RAM

The cluster RAM of RH850/E2x-FCC1 and E2M is divided into two, Cluster0 and Cluster1. Different clusters can be accessed concurrently. The upper 64 Kbytes of Cluster1 is the retention RAM that retains data even during poweroff standby state.

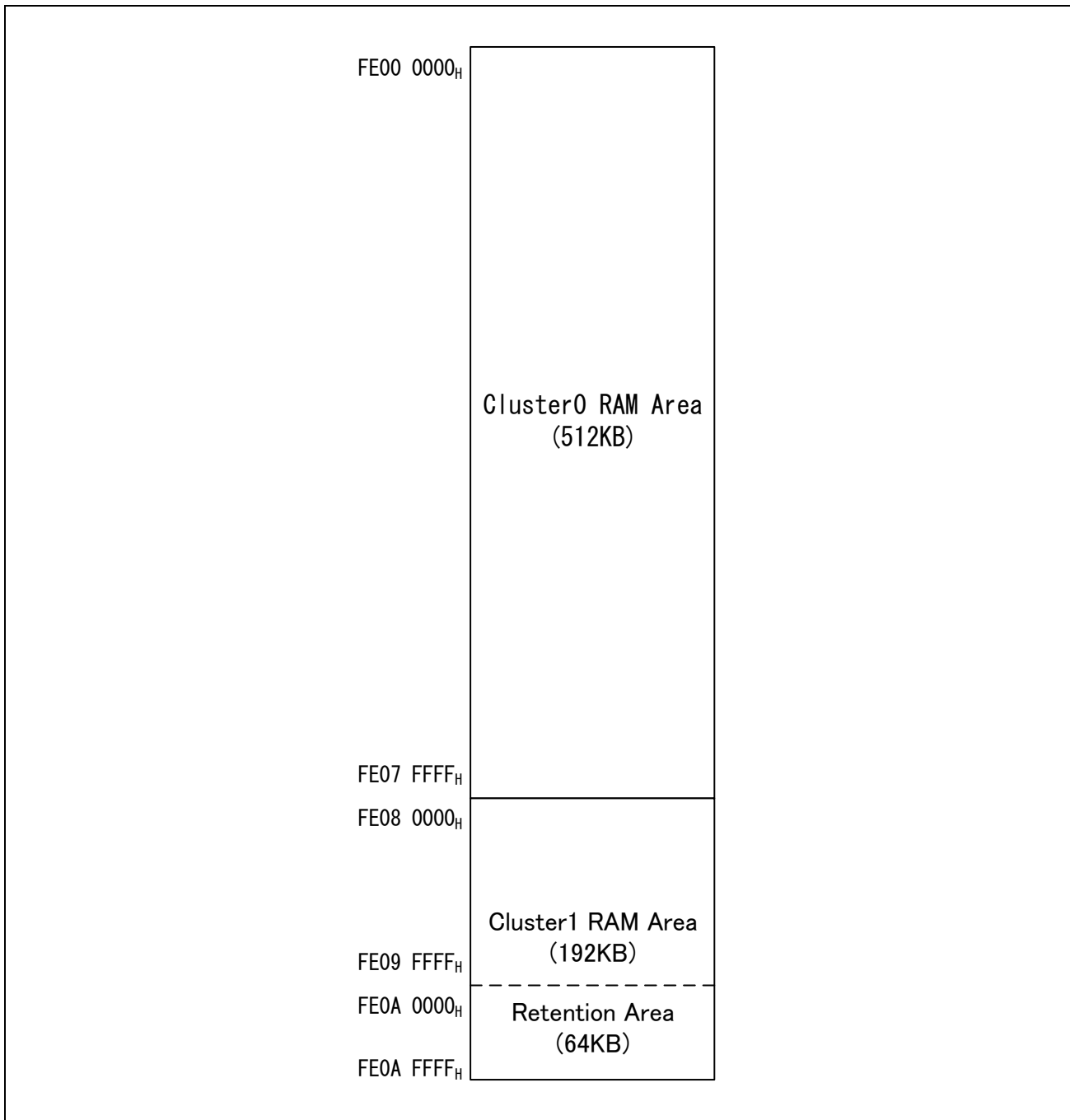


Figure 4.2 Cluster RAM of RH850/E2x-FCC1 and E2M

Section 5 Operating Modes

5.1 Features

This LSI has multiple operating modes, which can be selected with the three pins (MD1, MD0 and TRST) and the settings of option bytes STMSEL1 / STMSEL0. For the procedures to set STMSEL1 / STMSEL0, see **Section 41, Flash Memory**. **Table 5.1** lists the operating modes. The values of MD0, MD1 and TRST are latched at the rising edge of RES_IN to determine the operating mode. These pins are continually monitored during operation to detect mode errors.

Table 5.1 Mode List

Value Set to Pin			Value Set in Option Byte		Mode	Startup Area	Types of Debug Interface*1,*4	Flash Writer Interface	Remarks
MD1	MD0	TRST	STMS EL1	STMS EL0					
0	0	0	0	0	Normal operation mode	User area*2	Nexus/LPD	—	BIST is available
0	0	0	0	1	User boot mode	User boot area	Nexus/LPD	—	BIST is available
0	0	0	1	X	Serial programming mode	Boot firmware	Nexus/LPD*6	CSI	Serial programming is available
0	0	1*5,*8	X	0	Normal operation mode	User area*2	Nexus/LPD/ Boundary scan*7	—	CPU will not run*5 BIST is not available*8
0	0	1*5,*8	X	1	User boot mode	User boot area	Nexus/LPD/ Boundary scan*7	—	CPU will not run*5 BIST is not available*8
0	1	X	X	X	Serial programming mode	Boot firmware	Nexus/LPD*6	RS-CANFD / RS-CAN, CSI, 2-wire UART*3	Serial programming is available

Note 1. For the correspondence between the pin function and pin state in each debug interface. See **Section 2, Pin Function**.

Note 2. Reset vector address can be set by Option Bytes for the ROM area. See **Section 41, Flash Memory**.

Note 3. Method for changing the serial communication mode is shown in **Section 41, Flash Memory**. CSI is the "Clockless Serial Interface".

Note 4. Nexus/LPD for debug is determined by the TDI level when TRST is released. Nexus shall be selected when Boundary scan runs.

Note 5. CPU will not run after reset release in this mode when TRST = 1. This mode is for the purpose of On-Chip debug or when using a boundary scan. Option bytes can be used to disable the CPU from running after reset. See **Section 42, Basic Hardware Protection (BHP)** and **Section 41, Flash Memory** for details.

Note 6. When RS-CAN or RS-CANFD boot is selected.

Note 7. Boundary scan can be run when the CPU and ICUM are not running. Boundary scan is selected by using JTAG instructions. TDI shall be set 1 when TRST is released for running Boundary scan.

Note 8. BIST will not run when TRST = 1.

5.1.1 Normal Operation Mode

This mode is the default mode for execution of application software. The Boundary scan, Power-Up BIST, Shutdown BIST and the On-chip Debug capabilities also use this mode. After reset release, instruction fetch is carried out from the user area. For the reset vector of each CPU, see **Section 41, Flash Memory**.

The CPU will not autonomously run when TRST = 1. A debugger must be used to cause the CPU to run.

5.1.2 User Boot Mode

This mode is basically the same as Normal Operation Mode except for the fetch area.

This mode is used to provide an alternative boot location for application software.

After release from the reset state, instruction fetch is carried out from the user boot area.

5.1.3 Serial Programming Mode

This mode is initiated by an external programming device. Use it to store application code or application data in the flash memories of the device. After reset release, the MCU boots up from the on-chip boot firmware and establishes a connection in the specified transmission method. For details, see **Section 41, Flash Memory**.

5.2 Input Pins

Table 5.2 shows the pin information for mode setting. The states of all the mode pins are latched in the Mode Register.

Table 5.2 Pin Information for Mode Setting

Pin Name	I/O	Function
MD1	Input	Operating mode select pin
MD0	Input	Operating mode select pin
$\overline{\text{TRST}}$	Input	Shared pin used as Nexus/JTAG reset pin and operating mode select pin

5.3 Register Description

5.3.1 List of Registers

The following table lists the operating mode registers.

Table 5.3 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
SYCTRL	Mode Register	MODE	FF70 0000 _H	32	
SYCTRL	FHVE3 Control Register	FHVE3	FF70 3800 _H	32	
SYCTRL	FHVE15 Control Register	FHVE15	FF70 3804 _H	32	

Table 5.4 Register Reset Condition

Unit Name	Register Name	Reset Source					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
MODE	ALL Registers	✓	✓*1	—	—	—	—

Note 1. VMON Reset is excluded.

5.3.2 MODE — Mode Register

This register indicates the Operating Mode of the device.

The states of the mode pins and mode select option bytes are latched in this register when the external reset or standby reset is released.

Value after reset: 000x 000x_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STMSEL1	STMSEL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MD1	MD0	TRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 5.5 Mode Register Contents

Bit Position	Bit Name	Function
31 to 18	—	Reserved. When read, the value after reset is returned. When writing, write the value after reset.
17	STMSEL1	STMSEL1 This bit indicates the status of the Option byte STMSEL1 0: Low-level detected 1: High-level detected
16	STMSEL0	STMSEL0 This bit indicates the status of the Option byte STMSEL0 0: Low-level detected 1: High-level detected
15 to 3	—	Reserved. When read, the value after reset is returned. When writing, write the value after reset.
2	MD1	MD 1 This bit indicates the level of the signal latched for Mode Pin MD 1 0: Low-level detected 1: High-level detected
1	MD0	MD 0 This bit indicates the level of the signal latched for Mode Pin MD 0 0: Low-level detected 1: High-level detected
0	TRST	TRST This bit indicates the level of the signal latched for Mode Pin TRST 0: Low-level detected 1: High-level detected

5.3.3 FHVE3 — FHVE3 Control Register

Please refer to **Section 41, Flash Memory** for details on this register.

5.3.4 FHVE15 — FHVE15 Control Register

Please refer to **Section 41, Flash Memory** for details on this register.

5.4 Mode Error

This LSI has the ability to notify the ECM when it detects an unintended activation of a different mode or unintended de-activation of the currently selected mode. The MD1, MD0 and TRST pins are monitored, along with the STMSEL0 and STMSEL1 bits to detect unintended changes. If these values change when they are not expected to change, a mode check error occurs.

The unintended modes are as follows. For details, see **Section 39, Error Control Module (ECM)**.

Detectable unintended modes are shown in **Table 5.6**.

Table 5.6 List of Detectable Mode Errors

User Intended Mode	Detectable Mode Error
Normal Operation Mode	Unintended deactivation of Normal Operation Mode
	Unintended activation of User Boot Mode
User Boot Mode	Unintended activation of Normal Operation Mode
	Unintended deactivation of User Boot Mode
Normal Operation Mode / User Boot Mode	Unintended activation of Serial Programming Mode
	Unintended activation of Production Test Mode
Other than On chip debug mode	Unintended Debug Enabled
any mode	The detection of differences between the latched value and from Flash.

NOTE

Although it is possible to change the setting of MD0 or $\overline{\text{TRST}}$ (e.g. for the purpose of selecting communication method) after RES_IN is released, a mode error is also detected when MD0 or TRST is changed after RES_IN is released in serial programming mode. MD1 shall be fixed to "0" in user mode.

Mode check error functions work only when the settings of the external terminals and STMSEL1, STMSEL0 correspond with Normal operation mode or User boot mode under the condition of $\overline{\text{TRST}} = 0$.

Section 6 Interrupts

This section describes the interrupt units.

6.1 Features of Interrupt Units

6.1.1 Units and Channels

This microcontroller has the following number of interrupt units.

Table 6.1 Number of Units (INTC1)

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Units per PE	1	1
Unit Name	INTC1	

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Units per PE	1	1
Unit Name	INTC1	

Table 6.2 Number of Units (INTC2)

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Units	1	1
Unit Name	INTC2	

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Units	1	1
Unit Name	INTC2	

Table 6.3 Number of Units (INTIF)

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Units	1	1
Unit Name	INTIF	

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Units	1	1
Unit Name	INTIF	

Table 6.4 Number of Units (EINT)

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Units	1	1
Unit Name	EINT	

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Units	1	1
Unit Name	EINT	

6.1.2 Register Base Address

The register base addresses of the interrupt units are listed in **Table 6.5**.

The address of each register in this section is specified as an offset from one of these base addresses.

Table 6.5 Register Base Address

Base Address Name	SELF / PE*	Base Address	Bus Group
<INTC1_base>	INTC1 of SELF	FFFC 0000 _H	—
<INTC1_base>	INTC1 of PE0	FFFC 4000 _H	CPU Peripheral 0
<INTC1_base>	INTC1 of PE1	FFFC 8000 _H	CPU Peripheral 1
<INTC2_base>	—	FFF8 0000 _H	Peripheral Group 0
<INTIF_base>	—	FF09 0000 _H	Peripheral Group 9
<EINT_base>	—	FFC0 0000 _H	Peripheral Group 6

6.1.3 Clock Supply

Clock supplies for each interrupt unit are shown in the following table.

Table 6.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
INTC1	clk	CLK_CPU
INTC2	pe_clk	CLK_CPU
	bus_clk	CLK_HBUS
INTIF	pclk	CLK_HBUS
EINT	clk_lsb	CLK_LSB

6.1.4 Interrupt Requests

The INTC2 interrupt requests are listed in the following table.

Table 6.7 Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
INTBRDC0	INT_BRDC0	Broadcast notification 0	4	—	—
INTBRDC1	INT_BRDC1	Broadcast notification 1	5	—	—
INTBRDC2	INT_BRDC2	Broadcast notification 2	6	—	—
INTBRDC3	INT_BRDC3	Broadcast notification 3	7	—	—
INTEINTSW0	intreq_sint[0]	Software interrupt 0	14	—	—
INTEINTSW1	intreq_sint[1]	Software interrupt 1	15	—	—
INTEINTSW2	intreq_sint[2]	Software interrupt 2	16	—	—
INTEINTSW3	intreq_sint[3]	Software interrupt 3	17	—	—

6.1.5 Reset Sources

The reset sources for initialization of the interrupt units are shown in **Table 6.8**.

Table 6.8 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
INTC1	All registers*1	✓	✓	✓	✓	—	—
INTC2	All registers*1	✓	✓	✓	✓	—	—
INTIF	All registers*1	✓	✓	✓	✓	—	—
EINT	All registers*1	✓	✓	✓	✓	—	—

Note 1. See **Table 6.12**.

6.1.6 External Input/Output Signals

External input/output signals of the interrupt units are listed below.

Table 6.9 External Input/Output Signals

Unit Signal Name	Direction	Outline	Alternative Port Pin Signal
EINT			
IRQ0	Input	External pin IRQ0	IRQ0
IRQ1	Input	External pin IRQ1	IRQ1
IRQ2	Input	External pin IRQ2	IRQ2
IRQ3	Input	External pin IRQ3	IRQ3
IRQ4	Input	External pin IRQ4	IRQ4
IRQ5	Input	External pin IRQ5	IRQ5
IRQ6	Input	External pin IRQ6	IRQ6
IRQ7	Input	External pin IRQ7	IRQ7
FENMI	Input	Non-maskable interrupt	NMI

6.2 Overview

6.2.1 Outline

INTC features

- Interrupt sources:
 - Non-maskable interrupt (FENMI): 1 channel for each CPU
 - FE level interrupt (FEINT0 to 1): 2 channels for each CPU
 - Low latency EI level interrupt (maskable) EIINT0 to 31:
 - 32 channels for each CPU
 - Inter-processor interrupts: 4 channels shared on channels 0 to 3
 - Broadcast interrupts: 4 channels shared on channels 4 to 7
 - Low speed interrupts (maskable) (EIINT32 to 511):
 - 480 channels shared by all CPUs
- Interrupt priority levels:
 - Up to 16 interrupt priority levels can be set by the interrupt control registers.
- Interrupt detection method:
 - The detection method of FEINT and EIINT interrupts are individually set to edge detection or level detection.
- Register banks:
 - For details about register banks, see **Section 3.2.4.5, Register Bank Function**.

INTC consists of INTC1 and INTC2

- INTC1
 - All CPUs have their own interrupt controller, INTC1. Each CPU accesses the INTC1 that corresponds to that CPU. INTC1 controls low-latency interrupts and has the following functions:
 - Priority setting
 - Interrupt mask setting
- INTC2
 - INTC2 is a common interrupt controller that all the CPUs share. INTC2 controls low-speed interrupts and has the following functions:
 - Priority setting
 - Interrupt mask setting
 - CPU binding setting
 - Broadcast interrupt setting

Others

- INTIF
 - Peripheral interrupt / TPTM interrupt control function
- EINT
 - External interrupt / Software interrupt / NMI control function

6.2.2 Functional Overview

The act of branching from a currently running program to a different program in response to an event is called an exception. This microcontroller supports the following types of exceptions. Exceptions are described in detail in **Section 3.2.4, Exceptions and Interrupts**.

The following three exceptions are called interrupts, and are described in this section:

- FE level Non-Maskable Interrupt (FENMI)

An FENMI interrupt is acknowledged even if another FE level interrupt, FEINT, has been generated.

- An FENMI interrupt is acknowledged even if the CPU system register PSW.NP = 1.
- Return from an FENMI interrupt is not possible and recovery is disabled, when multiple interrupts occur.

- FE level maskable interrupt (FEINT)

An FEINT interrupt can be acknowledged only if an FE level non-maskable interrupt, FENMI, has not been generated.

- An FEINT can be acknowledged if the CPU system register PSW.NP = 0.
- Return is enabled and recovery is enabled.

- EI level maskable interrupt (EIINT)

An EIINT interrupt can be acknowledged only if an FE level interrupt, FENMI or FEINT, has not been generated.

- An EIINT can be acknowledged if the CPU system registers PSW.NP = 0 and PSW.ID = 0, and when the following two conditions are satisfied:
 - The interrupt priority level is higher than the priority level of the acknowledgment status, which is set to the CPU system register ISPR.
 - The interrupt priority level is higher than the priority level, which is set to the CPU system register PLMR.
- Return is enabled and recovery is enabled.
- Interrupt masking can be specified for each interrupt channel.
- One of 16 interrupt priority levels can be specified for each interrupt channel.

For details about the PSW register, see **Section 3.2.3.2(5), PSW — Program Status Word**.

6.2.3 Interrupt Sources

6.2.3.1 FE Level Non-Maskable Interrupts

See **Section 3.2.4, Exceptions and Interrupts** for details about the priority order, return PC, status registers and return instructions.

Table 6.10 Non-Maskable Interrupt Requests

Interrupt	Interrupt Request		Unit	Priority Order	Detection Type	Exception Cause Code	Offset Address
Symbol	Name	Cause					
FENMI	NMI	NMI pin	Port	*1	Edge	0E0 _H	0E0 _H

Note 1. See **Section 3.2.4, Exceptions and Interrupts**.

6.2.3.2 FE Level Maskable Interrupts

See **Section 3.2.4, Exceptions and Interrupts** for details about the priority order, return PC, status registers and return instructions.

Table 6.11 FE Level Interrupt Requests

Interrupt	Interrupt Request		Unit	Priority Order	Detection Type	Exception Cause Code	Offset Address
Symbol	Cause						
	PE0						
FEINT0	FE level interrupt from the ECM		ECM	*1	Edge	F0 _H	F0 _H
FEINT1	Interrupt from Time Protection Timer for PE0*2		Time Protection Timer	*1	Level	F1 _H	F0 _H
	PE1						
FEINT0	FE level interrupt from the ECM		ECM	*1	Edge	F0 _H	F0 _H
FEINT1	Interrupt from Time Protection Timer for PE1*2		Time Protection Timer	*1	Level	F1 _H	F0 _H

Note 1. See **Section 3.2.4, Exceptions and Interrupts**.

Note 2. This interrupt can be used as an EIINT. See **Section 6.3.13, TPTMSEL — TPTM Interrupt FE EI Select Register**.

6.2.3.3 EI Level Maskable Interrupts

See **Section 3.2.4, Exceptions and Interrupts** for details about the priority order, return PC and return instructions.

See **Section 6.3.2, EIC0 to EIC511 — EI Level Interrupt Control Register 0 to 511** for details about the control registers.

For details, refer to Appendix file “Interrupt_table.xlsx”.

6.2.4 Block Diagram

The block diagram of the interrupt units is shown in **Figure 6.1**.

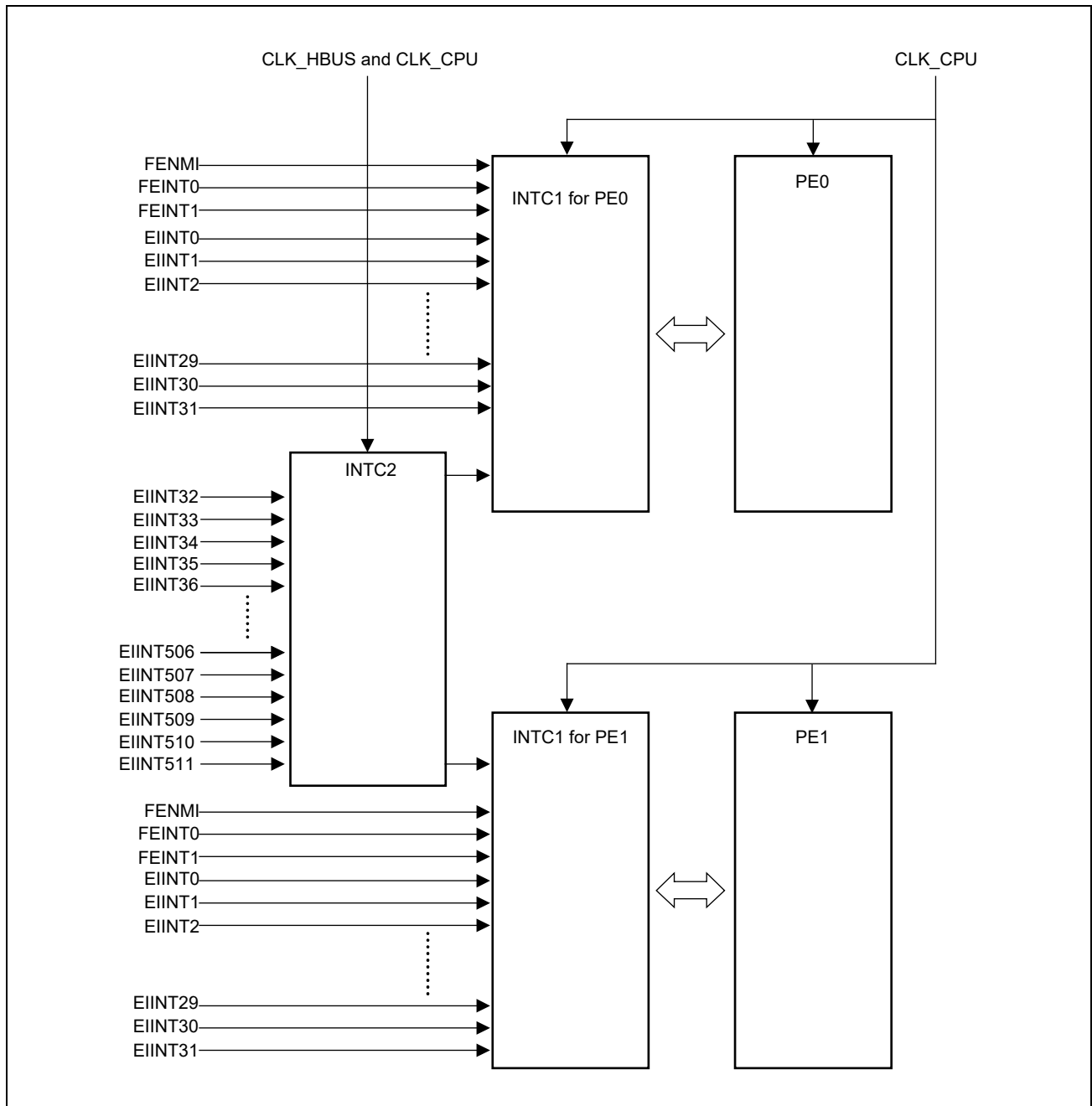


Figure 6.1 Block Diagram of Interrupt Units

6.3 Registers

6.3.1 List of Registers

Table 6.12 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protect
INTC1	EI Level Interrupt Control Register* ¹	EICn (n = 0 to 31)	<INTC1_base> + 0000 _H + 02 _H × n	8, 16	—
	EI Level Interrupt Mask Register* ¹	IMR0	<INTC1_base> + 00F0 _H	8, 16, 32	—
	EI Level Interrupt Bind Register* ¹	EIBDn (n = 0 to 31)	<INTC1_base> + 0100 _H + 04 _H × n	32	—
INTC2	EI Level Interrupt Control Register* ¹	EICn (n = 32 to 511)	<INTC2_base> + 0000 _H + 02 _H × n	8, 16	—
	EI Level Interrupt Mask Register* ¹	IMRn (n = 1 to 15)	<INTC2_base> + 1000 _H + 04 _H × n	8, 16, 32	—
	EI Level Interrupt Bind Register* ¹	EIBDn (n = 32 to 511)	<INTC2_base> + 2000 _H + 04 _H × n	32	—
INTIF	Peripheral Interrupt Status Register	PINTn (n = 0 to 7)	<INTIF_base> + 00 _H + 04 _H × n	32	—
	Peripheral Interrupt Status Clear Register	PINTCLRn (n = 0 to 7)	<INTIF_base> + 20 _H + 04 _H × n	32	—
	TPTM Interrupt FE EI Select Register	TPTMSEL	<INTIF_base> + C0 _H	8, 16, 32	—
EINT	NMI Interrupt Control Register	NMICTL	<EINT_base>	8	—
	External Interrupt Control Register	EXINTCTL	<EINT_base> + 10 _H	8, 16	—
	External Interrupt Status Register	EXINTSTR	<EINT_base> + 14 _H	8	—
	External Interrupt Status Clear Register	EXINTSTC	<EINT_base> + 18 _H	8	—
	Software Interrupt Register 0	SINTR0	<EINT_base> + 20 _H	8	—
	Software Interrupt Register 1	SINTR1	<EINT_base> + 24 _H	8	—
	Software Interrupt Register 2	SINTR2	<EINT_base> + 28 _H	8	—
Software Interrupt Register 3	SINTR3	<EINT_base> + 2C _H	8	—	

Note 1. It is recommended to execute a DI instruction to avoid incorrect detection of interrupts when the settings of these registers are changed.

6.3.2 EIC0 to EIC511 — EI Level Interrupt Control Register 0 to 511

These registers define control of EI level interrupts. There is one register for each EI interrupt source. For a complete list of interrupt sources, see Appendix file “Interrupt_table.xlsx”.

CAUTION

If the EIRFn bit is set to 0 immediately after a peripheral module generates the corresponding interrupt request upon detection of an edge (before an interrupt is acknowledged by the CPU), the request may be lost.

If the EIRFn bit is set to 1 immediately after an interrupt is acknowledged by the CPU, a new interrupt request will be generated.

This also applies to bit manipulation instructions (set1, clr1, not1) described in **Section 3.8.3, Accesses to Registers by Bit-Manipulation Instructions**.

Accessing only the lower byte of this register (including EIMKn) by a bit-manipulation instruction does not modify the EIRFn bit.

Value after reset: 008FH (edge detection), 808FH (level detection)*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EICTn	—	—	EIRFn	—	—	—	—	EIMKn	EITBn	—	—	EIPn[3:0]			
Value after reset	0/1*1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1
R/W	R	R	R	R/W	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Note 1. Interrupt detection by edge: 0_B
 Interrupt detection by level: 1_B

Table 6.13 EIC0 to EIC511 Register Contents (1/2)

Bit Position	Bit Name	Function
15	EICTn	This bit indicates the type of interrupt detection. This bit is read only. 0: Detection by edge 1: Detection by level When writing in 8-bit or 16-bit units, write the value after a reset.
14, 13	—	Reserved These bits are always read as 0. The write value should always be 0.
12	EIRFn	Interrupt Request Flag Operation varies depending on the interrupt input interface. 0: No interrupt request (Initial value) 1: Interrupt request present <ul style="list-style-type: none"> • Edge detection This flag is automatically cleared to 0 when an interrupt request from its own channel is acknowledged by the CPU core. • Level detection This bit cannot be set or cleared by the software. This is a read-only bit.
11 to 8	—	Reserved These bits are always read as 0. The write value should always be 0.

Table 6.13 EIC0 to EIC511 Register Contents (2/2)

Bit Position	Bit Name	Function
7	EIMKn	<p>Interrupt Mask</p> <p>If this bit is set to 1, interrupt requests set by the interrupt request flag (EIRFn) are masked to inhibit interrupt requests from the channel to the CPU core. The presence of unprocessed interrupts is not reported and the PME1 bit in ICSR is not set for the channel for which this bit is set to 1. Even when interrupt processing is disabled by the setting of this bit, the input of an interrupt signal is not masked and the interrupt request flag is set. The state of this bit is also reflected in the Interrupt Mask Register (IMR).</p> <p>When the interrupt request from the channel is masked with EIMKn = 1, the EIRFn still reflects the interrupt request for the channel and can be polled in software. When the EIMKn bit is cleared, interrupt requests from the channel are issued to the CPU core for subsequent processing. The state of the EIMKn bit is also reflected in the corresponding IMRm register.</p> <p>0: Interrupt processing is enabled. 1: Interrupt processing is disabled. (Initial value)</p>
6	EITBn	<p>Interrupt Vector Method Select</p> <p>0: Direct vector method 1: Table reference method</p>
5, 4	—	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
3 to 0	EIPn	<p>These bits specify 16 interrupt priority levels (0: highest priority, 15: lowest priority).</p> <p>If two or more EI level interrupt requests are generated simultaneously, the source with a higher priority specified by these bits is selected and is sent to the CPU core. If the priorities specified by these bits are equal, the source with the lower default priority channel number is selected.</p>

Note: n = 0 to 511

The values for the given channel numbers listed as reserved in Appendix file "Interrupt_table.xlsx" must not be set to anything other than their values after a reset.

NOTE

When a channel n is defined as broadcast interrupt (EIBDn.CST = 1), EIMKn and EIRFn bits of the EICn register of the channel must be set to 0 after an initial configuration of the channel. In the period when the EIINTn interrupt is enabled, it is prohibited to mask (EIMKn = 1) an interrupt processing of the channel. When it is necessary to mask a broadcast interrupt, EIC4 to EIC7 registers in INTC1 of each PE can be used to mask the corresponding broadcast interrupt.

6.3.3 IMR0 to IMR15 — EI Level Interrupt Mask Register 0 to 15

These registers are an aggregation of the EIMK bits from the EIC registers. The setting of the EIMK bit in the EIC register is reflected in this register. Also the setting of a bit in this register is reflected in the EIMK bit of the corresponding EIC register.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EIMK (m × 32 + 31)	EIMK (m × 32 + 30)	EIMK (m × 32 + 29)	EIMK (m × 32 + 28)	EIMK (m × 32 + 27)	EIMK (m × 32 + 26)	EIMK (m × 32 + 25)	EIMK (m × 32 + 24)	EIMK (m × 32 + 23)	EIMK (m × 32 + 22)	EIMK (m × 32 + 21)	EIMK (m × 32 + 20)	EIMK (m × 32 + 19)	EIMK (m × 32 + 18)	EIMK (m × 32 + 17)	EIMK (m × 32 + 16)
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EIMK (m × 32 + 15)	EIMK (m × 32 + 14)	EIMK (m × 32 + 13)	EIMK (m × 32 + 12)	EIMK (m × 32 + 11)	EIMK (m × 32 + 10)	EIMK (m × 32 + 9)	EIMK (m × 32 + 8)	EIMK (m × 32 + 7)	EIMK (m × 32 + 6)	EIMK (m × 32 + 5)	EIMK (m × 32 + 4)	EIMK (m × 32 + 3)	EIMK (m × 32 + 2)	EIMK (m × 32 + 1)	EIMK (m × 32 + 0)
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 6.14 IMR0 to IMR15 Register Contents

Bit Position	Bit Name	Function
31 to 0	EIMK (m × 32 + 31) to EIMK (m × 32 + 0)	These are interrupt mask bits for EI level maskable interrupt (INT) channels 0 to 511. 0: Interrupt processing enabled 1: Interrupt processing disabled

Note: m = 0 to 15

The value of the EIMK bits for the given channel numbers listed as reserved in Appendix file “Interrupt_table.xlsx” must be set to 1.

NOTE

When writing to the EIMK bit by writing the IMR1 to 15 registers, only access from the PE set in the EIBDn.PEID (n = 32 to 511) register in the corresponding channel is possible. The EIMK bit is not updated when writing from a different PE.

6.3.4 EIBD0 to EIBD31 — EI Level Interrupt Bind Register 0 to 31

These registers are provided for each EI level interrupt source to define binding between interrupt sources and PEs. For details on each of the interrupt sources, see Appendix file “Interrupt_table.xlsx”.

Value after reset:		0000 0000 _H (PE0)														
		0000 0001 _H (PE1)														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PEID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 6.15 EIBD0 to EIBD31 Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	PEID[2:0]	These bits specify the interrupt bind (request) destination. These bits are fixed in EIBD0 to EIBD31 and cannot be modified. 000: Interrupt is bound to PE0. 001: Interrupt is bound to PE1.

Note 1. The values for the given channel numbers listed as reserved in Appendix file “Interrupt_table.xlsx” must not be set to anything other than their values after a reset.

NOTE

Changing the corresponding EIBD_n register during the processing of an EIINT request is prohibited.

Note: n = 0 to 31

6.3.5 EIBD32 to EIBD511 — EI Level Interrupt Bind Register 32 to 511

These registers are provided for each EI level interrupt source to define a binding between interrupt sources 32 to 511 and CPU cores. For a list of interrupt sources, see Appendix file “Interrupt_table.xlsx”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CST	—	—	—	—	—	BCP[1:0]		—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PEID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 6.16 EIBD32 to EIBD511 Register Contents

Bit Position	Bit Name	Function
31	CST	Broadcast interrupt enable.
30 to 26	—	Reserved These bits are always read as 0. The write value should always be 0.
25 to 24	BCP[1:0]	When the CST bit is 1, these bits specify a broadcast interrupt port number setting. When the CST bit is 0, these bits do not specify anything.
23 to 3	—	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	PEID[2:0]	These bits specify the interrupt bind (request) destination. 000: Interrupt is bound to PE0. 001: Interrupt is bound to PE1. Other than above: Setting prohibited To execute an interrupt of the corresponding source, be sure to set one of the above values.

Note: The values for the given channel numbers listed as reserved in Appendix file “Interrupt_table.xlsx” must not be set to anything other than their values after a reset.

NOTE

Changing the corresponding EIBD_n register during the processing of an EIINT request is prohibited.

Note: n = 32 to 511

6.3.6 NMICL — NMI Interrupt Control Register

NMICL is an 8-bit register used to define the active edge for the NMI interrupt input. The active edge can be either rising or falling.

Value after reset: 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	NMIS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 6.17 NMICL Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned.
0	NMIS	NMI Interrupt Detecting Method Select This bit selects the method of detecting an NMI interrupt signal from either falling edge or rising edge. 0: An interrupt request is detected by the NMI input falling edge. 1: An interrupt request is detected by the NMI input rising edge.

6.3.7 EXINTCTL — External Interrupt Control Register

EXINTCTL is a 16-bit register. This register defines the detection method for external interrupt input pins IRQ0 to IRQ7. Detection methods are edge (rising or falling) or level (low or high).

Value after reset: 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ7S		IRQ6S		IRQ5S		IRQ4S		IRQ3S		IRQ2S		IRQ1S		IRQ0S	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 6.18 EXINTCTL Register Contents

Bit Position	Bit Name	Function
15 to 14	IRQ7S	External Interrupt Detecting Method Select
13 to 12	IRQ6S	These bits select the method of detecting an IRQ7 to IRQ0 interrupt signal from either low level, high level, falling edge, or rising edge.
11 to 10	IRQ5S	
9 to 8	IRQ4S	00: An interrupt request is detected by the IRQn input low level.*1
7 to 6	IRQ3S	01: An interrupt request is detected by the IRQn input high level.*1
5 to 4	IRQ2S	10: An interrupt request is detected by the IRQn input falling edge.
3 to 2	IRQ1S	11: An interrupt request is detected by the IRQn input rising edge.
1 to 0	IRQ0S	n = 0 to 7

Note 1. When level detection is selected, the active level should be retained until an interrupt request is acknowledged.

6.3.8 EXINTSTR — External Interrupt Status Register

EXINTSTR is an 8-bit register. This register indicates the presence of external interrupt requests IRQ0 to IRQ7. When an “edge” is specified as the detection method, interrupt requests retained in this register can be cleared by the EXINTSTC register.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 6.19 EXINTSTR Register Contents

Bit Position	Bit Name	Function
7	IRQ7F	External Interrupt Requests
6	IRQ6F	These bits indicate the IRQ7 to IRQ0 interrupt request status.
5	IRQ5F	<ul style="list-style-type: none"> When level detection is selected
4	IRQ4F	<ul style="list-style-type: none"> 0: No IRQn interrupt request is present. 1: IRQn interrupt request is present.
3	IRQ3F	[Clearing condition]
2	IRQ2F	The IRQn input is not at the level corresponding to the conditions specified in IRQnS in EXINTCTL
1	IRQ1F	[Setting condition]
0	IRQ0F	<ul style="list-style-type: none"> The IRQn input is at the level corresponding to the conditions specified in IRQnS in EXINTCTL When edge detection is selected 0: No IRQn interrupt request has been detected 1: IRQn interrupt request has been detected
		[Clearing condition]
		Writing 1 to the IRQnC bit in EXINTSTC
		[Setting condition]
		An edge corresponding to the conditions specified in IRQnS in EXINTCTL is generated

Note: n = 0 to 7

6.3.9 EXINTSTC — External Interrupt Status Clear Register

EXINTSTC is an 8-bit register used to clear IRQnF in EXINTSTR when edge detection is selected as the method of detecting IRQn. Writing 1 to IRQnC clears the corresponding IRQnF bit in EXINTSTR.

Value after reset: 00H

Bit	7	6	5	4	3	2	1	0
	IRQ7C	IRQ6C	IRQ5C	IRQ4C	IRQ3C	IRQ2C	IRQ1C	IRQ0C
Value after reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Table 6.20 EXINTSTC Register Contents

Bit Position	Bit Name	Function
7	IRQ7C	External Interrupt Request Clear
6	IRQ6C	These bits clear the interrupt request status when the edge detection is selected for detecting IRQ7 to IRQ0 interrupts.
5	IRQ5C	
4	IRQ4C	• When level detection is selected These bits have no function.
3	IRQ3C	• When edge detection is selected
2	IRQ2C	Writing 1 clears the corresponding IRQnF bit in the EXINTSTR register.
1	IRQ1C	
0	IRQ0C	

Note: n = 0 to 7

6.3.10 SINTR0 to SINTR3 — Software Interrupt Register

SINTR0 to SINTR3 are 8-bit registers used to control software interrupts 0 to 3 (SINT0 to SINT3).

Writing 01_H to these registers increments the value of the counter; writing 00_H decrements it. When the value of the counter for any of these registers is 1 or more, the corresponding interrupt from among software interrupts 0 to 3 (SINT0 to SINT3) is generated. When reading the value from any of these registers, the read value is the current value of the counter.

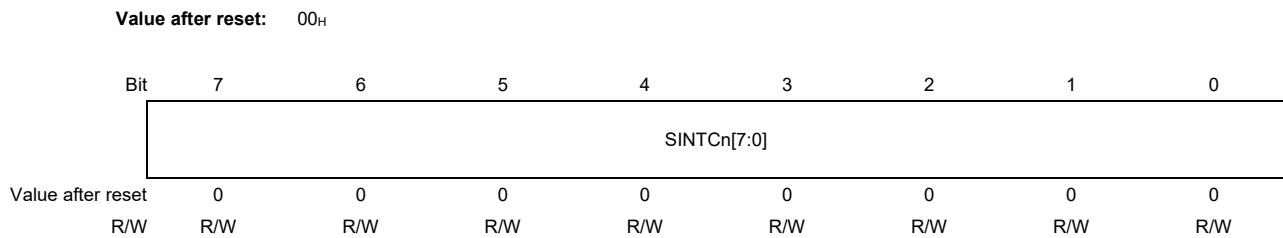


Table 6.21 SINTR0 to SINTR3 Register Contents

Bit Position	Bit Name	Function
7 to 0	SINTCn[7:0]	Software Interrupt Request This bit generates a software interrupt. [Reading operation] The number of SINTn interrupt request counts is read out. [Writing operation] Writing 01 _H : Increments the counter.*1 Writing 00 _H : Decrements the counter.*2

Note 1. When 01_H is written to the register while the value of the counter is FF_H, the counter is not incremented and its value remains FF_H.

Note 2. When 00_H is written to the register while the value of the counter is 00_H, the counter is not decremented and its value remains 00_H.

6.3.11 PINTn + x — Peripheral Interrupt Status Register

All 128 DTS interrupt flags are divided between 8 registers PINT0 to PINT7. Each register contains 32 interrupt flags. Only one bit in each register PINT0 to PINT7 can be set. If multiple DTS channels within the same PINTn register request an interrupt, only the bit of the lowest DTS channel in that register is set.

Value after reset: 0000 0000_H

• PINT n + x (n = 0 to 3, x = 0)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTDTS [31 + 32 * n]	INTDTS [30 + 32 * n]	INTDTS [29 + 32 * n]	INTDTS [28 + 32 * n]	INTDTS [27 + 32 * n]	INTDTS [26 + 32 * n]	INTDTS [25 + 32 * n]	INTDTS [24 + 32 * n]	INTDTS [23 + 32 * n]	INTDTS [22 + 32 * n]	INTDTS [21 + 32 * n]	INTDTS [20 + 32 * n]	INTDTS [19 + 32 * n]	INTDTS [18 + 32 * n]	INTDTS [17 + 32 * n]	INTDTS [16 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTDTS [15 + 32 * n]	INTDTS [14 + 32 * n]	INTDTS [13 + 32 * n]	INTDTS [12 + 32 * n]	INTDTS [11 + 32 * n]	INTDTS [10 + 32 * n]	INTDTS [9 + 32 * n]	INTDTS [8 + 32 * n]	INTDTS [7 + 32 * n]	INTDTS [6 + 32 * n]	INTDTS [5 + 32 * n]	INTDTS [4 + 32 * n]	INTDTS [3 + 32 * n]	INTDTS [2 + 32 * n]	INTDTS [1 + 32 * n]	INTDTS [0 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 6.22 PINTn + x Register Contents (n = 0 to 3, x = 0)

Bit Position	Bit Name	Function
31 to 0	INTDTS	DTS ch (0 + 32 × n) to (31 + 32 × n) transfer completion interrupt status 0: No transfer complete interrupt request present 1: Transfer complete interrupt request present

• PINT n + x (n = 0 to 3, x = 4)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTCDT S [31 + 32 * n]	INTCDT S [30 + 32 * n]	INTCDT S [29 + 32 * n]	INTCDT S [28 + 32 * n]	INTCDT S [27 + 32 * n]	INTCDT S [26 + 32 * n]	INTCDT S [25 + 32 * n]	INTCDT S [24 + 32 * n]	INTCDT S [23 + 32 * n]	INTCDT S [22 + 32 * n]	INTCDT S [21 + 32 * n]	INTCDT S [20 + 32 * n]	INTCDT S [19 + 32 * n]	INTCDT S [18 + 32 * n]	INTCDT S [17 + 32 * n]	INTCDT S [16 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTCDT S [15 + 32 * n]	INTCDT S [14 + 32 * n]	INTCDT S [13 + 32 * n]	INTCDT S [12 + 32 * n]	INTCDT S [11 + 32 * n]	INTCDT S [10 + 32 * n]	INTCDT S [9 + 32 * n]	INTCDT S [8 + 32 * n]	INTCDT S [7 + 32 * n]	INTCDT S [6 + 32 * n]	INTCDT S [5 + 32 * n]	INTCDT S [4 + 32 * n]	INTCDT S [3 + 32 * n]	INTCDT S [2 + 32 * n]	INTCDT S [1 + 32 * n]	INTCDT S [0 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 6.23 PINTn + x Register Contents (n = 0 to 3, x = 4)

Bit Position	Bit Name	Function
31 to 0	INTCDTS	DTS ch (0 + 32 × n) to (31 + 32 × n) transfer count match interrupt status 0: No transfer count match interrupt request present 1: Transfer count match Interrupt request present

6.3.12 PINTCLRn + x — Peripheral Interrupt Status Clear Register

Writing a value from the interrupt read register (PINTn) to the corresponding interrupt clear register (PINTCLRn) will clear that interrupt.

Value after reset: 0000 0000_H

• PINTCLR n + x (n = 0 to 3, x = 0)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTCLR [31 + 32 * n]	INTCLR [30 + 32 * n]	INTCLR [29 + 32 * n]	INTCLR [28 + 32 * n]	INTCLR [27 + 32 * n]	INTCLR [26 + 32 * n]	INTCLR [25 + 32 * n]	INTCLR [24 + 32 * n]	INTCLR [23 + 32 * n]	INTCLR [22 + 32 * n]	INTCLR [21 + 32 * n]	INTCLR [20 + 32 * n]	INTCLR [19 + 32 * n]	INTCLR [18 + 32 * n]	INTCLR [17 + 32 * n]	INTCLR [16 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTCLR [15 + 32 * n]	INTCLR [14 + 32 * n]	INTCLR [13 + 32 * n]	INTCLR [12 + 32 * n]	INTCLR [11 + 32 * n]	INTCLR [10 + 32 * n]	INTCLR [9 + 32 * n]	INTCLR [8 + 32 * n]	INTCLR [7 + 32 * n]	INTCLR [6 + 32 * n]	INTCLR [5 + 32 * n]	INTCLR [4 + 32 * n]	INTCLR [3 + 32 * n]	INTCLR [2 + 32 * n]	INTCLR [1 + 32 * n]	INTCLR [0 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 6.24 PINTCLRn + x Register Contents (n = 0 to 3, x = 0)

Bit Position	Bit Name	Function
31 to 0	INTCLR	These bits clear the DTS ch (0 + 32 × n) to (31 + 32 × n) transfer completion interrupt status. These bits should be written with the value corresponding to the PINT(n + x) register inside an interrupt handler routine.

• PINTCLR n + x (n = 0 to 3, x = 4)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTCTCL R [31 + 32 * n]	INTCTCL R [30 + 32 * n]	INTCTCL R [29 + 32 * n]	INTCTCL R [28 + 32 * n]	INTCTCL R [27 + 32 * n]	INTCTCL R [26 + 32 * n]	INTCTCL R [25 + 32 * n]	INTCTCL R [24 + 32 * n]	INTCTCL R [23 + 32 * n]	INTCTCL R [22 + 32 * n]	INTCTCL R [21 + 32 * n]	INTCTCL R [20 + 32 * n]	INTCTCL R [19 + 32 * n]	INTCTCL R [18 + 32 * n]	INTCTCL R [17 + 32 * n]	INTCTCL R [16 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTCTCL R [15 + 32 * n]	INTCTCL R [14 + 32 * n]	INTCTCL R [13 + 32 * n]	INTCTCL R [12 + 32 * n]	INTCTCL R [11 + 32 * n]	INTCTCL R [10 + 32 * n]	INTCTCL R [9 + 32 * n]	INTCTCL R [8 + 32 * n]	INTCTCL R [7 + 32 * n]	INTCTCL R [6 + 32 * n]	INTCTCL R [5 + 32 * n]	INTCTCL R [4 + 32 * n]	INTCTCL R [3 + 32 * n]	INTCTCL R [2 + 32 * n]	INTCTCL R [1 + 32 * n]	INTCTCL R [0 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 6.25 PINTCLRn + x Register Contents (n = 0 to 3, x = 4)

Bit Position	Bit Name	Function
31 to 0	INTCTCLR	These bits clear the DTS ch (0 + 32 × n) to (31 + 32 × n) transfer count match interrupt status. These bits should be written with the value corresponding to the PINT(n + x) register inside an interrupt handler routine.

6.3.13 TPTMSEL — TPTM Interrupt FE EI Select Register

This register selects how TPTM interrupts are mapped in each CPU core.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TPTMS EL1	TPTMS EL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 6.26 TPTMSEL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved These bits are always read as 0. The write value should always be 0.
1	TPTMSEL1	This bit selects whether the FEINT or the EIINT is the TPTM interrupt of PE1. 0: the TPTM interrupt of PE1 is connected to FEINT. 1: the TPTM interrupt of PE1 is connected to EIINT
0	TPTMSEL0	This bit selects whether the FEINT or the EIINT is the TPTM interrupt of PE0. 0: the TPTM interrupt of PE0 is connected to FEINT. 1: the TPTM interrupt of PE0 is connected to EIINT

6.4 Interrupt Operation

6.4.1 Level Interrupts

For the operation of level interrupts, see **Section 6.5.1, Level Interrupt Processing Flow**.

6.4.2 External Interrupts (NMI / IRQ)

NMI and IRQ interrupts are input from external devices. Detection methods for these interrupts can be chosen from four methods (two methods for NMI and all four methods for IRQ).

6.4.3 Inter-Processor Interrupts

For the operation of Inter-Processor Interrupts, see **Section 3.4, Inter-Processor Interrupts**.

6.4.4 Broadcast Interrupts

INTC2 has a function for broadcast interrupts. With this function, the EIINTn request input is transferred to EIINT4-7 of each CPU's INTC1 as a broadcast notification 0-3 with no priority judgement.

The interrupt detection type of the EIINTn must be edge detection if the broadcast function is used. The broadcast function cannot be activated (EIBDn.CST cannot be set) if the interrupt detection type of the EIINTn is configured as level detection.

When EIBDn.CST is set, EIINTn is used as the broadcast interrupt source.

EIBDn.BCP[1:0] determines which port of the broadcast interrupt (EIINT4-7) is triggered. It is prohibited to assign more than one interrupt channel to the same broadcast interrupt (EIINT4-7).

For the operation of broadcast interrupts, see **Section 6.3.5, EIBD32 to EIBD511 — EI Level Interrupt Bind Register 32 to 511** and **Section 6.5.4, Broadcast Interrupt Processing Flow**.

6.4.5 Software Interrupts

For the operation of software interrupts, see **Section 6.3.10, SINTR0 to SINTR3 — Software Interrupt Register** and **Section 6.4.7, Priority Level Handling**.

6.4.6 DTS Interrupt Merge Function

Up to 128 transfer end interrupts and up to 128 transfer count match interrupts are aggregated into one type of interrupt in units of 32 interrupts. 128 bits from all DTS channels are combined into 8 registers, PINT0 to PINT7. If multiple DTS channels within the same status register (PINT0 to PINT7) request an interrupt, only the bit of the lowest DTS channel in that register is set. For additional information see **Section 6.3.11, PINTn + x — Peripheral Interrupt Status Register** and **Section 6.3.12, PINTCLRn + x — Peripheral Interrupt Status Clear Register**.

6.4.7 Priority Level Handling

There is no priority control for FENMI because it has one channel.

FE-level maskable interrupt priority is determined within FEINT0-1. The channel with the lowest number is given priority over other channels.

EI-level maskable interrupt priority is determined within EIINT0-31 and EIINT32-511 in each PE. Priority is determined by 16 interrupt priority levels specified for each interrupt channel (EICn.EIP3-0). In channels with same priority level, the channel with the lowest number is given priority over other channels.

6.5 Interrupt Processing Flow

6.5.1 Level Interrupt Processing Flow

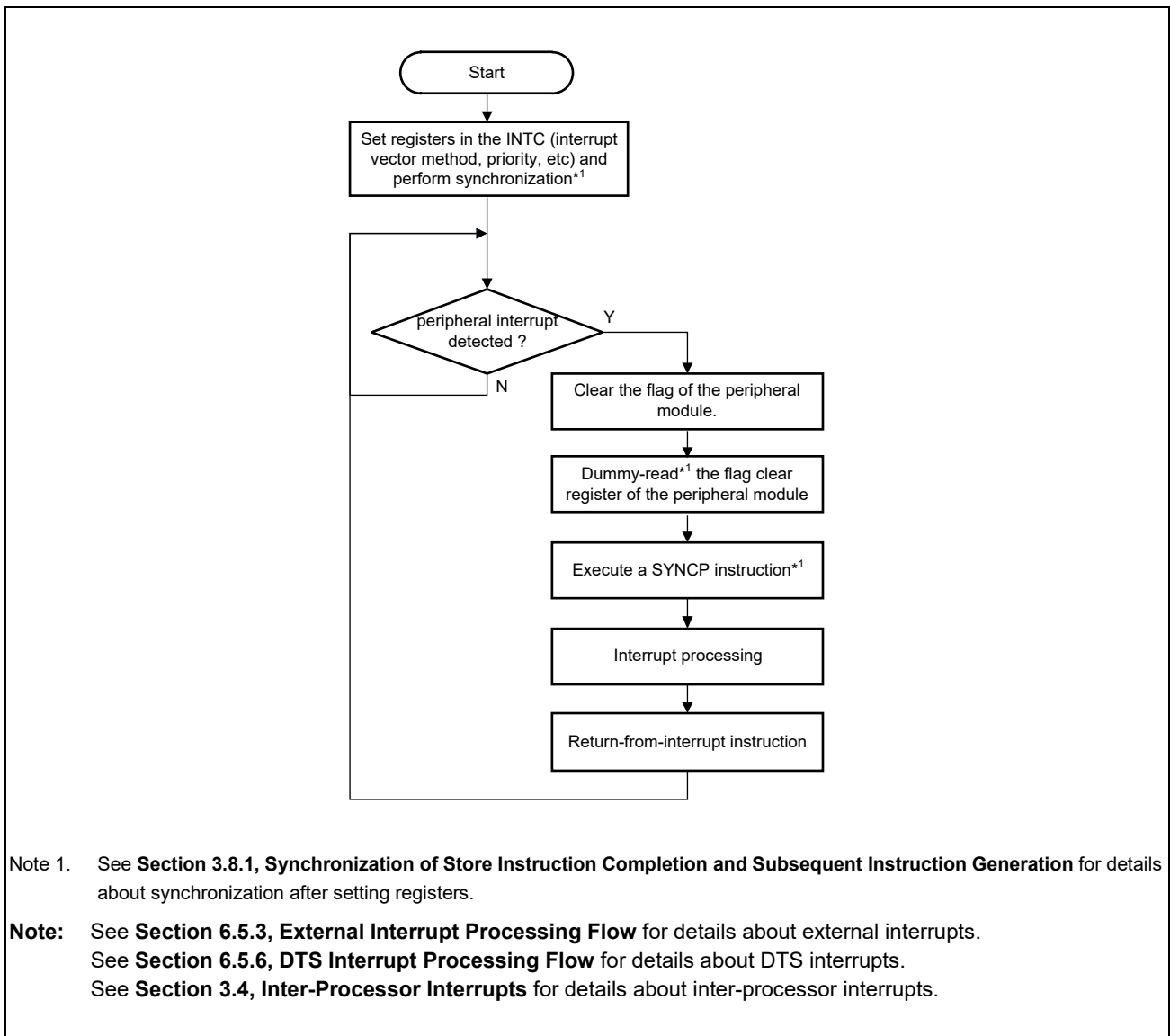


Figure 6.2 Example of the Level Interrupt Processing Flow

6.5.2 NMI Processing Flow

- Select the NMI detection method (falling edge or rising edge) by setting the NMICTL register.
- After an NMI has been detected, an interrupt request is sent to the INTC.
- NMI interrupts are acknowledged as the highest priority even when another FE level interrupts have been generated. NMI interrupts cannot be masked by the PSW.NP in the CPU system register. Recovery or return from an NMI interrupt is not possible.

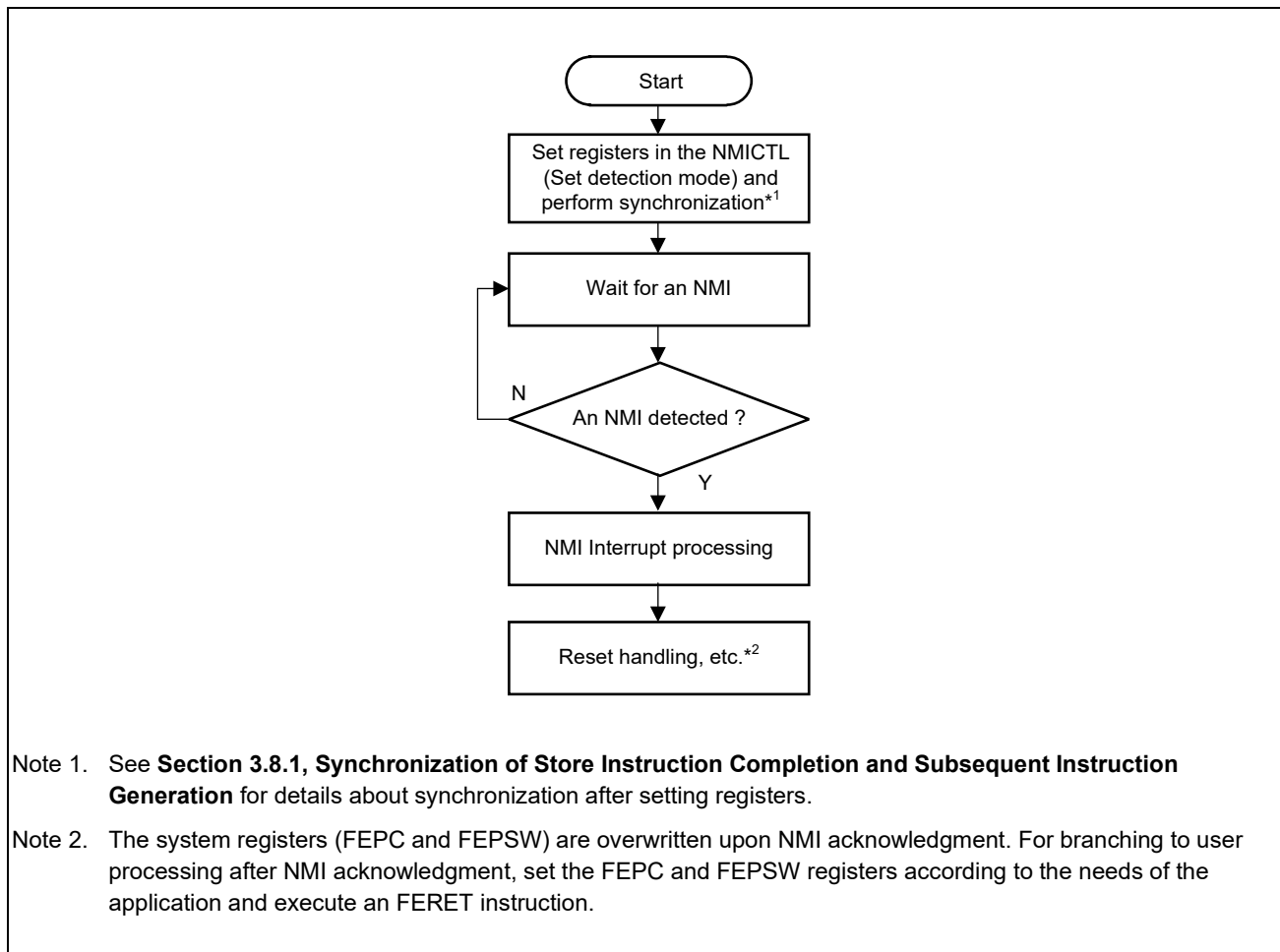


Figure 6.3 Example of the NMI Processing Flow

6.5.3 External Interrupt Processing Flow

- Select the IRQ detection method (edge detection or level detection) by setting the EXINTCTL register.
- After detection of an IRQ, an interrupt request is issued to the INTC.
- If level detection is selected, confirm that the IRQn pin is negated before returning from the interrupt service routine by the return instruction.
- If edge detection is selected, clear the interrupt request before returning from an interrupt service routine by the return instruction. To clear an interrupt request, write to the corresponding bit of the EXINTSTC register.

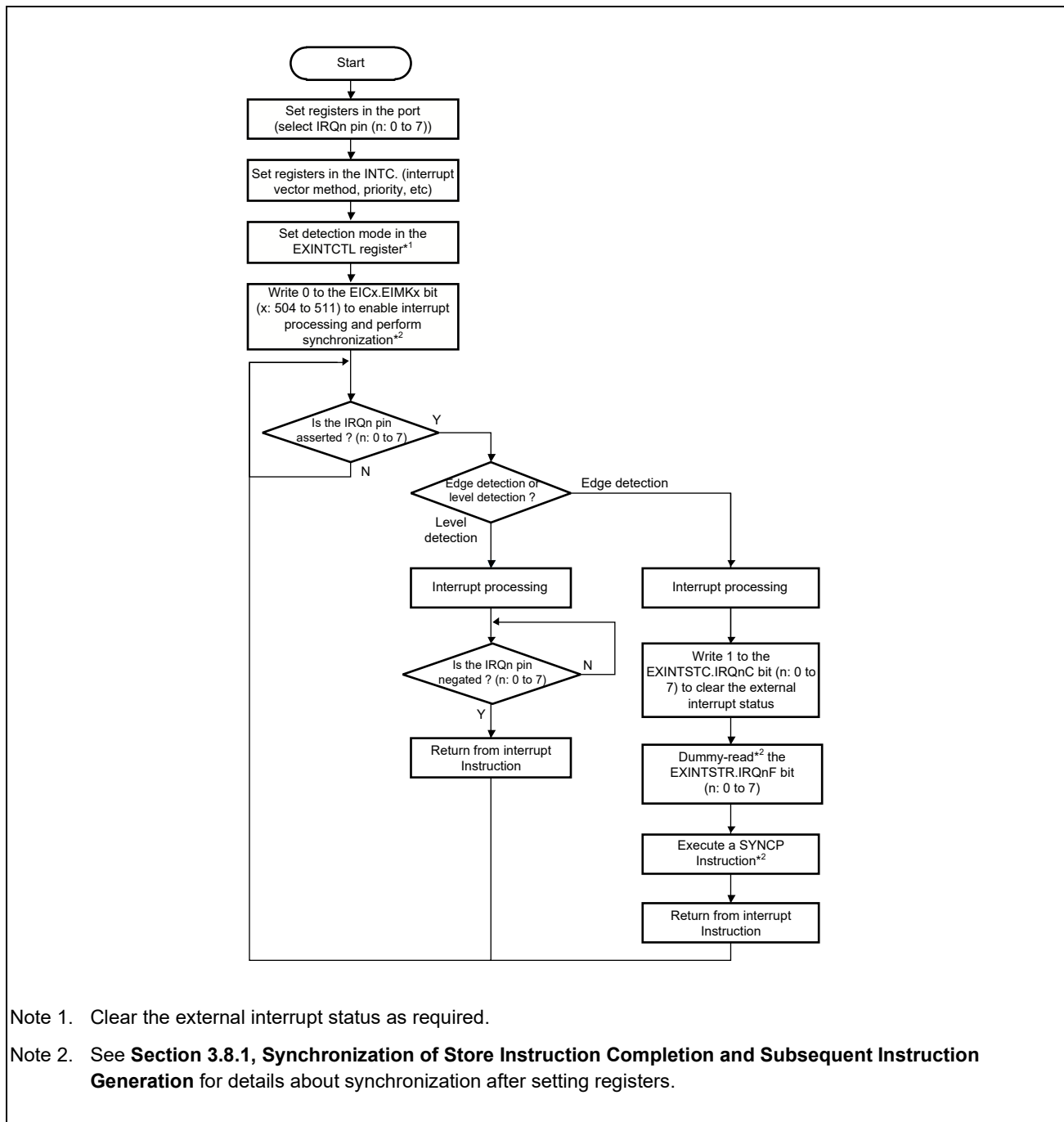


Figure 6.4 Example of the External Interrupt Processing Flow

6.5.4 Broadcast Interrupt Processing Flow

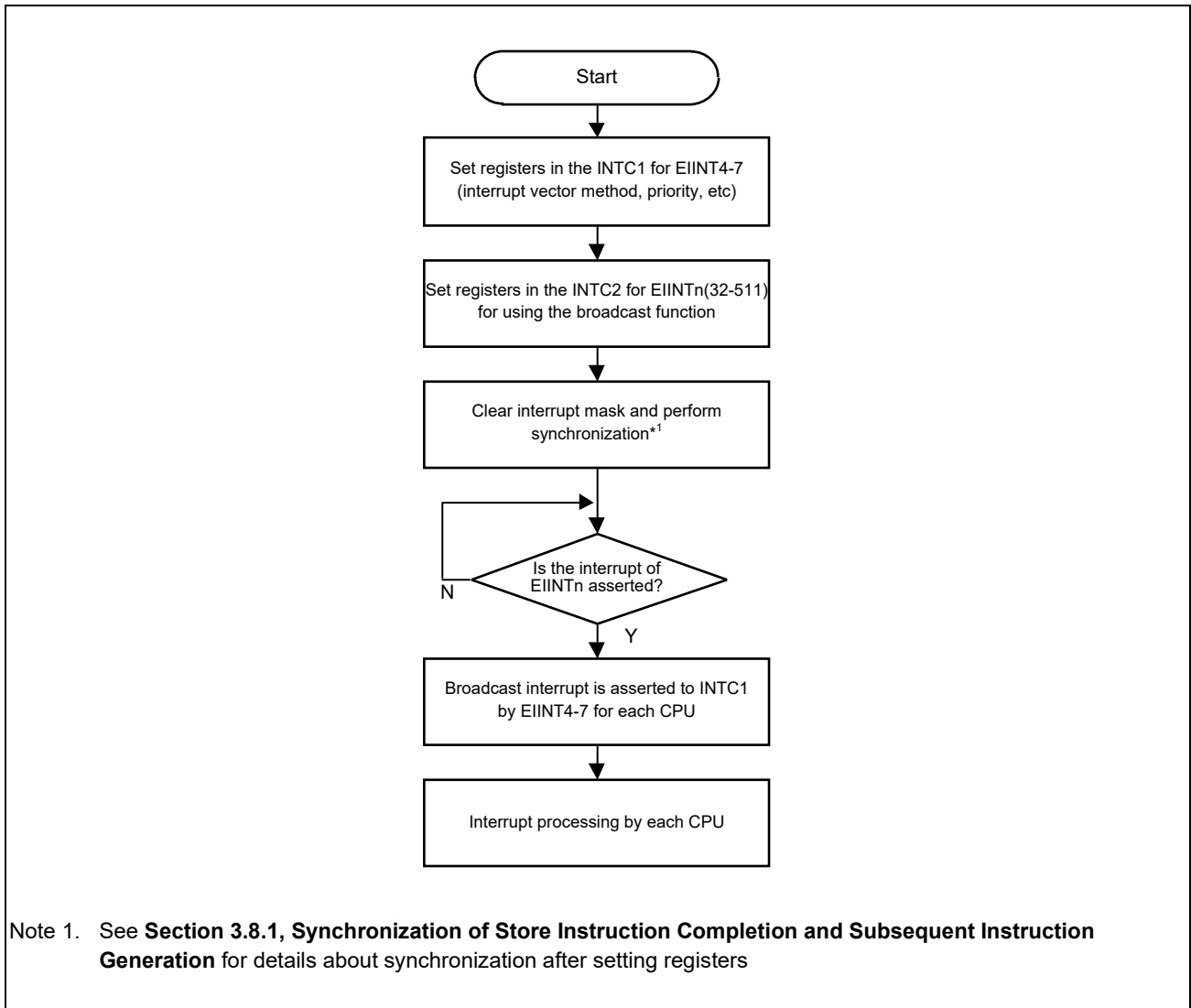


Figure 6.5 Example for the Broadcast Interrupt

NOTE

When using the broadcast function, the EIINTn (n = 32 to 511) interrupts are not generated individually upon the occurrence of each interrupt factor. Interrupts are generated in EIINT4-7 as broadcast interrupts.

6.5.5 Software Interrupt Processing Flow

- Software interrupt requests are controlled by writing 00_H or 01_H to the counter registers (SINTR0 to SINTR3).
- Writing 00_H leads to the counter's value being decremented by 1.
- Writing 01_H leads to the counter's value being incremented by 1.
- If the incremented counter value is 1 or above, an interrupt request for the INTC is generated.
- Before returning from the interrupt service routine, decrement the SINTR_n counter by 1. If SINTR_n is 00_H after issuing the instruction to return from the interrupt, wait for the writing of 01_H to SINTR_n.

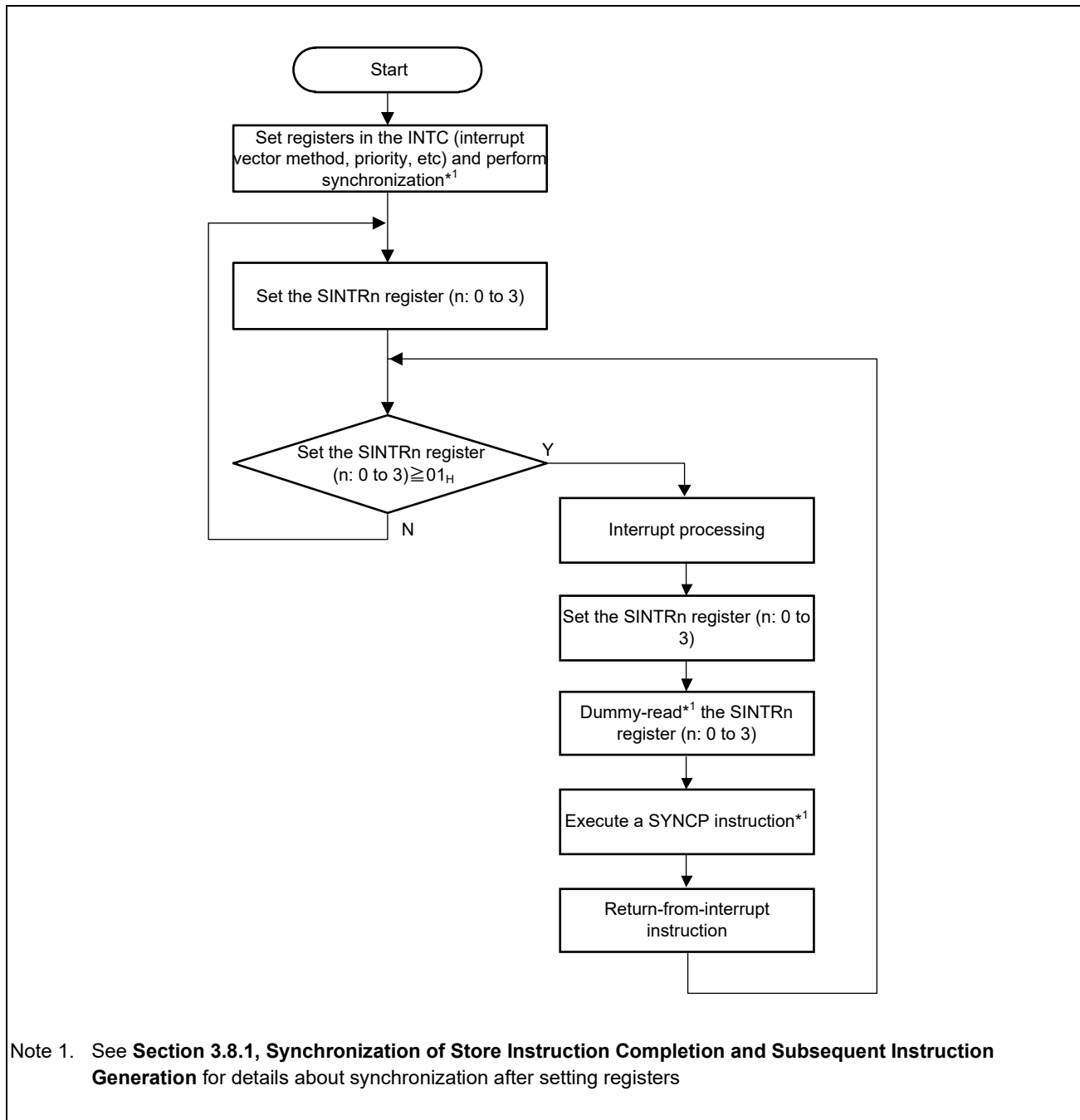


Figure 6.6 Example of the Software Interrupt Processing Flow

6.5.6 DTS Interrupt Processing Flow

- When only one interrupt request is generated out of the bundled 32 interrupt sources:
 - The bit corresponding to the interrupt request in the PINTn register is set to 1 and an interrupt request is output.
 - On completion of interrupt processing, write 1 to the interrupt clear register (PINTCLRn) to clear the interrupt request before issuing the return from interrupt instruction, then wait for the next one.
- When multiple interrupt sources are generated out of the bundled 32 interrupt sources:
 - Only the bit of the DTS channel with the lowest number is set in the PINTn register and the request is generated.
 - On completion of interrupt processing (before returning from the interrupt service routine), write 1 to the corresponding bit in the interrupt clear register (PINTCLRn) to clear the interrupt request.
 - After clearing the interrupt request bit of the first DTS channel, the bit corresponding to the next lowest DTS channel requesting an interrupt is set and the interrupt request is output.
 - These steps are repeated until all interrupt sources bundled into the same 32-bit register are cleared.

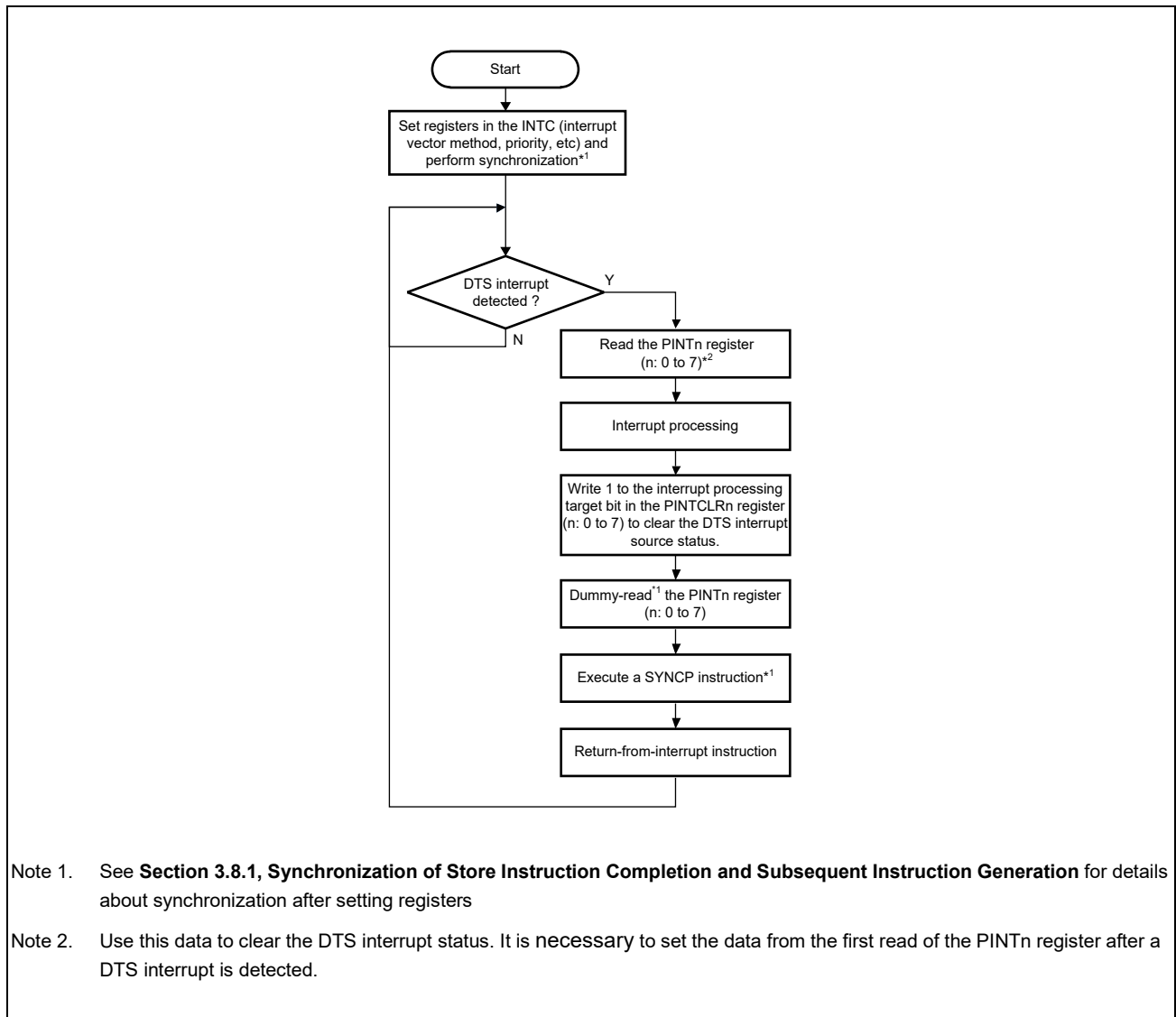


Figure 6.7 Example of the DTS Interrupt Processing Flow

6.6 Interrupt Response Times

Table 6.27 Interrupt Response Times (Min)

Target	Interrupt Request Source		Number of Cycles for Processing			
	INTC Connection	Operating Clock	Synchronization	INTC2	INTC1	In PE0/PE1
PE0/PE1	Directly input to INTC1	CLK_CPU	$1 \times I\phi$	—	$2 \times I\phi$ < $1 \times I\phi$ >	See the description under "In PE0/PE1" below.
		CLK_LSB	$1 \times P\phi$			
		CLK_LSB (Peripheral Group 2L)	$1 \times L\phi + 2 \times P\phi$			
		CLK_HSB	—			
		CLK_HBUS	$1 \times P\phi$			
		CLK_UHSB	—			
	Input via INTC2	CLK_CPU	—	$3 \times P\phi + 2 \times I\phi$ < $2 \times P\phi + 2 \times I\phi$ >	—	
		CLK_LSB	$1 \times P\phi$			
		CLK_LSB (Peripheral Group 2L)	$1 \times L\phi + 2 \times P\phi$			
		CLK_HSB	$1 \times H\phi + 2 \times P\phi$			
		CLK_HSB (Peripheral Group 2H)	$1 \times U\phi + 3 \times P\phi$			
		CLK_HBUS	$1 \times P\phi$			
		CLK_UHSB	$1 \times U\phi + 3 \times P\phi$			

Note: The numbers in < > indicate the number of cycles in the case of level detection.

Note: $I\phi$: CPU clock (CLK_CPU)

$P\phi$: HBUS clock (CLK_HBUS)

$H\phi$: Peripheral high speed clock (CLK_HSB)

$L\phi$: Peripheral low speed clock (CLK_LSB)

$U\phi$: Peripheral ultra high speed clock (CLK_UHSB)

Table 6.28 Response Cycles in CPU Operation(Min)

Interrupt Vector Method	Cache HIT/MISS	RBCR 0.MD	In PE0/PE1 400MHz	In PE0/PE1 240MHz
Direct Vector	Cache HIT	—	$5 \times I\phi$	$5 \times I\phi$
	Cache MISS	—	$11 \times I\phi$	$10 \times I\phi$
Table Reference (The vector code is stored in Code Flash)	Cache HIT	—	$14 \times I\phi$	$13 \times I\phi$
		0 _B	$19 \times I\phi$	$18 \times I\phi$
		1 _B	$25 \times I\phi$	$24 \times I\phi$
	Cache MISS	—	$22 \times I\phi$	$20 \times I\phi$
		0 _B	$22 \times I\phi$	$20 \times I\phi$
		1 _B	$25 \times I\phi$	$24 \times I\phi$

Note: $I\phi$: CPU clock (CLK_CPU)

Note: See Section 3.2.3.2(28), RBCR0 — Register Bank Control 0 for RBCR0.MD.

Section 7 sDMA Controller (sDMAC)

This section contains a generic description of the sDMA Controller (sDMAC).

The first part of this section describes this product's specific properties, such as the number of units and register base addresses.

The remainder of the section describes the functions and registers of the sDMAC.

7.1 Features of sDMAC

7.1.1 Number of Units

This microcontroller has the following number of sDMAC units.

Table 7.1 Number of Units

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Units	2	2
Number of Channels per Unit	16	16
Unit Name	SDMAC _j (j = 0, 1)	SDMAC _j (j = 0, 1)

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Units	2	2
Number of Channels per Unit	16	16
Unit Name	SDMAC _j (j = 0, 1)	SDMAC _j (j = 0, 1)

Table 7.2 Index

Index	Meaning
j	The individual units are identified by the index "j" (j = 0 to 1).
n	The individual channels are identified by the index "n" (n = 0 to 15).
m	The individual trigger group selection registers are identified by the index "m" (m = 0 to 15).

7.1.2 Register Base Addresses

sDMAC register addresses are given as offsets from the base addresses.

sDMAC base addresses are listed in the following table.

Table 7.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<SDMAC0_base>	FFF9 0000 _H	Peripheral Group 0
<SDMAC1_base>	FFF9 8000 _H	Peripheral Group 0
<DMATRGSSEL_base>	FF09 0100 _H	Peripheral Group 9

7.1.3 Clock Supply

The sDMAC clock supplies are shown in the following table.

Table 7.4 sDMAC Clock Supply

Unit Name	Clock Name in the Unit	Supply Clock Name	Description
SDMACj	SCLK	CLK_HBUS	System interconnect clock
	PCLK	CLK_HBUS	Register access clock
DMATRGSSEL	PCLK	CLK_HBUS	Register access clock

7.1.4 Interrupts

sDMAC interrupt requests are listed in the following table.

Table 7.5 Interrupt Requests (1/3)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number
INTSDMAC0CH0	SDMAC0.intreq_ch[0]	sDMAC0 channel 0 transfer end interrupt sDMAC0 channel 0 descriptor step end interrupt sDMAC0 channel 0 address error interrupt	45
INTSDMAC0CH1	SDMAC0.intreq_ch[1]	sDMAC0 channel 1 transfer end interrupt sDMAC0 channel 1 descriptor step end interrupt sDMAC0 channel 1 address error interrupt	46
INTSDMAC0CH2	SDMAC0.intreq_ch[2]	sDMAC0 channel 2 transfer end interrupt sDMAC0 channel 2 descriptor step end interrupt sDMAC0 channel 2 address error interrupt	47
INTSDMAC0CH3	SDMAC0.intreq_ch[3]	sDMAC0 channel 3 transfer end interrupt sDMAC0 channel 3 descriptor step end interrupt sDMAC0 channel 3 address error interrupt	48
INTSDMAC0CH4	SDMAC0.intreq_ch[4]	sDMAC0 channel 4 transfer end interrupt sDMAC0 channel 4 descriptor step end interrupt sDMAC0 channel 4 address error interrupt	49
INTSDMAC0CH5	SDMAC0.intreq_ch[5]	sDMAC0 channel 5 transfer end interrupt sDMAC0 channel 5 descriptor step end interrupt sDMAC0 channel 5 address error interrupt	50
INTSDMAC0CH6	SDMAC0.intreq_ch[6]	sDMAC0 channel 6 transfer end interrupt sDMAC0 channel 6 descriptor step end interrupt sDMAC0 channel 6 address error interrupt	51
INTSDMAC0CH7	SDMAC0.intreq_ch[7]	sDMAC0 channel 7 transfer end interrupt sDMAC0 channel 7 descriptor step end interrupt sDMAC0 channel 7 address error interrupt	52
INTSDMAC0CH8	SDMAC0.intreq_ch[8]	sDMAC0 channel 8 transfer end interrupt sDMAC0 channel 8 descriptor step end interrupt sDMAC0 channel 8 address error interrupt	53

Table 7.5 Interrupt Requests (2/3)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number
INTSDMAC0CH9	SDMAC0.intreq_ch[9]	sDMAC0 channel 9 transfer end interrupt sDMAC0 channel 9 descriptor step end interrupt sDMAC0 channel 9 address error interrupt	54
INTSDMAC0CH10	SDMAC0.intreq_ch[10]	sDMAC0 channel 10 transfer end interrupt sDMAC0 channel 10 descriptor step end interrupt sDMAC0 channel 10 address error interrupt	55
INTSDMAC0CH11	SDMAC0.intreq_ch[11]	sDMAC0 channel 11 transfer end interrupt sDMAC0 channel 11 descriptor step end interrupt sDMAC0 channel 11 address error interrupt	56
INTSDMAC0CH12	SDMAC0.intreq_ch[12]	sDMAC0 channel 12 transfer end interrupt sDMAC0 channel 12 descriptor step end interrupt sDMAC0 channel 12 address error interrupt	57
INTSDMAC0CH13	SDMAC0.intreq_ch[13]	sDMAC0 channel 13 transfer end interrupt sDMAC0 channel 13 descriptor step end interrupt sDMAC0 channel 13 address error interrupt	58
INTSDMAC0CH14	SDMAC0.intreq_ch[14]	sDMAC0 channel 14 transfer end interrupt sDMAC0 channel 14 descriptor step end interrupt sDMAC0 channel 14 address error interrupt	59
INTSDMAC0CH15	SDMAC0.intreq_ch[15]	sDMAC0 channel 15 transfer end interrupt sDMAC0 channel 15 descriptor step end interrupt sDMAC0 channel 15 address error interrupt	60
INTSDMACERR	SDMAC0.intreq_aerr SDMAC1.intreq_aerr	sDMAC0 address error interrupt sDMAC1 address error interrupt	29
INTSDMAC1CH0	SDMAC1.intreq_ch[0]	sDMAC1 channel 0 transfer end interrupt sDMAC1 channel 0 descriptor step end interrupt sDMAC1 channel 0 address error interrupt	61
INTSDMAC1CH1	SDMAC1.intreq_ch[1]	sDMAC1 channel 1 transfer end interrupt sDMAC1 channel 1 descriptor step end interrupt sDMAC1 channel 1 address error interrupt	62
INTSDMAC1CH2	SDMAC1.intreq_ch[2]	sDMAC1 channel 2 transfer end interrupt sDMAC1 channel 2 descriptor step end interrupt sDMAC1 channel 2 address error interrupt	63
INTSDMAC1CH3	SDMAC1.intreq_ch[3]	sDMAC1 channel 3 transfer end interrupt sDMAC1 channel 3 descriptor step end interrupt sDMAC1 channel 3 address error interrupt	64
INTSDMAC1CH4	SDMAC1.intreq_ch[4]	sDMAC1 channel 4 transfer end interrupt sDMAC1 channel 4 descriptor step end interrupt sDMAC1 channel 4 address error interrupt	65
INTSDMAC1CH5	SDMAC1.intreq_ch[5]	sDMAC1 channel 5 transfer end interrupt sDMAC1 channel 5 descriptor step end interrupt sDMAC1 channel 5 address error interrupt	66
INTSDMAC1CH6	SDMAC1.intreq_ch[6]	sDMAC1 channel 6 transfer end interrupt sDMAC1 channel 6 descriptor step end interrupt sDMAC1 channel 6 address error interrupt	67
INTSDMAC1CH7	SDMAC1.intreq_ch[7]	sDMAC1 channel 7 transfer end interrupt sDMAC1 channel 7 descriptor step end interrupt sDMAC1 channel 7 address error interrupt	68
INTSDMAC1CH8	SDMAC1.intreq_ch[8]	sDMAC1 channel 8 transfer end interrupt sDMAC1 channel 8 descriptor step end interrupt sDMAC1 channel 8 address error interrupt	69
INTSDMAC1CH9	SDMAC1.intreq_ch[9]	sDMAC1 channel 9 transfer end interrupt sDMAC1 channel 9 descriptor step end interrupt sDMAC1 channel 9 address error interrupt	70
INTSDMAC1CH10	SDMAC1.intreq_ch[10]	sDMAC1 channel 10 transfer end interrupt sDMAC1 channel 10 descriptor step end interrupt sDMAC1 channel 10 address error interrupt	71

Table 7.5 Interrupt Requests (3/3)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number
INTSDMAC1CH11	SDMAC1.intreq_ch[11]	sDMAC1 channel 11 transfer end interrupt sDMAC1 channel 11 descriptor step end interrupt sDMAC1 channel 11 address error interrupt	72
INTSDMAC1CH12	SDMAC1.intreq_ch[12]	sDMAC1 channel 12 transfer end interrupt sDMAC1 channel 12 descriptor step end interrupt sDMAC1 channel 12 address error interrupt	73
INTSDMAC1CH13	SDMAC1.intreq_ch[13]	sDMAC1 channel 13 transfer end interrupt sDMAC1 channel 13 descriptor step end interrupt sDMAC1 channel 13 address error interrupt	74
INTSDMAC1CH14	SDMAC1.intreq_ch[14]	sDMAC1 channel 14 transfer end interrupt sDMAC1 channel 14 descriptor step end interrupt sDMAC1 channel 14 address error interrupt	75
INTSDMAC1CH15	SDMAC1.intreq_ch[15]	sDMAC1 channel 15 transfer end interrupt sDMAC1 channel 15 descriptor step end interrupt sDMAC1 channel 15 address error interrupt	76

7.1.5 sDMAC Transfer Requests

The sDMAC transfer request table lists all sDMAC transfer request factors that can be selected by the DMAjRS_n registers and groups that can be selected by the DMACSELj_m registers.

For details about the sDMAC transfer request table lists, refer to Appendix file “sDMAC_Transfer_request_Table.xlsx”.

7.1.6 Reset Sources

sDMAC reset sources are listed in the following table. The sDMAC is initialized by these reset sources.

Table 7.6 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
SDMACj	All registers	✓	✓	✓	✓	—	—
DMATRGSSEL	All registers	✓	✓	✓	✓	—	—

7.2 Overview

The sDMAC can be used in place of the CPU to perform high-speed data transfer between on-chip memories.

7.2.1 Functional Overview

The sDMAC has the following features:

- 16 channels
- 4-Gbyte physical address space
 - Source and destination addresses are 32 bits wide
- Connected to the System Bus directly. If a conflict occurs with another bus master in the System Bus, the transfer is arbitrated by round-robin arbitration.
- Physical bus data width: 64 bits
- Transfer data length: 1 byte, 2 bytes, 4 bytes, 8 bytes, 16 bytes, 32 bytes, 64 bytes
- The maximum transfer size: 4,294,967,295 bytes ($2^{32}-1$ bytes)
- Parallel reads and writes (fly-by)
- Address mode: Dual address mode
- Transfer requests:

Two request modes are available. Requests can be triggered by an auto request (a transfer request always occurs) or a peripheral hardware request.
- Bus mode:

Each channel can be set to either normal speed mode or slow speed mode.
- Mode for arbitration between transfer channels:

The arbitration mode can be set to either fixed priority arbitration mode or round-robin arbitration mode.
- Interrupt requests:

CPU interrupts from the sDMAC are triggered by three factors: termination of the descriptor step, termination of a data transfer and occurrence of an address error.
- Functions of the descriptor:

A descriptor contains register settings (source address register, destination address register, transfer size register, transfer mode register and scatter gather registers) and can be chained to the next descriptor. The descriptor memory size is 4 KB. This descriptor memory is shared by all channels.
- Scatter-gather transfer:

Inner and outer loops for transfers with inner and outer address increments are supported. Scattering at the destination interface and gathering at the source interface can be enabled independently and in parallel.

 - Inner and outer address increments are 32 bits wide
- sDMAC has a data RAM as a pending transfer data buffer.

The data RAM size is 2 KB and a DMA channel can use 64 bits × 16 lines for its own transactions.

7.2.2 Block Diagram

The following is a block diagram of sDMAC.

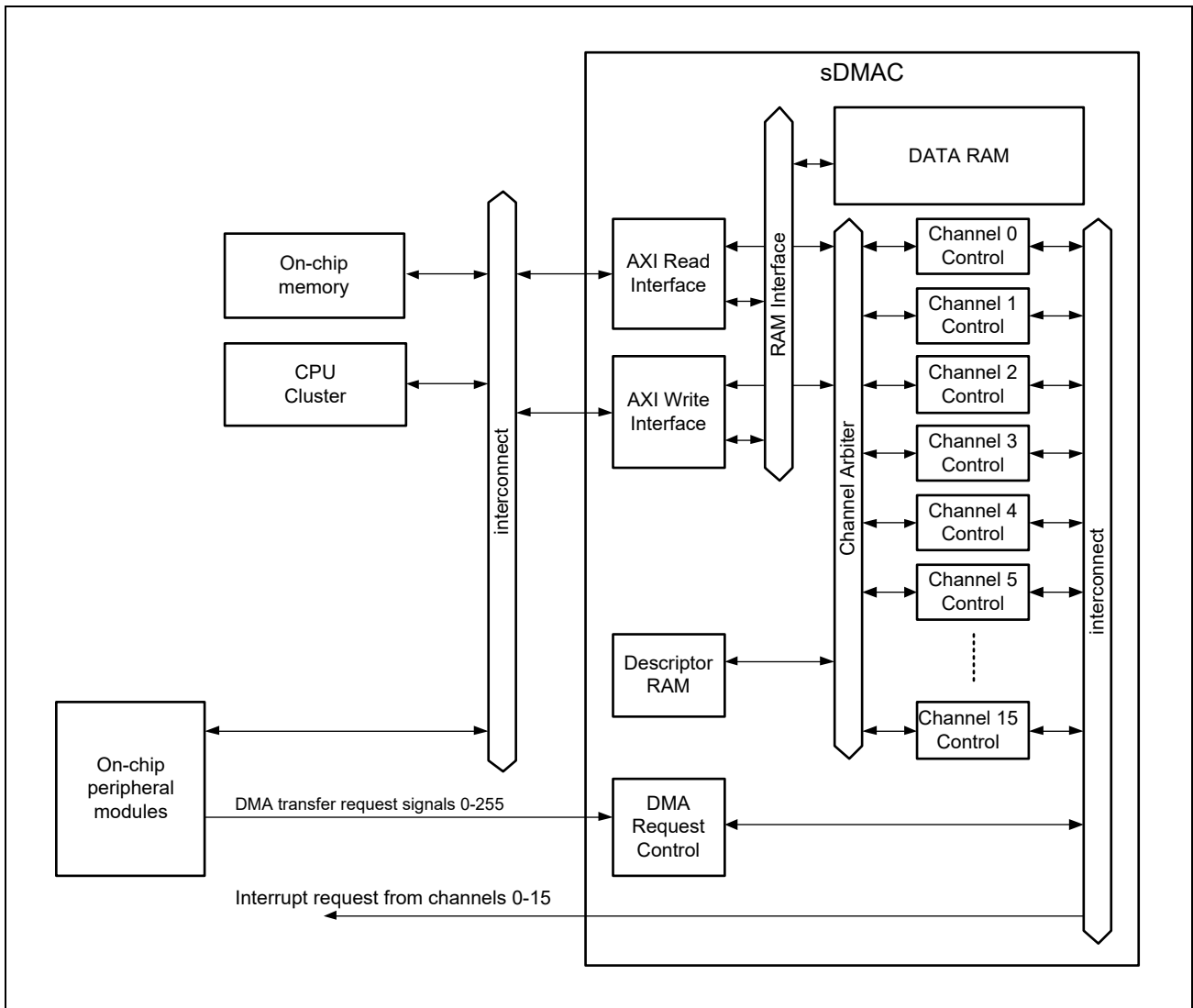


Figure 7.1 Block Diagram of sDMAC

The following block diagram shows sDMAC transfer request source selection by the DMACSELj_m registers.

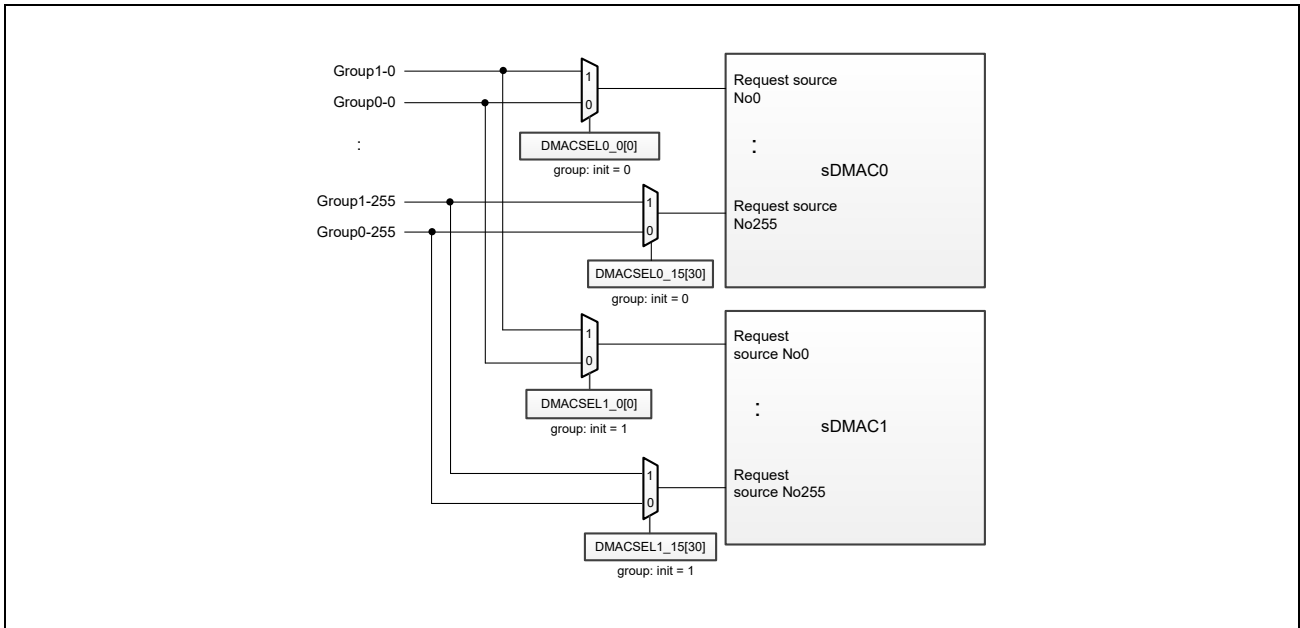


Figure 7.2 Block Diagram of sDMAC Transfer Request Selection

7.3 Registers of sDMAC

7.3.1 List of Registers

sDMAC registers are listed in the following table.

For details about <SDMACj_base>, see **Section 7.1.2, Register Base Address**.

Table 7.7 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection
Global Registers					
SDMACj	DMA Address Error Notification Status Register	DMAjESTA	<SDMACj_base> + 0010 _H	32	—
	DMA Interrupt Status Register	DMAjISTA	<SDMACj_base> + 0020 _H	32	—
	DMA Channel Request Priority Register	DMAjCHPRI	<SDMACj_base> + 0040 _H	32	—
	DMA Operation Register	DMAjOR	<SDMACj_base> + 0060 _H	16	—
	DMA Channel Reset Register	DMAjCHRST	<SDMACj_base> + 0080 _H	32	—
	DMA Channel Master Setting Register	DMAjCM_n	<SDMACj_base> + (n × 4 _H) + 100 _H	32	—
Channel Registers					
SDMACj	DMA Source Address Register	DMAjSAR_n	<SDMACj_base> + (n × 80 _H) + 2000 _H	32	—
	DMA Destination Address Register	DMAjDAR_n	<SDMACj_base> + (n × 80 _H) + 2004 _H	32	—
	DMA Transfer Size Register	DMAjTSR_n	<SDMACj_base> + (n × 80 _H) + 2008 _H	32	—
	DMA Transfer Size Register B	DMAjTSRB_n	<SDMACj_base> + (n × 80 _H) + 200C _H	32	—
	DMA Transfer Mode Register	DMAjTMR_n	<SDMACj_base> + (n × 80 _H) + 2010 _H	32	—
	DMA Channel Control Register	DMAjCHCR_n	<SDMACj_base> + (n × 80 _H) + 2014 _H	16	—
	DMA Channel Stop Register	DMAjCHSTP_n	<SDMACj_base> + (n × 80 _H) + 2016 _H	16	—
	DMA Channel Status Register	DMAjCHSTA_n	<SDMACj_base> + (n × 80 _H) + 2018 _H	32	—
	DMA Channel Flag Clear Register	DMAjCHFCR_n	<SDMACj_base> + (n × 80 _H) + 201C _H	32	—
	DMA Gather Inner Address Increment Register	DMAjGIAI_n	<SDMACj_base> + (n × 80 _H) + 2020 _H	32	—
	DMA Gather Outer Address Increment Register	DMAjGOAI_n	<SDMACj_base> + (n × 80 _H) + 2024 _H	32	—
	DMA Scatter Inner Address Increment Register	DMAjSIAI_n	<SDMACj_base> + (n × 80 _H) + 2028 _H	32	—
	DMA Scatter Outer Address Increment Register	DMAjSOAI_n	<SDMACj_base> + (n × 80 _H) + 202C _H	32	—
	DMA Scatter Gather Status Register	DMAjSGST_n	<SDMACj_base> + (n × 80 _H) + 2038 _H	32	—
	DMA Scatter Gather Control Register	DMAjSGCR_n	<SDMACj_base> + (n × 80 _H) + 203C _H	32	—
	DMA Resource Select Register	DMAjRS_n	<SDMACj_base> + (n × 80 _H) + 2040 _H	32	—
	DMA Buffer Control Register	DMAjBUFCR_n	<SDMACj_base> + (n × 80 _H) + 2048 _H	32	—
	DMA Descriptor Pointer Register	DMAjDPPTR_n	<SDMACj_base> + (n × 80 _H) + 2050 _H	32	—
DMA Descriptor Control Register	DMAjDPCR_n	<SDMACj_base> + (n × 80 _H) + 2054 _H	32	—	
Descriptor memory					
SDMACj	Descriptor Memory	Descriptor RAM	<SDMACj_base> + 4000 _H to <SDMACj_base> + 4FFC _H	32	—

Table 7.7 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection
Transfer trigger selection					
DMATRGSEL	sDMACj Transfer Request Group Selection Register m	DMACSELj_m	<DMATRGSEL_base> + (j × 40 _H) + (m × 4 _H)	8, 16, 32	—

7.3.2 DMAjESTA — DMA Address Error Notification Status Register

DMAjESTA is a 32-bit read-only register that indicates the address error notification (INTSDMACERR) status of each channel. Causes of address errors are shown in **Section 7.5(1), Address Errors**. Specification of the error notification routes is shown in **Section 7.4.2, DMA Interrupts**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEN[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7.8 DMAjESTA Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When read, the value after reset is returned.
15 to 0	AEN[15:0]	Channel address error notification INTSDMACERR status These bits reflect the CAE bit status of the channels whose CAEE bit is set to 1. AEN[n]: 0: Channel n address error notification does not occur 1: Channel n address error notification occurs

7.3.3 DMAjISTA — DMA Interrupt Status Register

DMAjISTA is a 32-bit read-only register that indicates the interrupt signal status of each channel. Interrupt routes are shown in **Section 7.4.2, DMA Interrupts**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7.9 DMAjISTA Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When read, the value after reset is returned.
15 to 0	INT[15:0]	Channel interrupt status INT[n]: 0: Channel n interrupt does not occur 1: Channel n Interrupt occurs

7.3.4 DMAjCHPRI — DMA Channel Request Priority Register

DMAjCHPRI is a 32-bit read/write register that specifies the channel request priority count timing.

See **Section 7.4.3(3), Channel Request Priority** for details.

Value after reset: 0000 0010_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PRICNT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.10 DMAjCHPRI Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PRICNT[15:0]	Channel Request Priority Count value This bit is used to control the timing of the channel request priority function.

7.3.5 DMAjOR — DMA Operation Register

DMAjOR is a 16-bit read/write register that controls master enable and specifies the priority level of all DMA channels.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PR[1:0]		—	—	—	—	—	—	—	DME
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 7.11 DMAjOR Register Contents

Bit Position	Bit Name	Function
15 to 10	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
9, 8	PR[1:0]	Priority Mode Select the priority level between channels when there are transfer requests for multiple channels simultaneously. 00: CH0 > CH1 > ... > CH14 > CH15 11: Round-robin mode Other than above: Setting prohibited
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	DME	DMA Master Enable Enables or disables DMA transfers on all channels. If the DME bit and the DE bit in DMAjCHCR_n are set to 1, DMA transfer is enabled. At this time, the TE bits in DMAjCHSTA_n and the CAE bits in DMAjCHSTA_n must be all 0. If this bit is cleared during transfer, transfers on all channels are terminated. 0: Disable DMA transfers on all channels 1: Enable DMA transfers on all channels

7.3.6 DMAjCHRST — DMA Channel Reset Register

DMAjCHRST register is a 32-bit write-only register that initializes each channel.

When this register is set, the specified channel state is initialized completely and the following registers are initialized.

DMAjSAR_n, DMAjDAR_n, DMAjTSR_n, DMAjTSRB_n, DMAjTMR_n, DMAjCHCR_n, DMAjCHSTP_n, DMAjCHSTA_n, DMAjGIAI_n, DMAjGOAI_n, DMAjSIAI_n, DMAjSOAI_n, DMAjSGCR_n, DMAjRS_n, DMAjBUFCR_n, DMAjDPPTR_n, DMAjDPCR_n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 7.12 DMAjCHRST Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing, write the value after reset.
15 to 0	CLR[15:0]	All registers in the channel can be cleared by writing to these bits. CLR[0] 0: Ignored 1: Channel 0 registers are cleared CLR[1] 0: Ignored 1: Channel 1 registers are cleared ... CLR[15] 0: Ignored 1: Channel 15 registers are cleared Before writing to this register, confirm that channel BUSY is 0.

7.3.7 DMAjCM_n — DMA Channel Master Setting Register

DMAjCM_n is a 32-bit read/write register that configures the DMA master information.

Value after reset: DMA0CM_n: 0000 1C00H

DMA1CM_n: 0000 1B00H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SPID[4:0]				—	—	—	—	—	—	—	—	UM
Value after reset	0	0	0	*	*	*	*	*	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Note: This value is 1C_H (sDMAC0) or 1B_H (sDMAC1).

Table 7.13 DMAjCM_n Register Contents

Bit Position	Bit Name	Function
31 to 13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	SPID	Channel master SPID setting Specifies the SPID information of the master assigned to the channel. Value after reset of DMA0CM_n: 1C _H Value after reset of DMA1CM_n: 1B _H
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	UM	Channel master UM setting Specifies the UM information of the master assigned to the channel. 0: Supervisor mode 1: User mode

7.3.8 DMAjSAR_n — DMA Source Address Register

DMAjSAR_n is a 32-bit read/write register that specifies the source address of DMA transfer. During DMA transfer, this register indicates the next source address.

When the selected source address mode is “increment” (SM bit in DMAjTMR_n register is 01_B), the source address can be specified on a byte boundary. Otherwise the source address must be specified on the transaction size boundary (STS bit in DMAjTMR_n register).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAR[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.3.9 DMAjDAR_n — DMA Destination Address Register

DMAjDAR_n is a 32-bit read/write register that specifies the destination address of DMA transfer. During DMA transfer, this register indicates the next destination address.

When the selected destination address mode is “increment” (DM bit in DMAjTMR_n register is 01_B), the destination address can be specified on a byte boundary. Otherwise the destination address must be specified on the transaction size boundary (DTS bit in DMAjTMR_n register).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAR[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.3.10 DMAjTSR_n — DMA Transfer Size Register

DMAjTSR_n is a 32-bit read/write register that specifies the DMA transfer size. The total size of DMA transfer is 1 byte when the setting is 0000 0001_H and 4,294,967,295 bytes ($2^{32}-1$ bytes) when FFFF FFFF_H is set. When the setting is 0000 0000_H, transfer does not occur, and TE is set immediately. During DMA transfer, DMAjTSR_n indicates the remaining transfer size.

In DMA transfer, the size of the read transfer and write transfer may differ. This is because the DMAC uses flyby DMA transfer via the data RAM. DMAjTSR_n indicates the read transfer size.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSR[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSR[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.3.11 DMAjTSRB_n — DMA Transfer Size Register B

DMAjTSRB_n is a 32-bit read-only register that indicates the DMA transfer size. During DMA transfer, this register indicates the remaining transfer size (in bytes). When DMAjTSR_n is written, the same value is set in this register.

In DMA transfer, the size of the read transfer and write transfer may differ. This is because the DMAC uses flyby DMA transfer via data RAM. DMAjTSRB_n indicates the write transfer size. The DMAjTSR_n value is copied to the DMAjTSRB_n register when DMA transfer starts.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSRW[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSRW[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

7.3.12 DMAjTMR_n — DMA Transfer Mode Register

DMAjTMR_n is a 32-bit read/write register that specifies the DMA transfer mode.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SLM[3:0]			PRI[3:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	TRS	DM[1:0]		SM[1:0]		DTS[3:0]			STS[3:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.14 DMAjTMR_n Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
23 to 20	SLM[3:0]	DMA Transfer Slow Speed mode Specifies the number of clock cycles (bus clock) between two DMA transfers. DMA transfer is executed once every number of clock cycles specified by SLM. 0000: Normal mode 1000: 256 clock cycles 1001: 512 clock cycles 1010: 1024 clock cycles : 1111: 32768 clock cycles Other than above: Setting prohibited
19 to 16	PRI[3:0]	Channel Request Priority Setting These bits set the priority of channel requests. Refer to 7.4.3(3) for details. PRI[3] 0: Channel request priority is disabled. 1: Channel request priority is enabled. PRI[2:0] 111: Highest priority : 000: Lowest priority
15 to 13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12	TRS	Transfer request source Specifies which transfer request source will be sent to the DMAC. Before changing transfer request sources, the DMA enable bit (DE) should be set to 0. 0: Auto request 1: Hardware request

Table 7.14 DMAjTMR_n Register Contents (2/2)

Bit Position	Bit Name	Function
11, 10	DM[1:0]	Destination address mode Specify whether the DMA destination address is incremented or fixed, and the amount of increment. 00: Fixed destination address 01: Destination address is incremented based on destination transaction size (DTS) +1 in 1-byte unit transfers +2 in 2-byte unit transfers +4 in 4-byte unit transfers +8 in 8-byte unit transfers +16 in 16-byte unit transfers +32 in 32-byte unit transfers +64 in 64-byte unit transfers Other than above: Setting prohibited
9, 8	SM[1:0]	Source Address Mode Specify whether the DMA source address is incremented or fixed, and the amount of increment. 00: Fixed source address 01: Source address is incremented based on source transaction size (STS) +1 in 1-byte unit transfers +2 in 2-byte unit transfers +4 in 4-byte unit transfers +8 in 8-byte unit transfers +16 in 16-byte unit transfers +32 in 32-byte unit transfers +64 in 64-byte unit transfers Other than above: Setting prohibited
7 to 4	DTS[3:0]	DMA destination transaction size 0000: 1-byte unit transfer 0001: 2-byte unit transfer 0010: 4-byte unit transfer 0011: 8-byte unit transfer 0100: 16-byte unit transfer 0101: 32-byte unit transfer 0110: 64-byte unit transfer Other than above: Setting prohibited
3 to 0	STS[3:0]	DMA source transaction size 0000: 1-byte unit transfer 0001: 2-byte unit transfer 0010: 4-byte unit transfer 0011: 8-byte unit transfer 0100: 16-byte unit transfer 0101: 32-byte unit transfer 0110: 64-byte unit transfer Other than above: Setting prohibited

7.3.13 DMAjCHCR_n — DMA Channel Control Register

DMAjCHCR_n is a 16-bit read/write register that controls DMA transfer.

Value after reset: 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DPE	DPB	—	—	—	CAEE	CAIE	DSIE	IE	DE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 7.15 DMAjCHCR_n Register Contents (1/2)

Bit Position	Bit Name	Function
15 to 10	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
9	DPE	Descriptor enable bit Specifies whether the descriptor is enabled. 0: Disabled Descriptor operation is disabled. 1: Enabled Descriptor operation is enabled. When the DMAjDPPTR_n.CF bit is set to 1, after reading the register setting from the descriptor memory, DMA transfer is continued. Clearing the DPE bit by setting the DMAjCHFCR_n.DPEC bit to 1 terminates the descriptor chain operation.
8	DPB	Descriptor start bit Specifies the configuration to be loaded at the beginning of the descriptor. After reading the descriptor, this bit is cleared. While DE = 1, DPB must not be set to 1. 0: Start the DMA transfer via a register setting. 1: Start DMA transfer after the channel configuration is copied from the descriptor memory. The configurations to copy from the Descriptor memory are specified in the DMAjDPCR_n register.
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	CAEE	Channel address error notification INTSDMACERR enable Specifies whether a channel can issue a notification upon the occurrence of an address error. When the CAEE bit and the DMAjCHSTA_n.CAE bit are set to 1, a channel (sDMAC_CH0 to sDMAC_CH15) can issue notifications about address errors. (INTSDMACERR) For details, see Section 7.4.2, DMA Interrupts . 0: Address error notification disabled 1: Address error notification enabled
3	CAIE	Channel address error interrupt INTSDMACjCHn enable Specifies whether a channel can generate an interrupt request upon occurrence of an address error. When the CAIE bit and DMAjCHSTA_n.CAE bit are set to 1, a channel will generate an interrupt request (INTSDMACjCHn) upon the occurrence of an address error. For details, see Section 7.4.2, DMA Interrupts . 0: Interrupt request disabled 1: Interrupt request enabled
2	DSIE	Descriptor step end interrupt master enable Specifies whether a channel can generate an interrupt at the end of a descriptor step. When the DMAjDPPTR_n.DIE bit and DSIE bit are set to 1, a channel will generate an interrupt request (INTSDMACjCHn) upon completion of a descriptor step. See Section 7.5, Usage Notes for details about DMA transfer interrupts. 0: Interrupt request is disabled 1: Interrupt request is enabled

Table 7.15 DMAjCHCR_n Register Contents (2/2)

Bit Position	Bit Name	Function
1	IE	<p>Transfer end interrupt enable</p> <p>Specifies whether or not a channel can generate an interrupt at the end of DMA transfer. Setting this bit to 1 will allow a channel to generate an interrupt request (INTSDMACjCHn) when the DMAjCHSTA_n.TE bit is set to 1. See Section 7.5, Usage Notes for details about DMA transfer interrupts.</p> <p>0: Interrupt request is disabled. 1: Interrupt request is enabled.</p>
0	DE	<p>DMA Enable</p> <p>Enables or disables DMA transfer for the corresponding channel. Clearing the DE bit to 0 by setting the DMAjCHFCR_n.DEC bit to 1 terminates DMA transfer. The DE bit is also cleared automatically when an ongoing DMA transfer is finished or an ECC error of the descriptor RAM occurs. See Section 7.5, Usage Notes for details about DMA transfer interrupts.</p> <p>0: DMA transfer disabled 1: DMA transfer enabled</p>

7.3.14 DMAjCHSTP_n — DMA Channel Stop Register

DMAjCHSTP_n is a 16-bit read/write register that controls the stop condition for DMA transfer.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 7.16 DMAjCHSTP_n Register Contents

Bit Position	Bit Name	Function
15 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	STP	DMA transmission stop When this bit is set to 1, packet requests stop immediately. When this bit is set to 0 after transmission has been stopped, the stopped transmission is resumed. 0: No operation 1: DMA transfer stop

7.3.15 DMAjCHSTA_n — DMA Channel Status Register

DMAjCHSTA_n register is a 32-bit read-only register that shows the DMA transfer status.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	OVF	DRQ	—	—	—	—	—	—	—	—	CAE	DSE	TE	BUSY
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7.17 DMAjCHSTA_n Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 14	—	Reserved When read, the value after reset is returned.
13	OVF	Hardware transfer request overflow flag Indicates that a hardware trigger was asserted before a pending hardware transfer request has been acknowledged. To clear the OVF bit, write 1 to the DMAjCHFCR_n.OVFC bit. 0: No hardware transfer request overflow 1: Hardware transfer request overflow occurred
12	DRQ	Hardware request status If this bit is set, it means that a hardware transfer request exists or is retained. <ul style="list-style-type: none"> This bit shows whether a hardware transfer request is retained. When the transfer specified by DMAjRS_n.TL bits is completed, this bit is automatically cleared. This bit can be cleared by writing 1 to the DMAjCHFCR_n.DRQC bit. This bit changes regardless of the value of the DMAjTMR_n.TRS bit when a hardware DMA transfer request is generated. 0: There is no hardware transfer request 1: There is a hardware transfer request
11 to 4	—	Reserved When read, the value after reset is returned.

Table 7.17 DMAjCHSTA_n Register Contents (2/2)

Bit Position	Bit Name	Function
3	CAE	<p>Address error flag</p> <p>Indicates that an address error interrupt occurred in the DMA transfer. The following conditions can set this bit:</p> <ul style="list-style-type: none"> The transfer source or transfer destination is an invalid space. A GUARD error occurred upon access to a source or a destination address. Channel register setting is prohibited. <p>See section 7.5(1), Address Errors for detail about address errors.</p> <p>If this bit is set, DMA transfer is not enabled for the channel even if the DE bit is set to 1. To clear the CAE bit, write 1 to the DMAjCHFCR_n.CAEC bit.</p> <ul style="list-style-type: none"> 0: Address error interrupt did not occur in the DMA transfer 1: Address error interrupt occurred in the DMA transfer
2	DSE	<p>Descriptor step end flag</p> <p>When the DMAjCHCR_n.DSIE bit is set to 1 and the DMAjDPPTR_n.DIE bit is set to 1, the DSE bit is set to 1 at the end of DMA transfer. To clear the DSE bit, write 1 to the DMAjCHFCR_n.DSEC bit.</p> <ul style="list-style-type: none"> 0: DMA transfer is in progress or has finished 1: The descriptor step has finished
1	TE	<p>Transfer end flag</p> <p>The TE bit is set to 1 when data transfer ends (DMAjTSR_n becomes 0). When the descriptor is enabled, the TE bit is set to 1 at the end of all descriptor steps.</p> <p>The TE bit is not set to 1 in the following cases:</p> <ul style="list-style-type: none"> DMA transfer ends due to a DMA address error before DMAjTSR_n becomes 0. DMA transfer is terminated by clearing the DE bit and DME bit in DMAjOR. <p>To clear the TE bit, write 1 to the DMAjCHFCR_n.TEC bit. If the DMAjCHCR_n.DE bit is set to 1 while TE bit is set to 1, DMA transfer will not be enabled.</p> <ul style="list-style-type: none"> 0: DMA transfer is in progress or DMA transfer has finished 1: DMA transfer ended at the specified count (TSR = 0)
0	BUSY	<p>Channel Busy flag</p> <p>This bit indicates that the channel is busy. When transfer is started, the BUSY bit is set to 1. After the DMAjCHCR_n.DE bit is cleared or the DMAjCHSTP_n.STP bit is set, the BUSY bit is cleared after receiving the response of all packets.</p> <ul style="list-style-type: none"> 0: Channel status is idle. 1: Channel status is busy.

7.3.16 DMAjCHFCR_n — DMA Channel Flag Clear Register

DMAjCHFCR_n is a 32-bit write-only register that clears the DMA channel status flags.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	OVFC	DRQC	—	—	DPEC	—	—	—	—	—	CAEC	DSEC	TEC	DEC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	W	W	R	R	W	R	R	R	R	R	W	W	W	W

Table 7.18 DMAjCHFCR_n Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 14	—	Reserved When writing, write the value after reset.
13	OVFC	Hardware transfer request overflow flag clear When the OVFC bit is set to 1, the DMAjCHSTA_n.OVF bit is cleared. 0: No operation 1: DMAjCHSTA_n.OVF bit is cleared.
12	DRQC	Hardware transfer request clear When the DRQC bit is set to 1, the DMAjCHSTA_n.DRQ bit is cleared. 0: No operation 1: DMAjCHSTA_n.DRQ bit is cleared.
11 to 10	—	Reserved When writing, write the value after reset.
9	DPEC	Descriptor enable clear When the DPEC bit is set to 1, the DMAjCHCR_n.DPE bit is cleared. The DMAjCHCR_n.DPE bit is the enable condition of the descriptor operation. To suspend DMA transfer at the end of a descriptor step, clear the DPE bit by using this bit. 0: No operation 1: DMAjCHCR_n.DPE bit is cleared
8 to 4	—	Reserved When writing, write the value after reset.
3	CAEC	Address error flag clear When the CAEC bit is set to 1, the DMAjCHSTA_n.CAE bit is cleared. The DMAjCHSTA_n.CAE bit is the stop condition of DMA transfer. To end DMA transfer, clear the DMAjCHCR_n.DE bit before clearing the DMAjCHSTA_n.CAE bit. 0: No operation 1: DMAjCHSTA_n.CAE bit is cleared
2	DSEC	Descriptor step end flag clear When the DSEC bit is set to 1, the DMAjCHSTA_n.DSE bit is cleared. 0: No operation 1: DMAjCHSTA_n.DSE bit is cleared

Table 7.18 DMAjCHFCR_n Register Contents (2/2)

Bit Position	Bit Name	Function
1	TEC	<p>Transfer end flag clear</p> <p>When the TEC bit is set to 1, the DMAjCHSTA_n.TE bit is cleared. The DMAjCHSTA_n.TE bit is the stop condition of DMA transfer. To end DMA transfer, clear the DMAjCHCR_n.DE bit before clearing the DMAjCHSTA_n.TE bit.</p> <p>0: No operation 1: DMAjCHSTA_n.TE bit is cleared</p>
0	DEC	<p>DMA enable clear</p> <p>When the DEC bit is set to 1, the DMAjCHCR_n.DE bit is cleared. The DMAjCHCR_n.DE bit is the enable condition of DMA transfer. To suspend DMA transfer, clear the DE bit by using this bit. See Section 7.5, Usage Notes for details about DMA transfer suspension.</p> <p>0: No operation 1: DMAjCHCR_n.DE bit is cleared</p>

7.3.17 DMAjGIAI_n — DMA Gather Inner Address Increment Register

DMAjGIAI_n is a 32-bit read/write register that specifies the source address increment for the inner loop of a gather transfer.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GIAI[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GIAI[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.3.18 DMAjGOAI_n — DMA Gather Outer Address Increment Register

DMAjGOAI_n is a 32-bit read/write register that specifies the source address increment for the outer loop of a gather transfer.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GOAI[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GOAI[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.3.19 DMAjSIAI_n — DMA Scatter Inner Address Increment Register

DMAjSIAI_n is a 32-bit read/write register that specifies the destination address increment for the inner loop of a scatter transfer.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIAI[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIAI[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.3.20 DMAjSOAI_n — DMA Scatter Outer Address Increment Register

DMAjSOAI_n is a 32-bit read/write register that specifies the destination address increment for the outer loop of a scatter transfer.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SOAI[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOAI[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.3.21 DMAjSGST_n — DMA Scatter Gather Status Register

DMAjSGST_n is a 32-bit read only register that indicates the count of inner scatter and gather loops. The detailed operation of scatter/gather is explained in **Section 7.4.5, Scatter Gather Transfer**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SICNT[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	GICNT[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7.19 DMAjSGST_n Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved When read, the value after reset is returned.
29 to 16	SICNT[13:0]	Indicates the current repeat count of the scatter inner DMA loop.
15, 14	—	Reserved When read, the value after reset is returned.
13 to 0	GICNT[13:0]	Indicates the current repeat count of the gather inner DMA loop.

7.3.22 DMAjSGCR_n — DMA Scatter Gather Control Register

DMAjSGCR_n is a 32-bit read/write register that controls scattering on the source side and gathering on the destination side. The operation of scatter/gather is explained in detail in **Section 7.4.5, Scatter Gather Transfer**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEN	ZF	SIRPT[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GEN	—	GIRPT[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.20 DMAjSGCR_n Register Contents

Bit Position	Bit Name	Function
31	SEN	Scatter enable on source side 0: Scattering is disabled 1: Scattering is enabled
30	ZF	Zero fill 0: Zero fill is disabled 1: Zero fill is enabled
29 to 16	SIRPT[13:0]	Repeat count of the inner DMA loop if scattering is enabled 0: Zero repeats 1: One repeat : 16383: 16383 repeats
15	GEN	Gather enable on destination side 0: Gathering is disabled 1: Gathering is enabled
14	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
13 to 0	GIRPT[13:0]	Repeat count of the inner DMA loop if gathering is enabled 0: Zero repeats 1: One repeat : 16383: 16383 repeats

7.3.23 DMAjRS_n — DMA Resource Select Register

DMAjRS_n is a 32-bit read/write register that specifies the source and control settings for hardware transfer requests. The operation of hardware transfer requests is explained in detail in **Section 7.4.1, DMA Transfer Requests**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TL[2:0]			FPT	PLE	DRQI	—	RS[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.21 DMAjRS_n Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	TC[15:0]	Transfer count per hardware request Specify the number of DMA transfers in hardware request mode. This bit is valid only when the DMAjRS_n.TL bit is set to 000 or 001. In this case, setting 0 is prohibited.
15	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
14 to 12	TL[2:0]	Transfer limit per hardware request Specifies the limit of transfers generated by a hardware request. 000: Transaction size indicated by DMAjTMR_n.STS * DMAjRS_n.TC (Setting prohibited when PLE = 1) 001: Transaction size indicated by DMAjTMR_n.DTS * DMAjRS_n.TC (Setting prohibited when PLE = 0) 010: Transfer size indicated by DMAjTSR_n.TSR 011: Until the DSE flag is asserted. (Transfer also ends when the TE flag is asserted.) 100: Until the TE flag is asserted Other than above: Setting prohibited
11	FPT	First pre-load trigger Specifies the trigger to start the first pre-load transfer. If transfer data is not prepared before DE is set, this bit should be set to 1. When the pre-load function is disabled (PLE = 0), this setting is ignored. 0: First pre-load start when DE is set to 1 1: First pre-load start when hardware request is asserted
10	PLE	Pre-load enable When the pre-load function is enabled, the DMAC loads transfer data from the source memory before a hardware request is asserted. 0: Disable 1: Enable
9	DRQI	DMA request initialization when descriptor settings are loaded Specifies whether to initialize the DRQ bits when this register setting is loaded from the descriptor memory. When DRQI is set to 1, the remaining DRQ bits are cleared. The DRQ bits are also cleared when a different source (RS) is loaded. For details, see Section 7.4.6(4)(h), Example 8: Hardware Request Operation with Different Request Source Settings . 0: DRQ initialize disabled 1: DRQ initialize enabled

Table 7.21 DMAjRS_n Register Contents (2/2)

Bit Position	Bit Name	Function
8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	RS[7:0]	DMA request source See the DMA request resources of on-chip peripheral modules.

7.3.24 DMAjBUF_{CR_n} — DMA Buffer Control Register

DMAjBUF_{CR_n} is a 32-bit read/write register that controls the upper limit of the data RAM buffer. This register specifies the upper limit of the RAM buffer available to a channel for data prefetching. This setting should not be changed unless it is necessary to limit the amount of prefetched data.

Value after reset: 0000 0080_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ULB[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.22 DMAjBUF_{CR_n} Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	ULB[7:0]	Upper limit of buffer This register controls the upper limit of a channel's data buffer. A power-of-two setting is recommended. Setting ULB to less than the transaction size (DMAjTMR _n .STS, DMAjTMR _n .DTS) is prohibited. The maximum value is 128 (bytes).

7.3.25 DMAjDPPTR_n — DMA Descriptor Pointer Register

DMAjDPPTR_n is a 32-bit read/write register that specifies the address of the next register setting in the descriptor memory. The operation of descriptors is explained in detail in **Section 7.4.6, Descriptors**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PTR[11:2]										DIE	CF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.23 DMAjDPPTR_n Register Contents

Bit Position	Bit Name	Function
31 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 2	PTR[11:2]	Address pointer of Descriptor These bits indicate the address of the next register setting in the descriptor memory.
1	DIE	Descriptor interrupt enable When the DMAjCHCR_n.DSIE bit and this bit are set to 1 at the end of the current DMA transfer, the DSE bit in DMAjCHSTA_n set to 1 and the descriptor step end interrupt is issued. 0: Interrupt request is disabled 1: Interrupt request is enabled
0	CF	Continuation flag of descriptor 0: Disable Transfer is not continued. The TE bit in the DMAjCHCR_n register is set to 1 at the termination of the current DMA transfer. 1: Enable When DMA transfer with the current register settings is terminated and DMAjCHCR_n.DPE is set to 1, transfer continues to read the register settings from the DPTR address of the descriptor memory.

7.3.26 DMAjDPCR_n — DMA Descriptor Control Register

DMAjDPCR_n is a 32-bit read/write register that specifies which channel settings will be updated with the contents of the descriptor memory. The operation of descriptors is explained in details in **Section 7.4.6, Descriptors**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	UPF[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.24 DMAjDPCR_n Register Contents

Bit Position	Bit Name	Function
31 to 11	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	UPF[10:0]	Update flag of Descriptor These bits specify the registers to be updated by a descriptor operation. The DMAjDPTR_n register is always updated. UPF[0]: Enable/disable updating the DMAjSAR_n register UPF[1]: Enable/disable updating the DMAjDAR_n register UPF[2]: Enable/disable updating the DMAjTSR_n register UPF[3]: Enable/disable updating the DMAjTMR_n register UPF[4]: Enable/disable updating the DMAjGIAI_n register UPF[5]: Enable/disable updating the DMAjGOAI_n register UPF[6]: Enable/disable updating the DMAjSIAI_n register UPF[7]: Enable/disable updating the DMAjSOAI_n register UPF[8]: Enable/disable updating the DMAjSGCR_n register UPF[9]: Enable/disable updating the DMAjRS_n register UPF[10]: Enable/disable updating the DMAjBUFCR_n register

7.3.27 Descriptor RAM — Descriptor Memory

See **Section 7.4.6, Descriptor** for details about the descriptor memory.

7.3.28 DMACSELj_m — sDMACj Transfer Request Group Selection Register m (m = 0 to 15)

DMACSELj_m is a 32-bit read/write register that specifies the sDMACj transfer request group for each channel.

Value after reset: DMACSEL0_m: 0000 0000_H
DMACSEL1_m: 5555 5555_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	SEL15	—	SEL14	—	SEL13	—	SEL12	—	SEL11	—	SEL10	—	SEL9	—	SEL8
Value after reset	0	*	0	*	0	*	0	*	0	*	0	*	0	*	0	*
R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	SEL7	—	SEL6	—	SEL5	—	SEL4	—	SEL3	—	SEL2	—	SEL1	—	SEL0
Value after reset	0	*	0	*	0	*	0	*	0	*	0	*	0	*	0	*
R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W

Note: This value is 0_B (sDMAC0), 1_B (sDMAC1).

Table 7.25 DMACSELj_m Register Contents (1/2)

Bit Position	Bit Name	Function
31	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
30	SEL15	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 15
29	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
28	SEL14	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 14
27	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
26	SEL13	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 13
25	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
24	SEL12	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 12
23	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
22	SEL11	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 11
21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	SEL10	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 10
19	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
18	SEL9	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 9
17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	SEL8	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 8
15	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
14	SEL7	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 7

Table 7.25 DMACSELj_m Register Contents (2/2)

Bit Position	Bit Name	Function
13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12	SEL6	Selects the DMA transfer request group for sDMACj transfer request signal $16 \times m + 6$
11	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
10	SEL5	Selects the DMA transfer request group for sDMACj transfer request signal $16 \times m + 5$
9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	SEL4	Selects the DMA transfer request group for sDMACj transfer request signal $16 \times m + 4$
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	SEL3	Selects the DMA transfer request group for sDMACj transfer request signal $16 \times m + 3$
5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	SEL2	Selects the DMA transfer request group for sDMACj transfer request signal $16 \times m + 2$
3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	SEL1	Selects the DMA transfer request group for sDMACj transfer request signal $16 \times m + 1$
1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SEL0	Selects the DMA transfer request group for sDMACj transfer request signal $16 \times m + 0$

The following table shows the DMACSELj_0 to DMACSELj_15 register contents.

Bit Name	Function
SELn	0 _B : Select DMA transfer request group 0 1 _B : Select DMA transfer request group 1

7.4 Operation

7.4.1 DMA Transfer Requests

The DMAC starts DMA transfer by accepting a hardware DMA transfer request or software DMA transfer request. The DMA transfer request source (TRS) bit in the DMA transfer mode register (DMAjTMR_n) determines whether a hardware DMA transfer request or a software DMA transfer request is used. How to set up DMA transfer requests is described in **Section 7.7, Setting up DMA Transfer**.

(1) Auto-Request Mode

When the TRS bit in the DMAjTMR_n register is set to 0, a software DMA transfer request can be generated. When the DE bit in DMAjCHCR_n and the DME bit in DMAjOR are set to 1 for the target channel, the transfer begins providing the CAE bit in DMAjCHSTA_n is 0.

(2) Hardware-Request Mode

When the TRS bit in the DMAjTMR_n register is set to 1, if DMA transfer is enabled (DE = 1, DME = 1, TE = 0, CAE = 0), a transfer is performed upon the input of a transfer request signal.

In the case of a hardware DMA transfer request for the DMAC, one out of 256 hardware DMA transfer sources is selected and assigned to each channel of the DMAC in the DMA request control module. This assignment is configured by the resource select (RS) bit in the DMAjRS_n register.

Examples of using hardware request mode are shown below.

(a) Example 1: Hardware Request Operation to Transfer Data from a Peripheral Module

Figure 7.3 and Figure 7.4 show examples of a peripheral module requesting sDMAC transfer from peripheral memory to on-chip memory. When the hardware transfer request triggers a load data transfer (DMAjRS_n.PLE = 0), the DMAjRS_n.TL bit determines the load data transfer limit per hardware transfer request. Store data transfers are executed in auto request mode in this case.

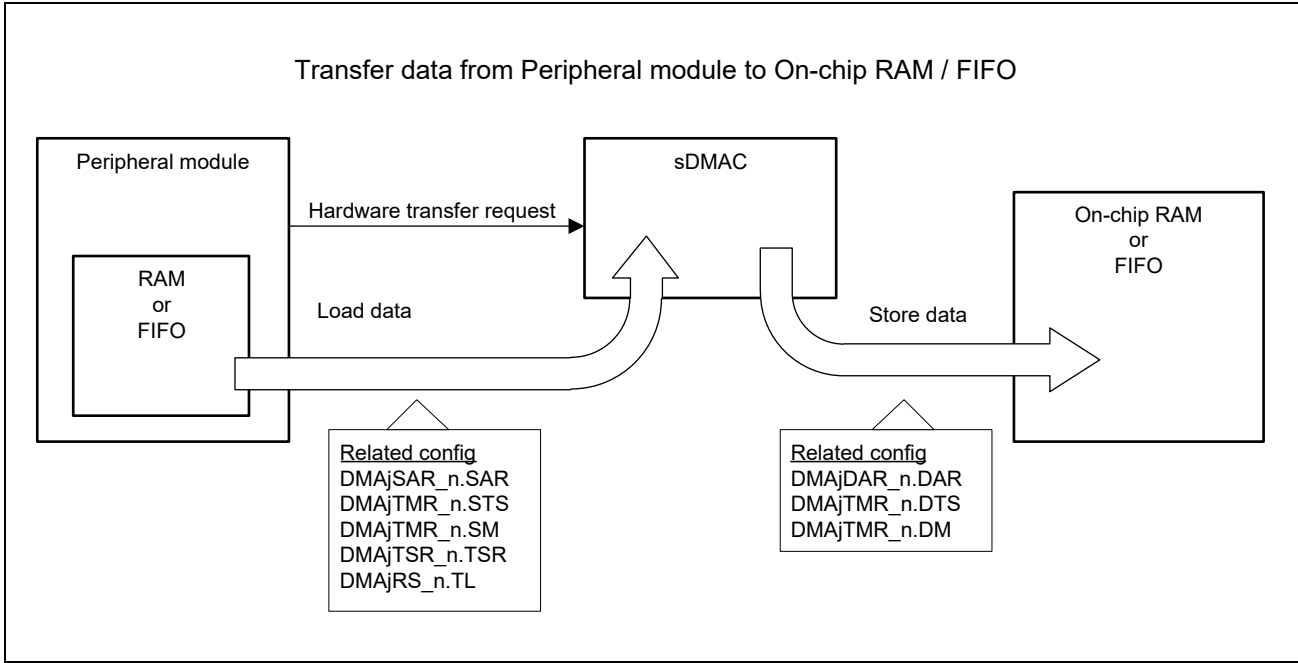


Figure 7.3 Transferring Data from Peripheral Module in Hardware-Request Mode

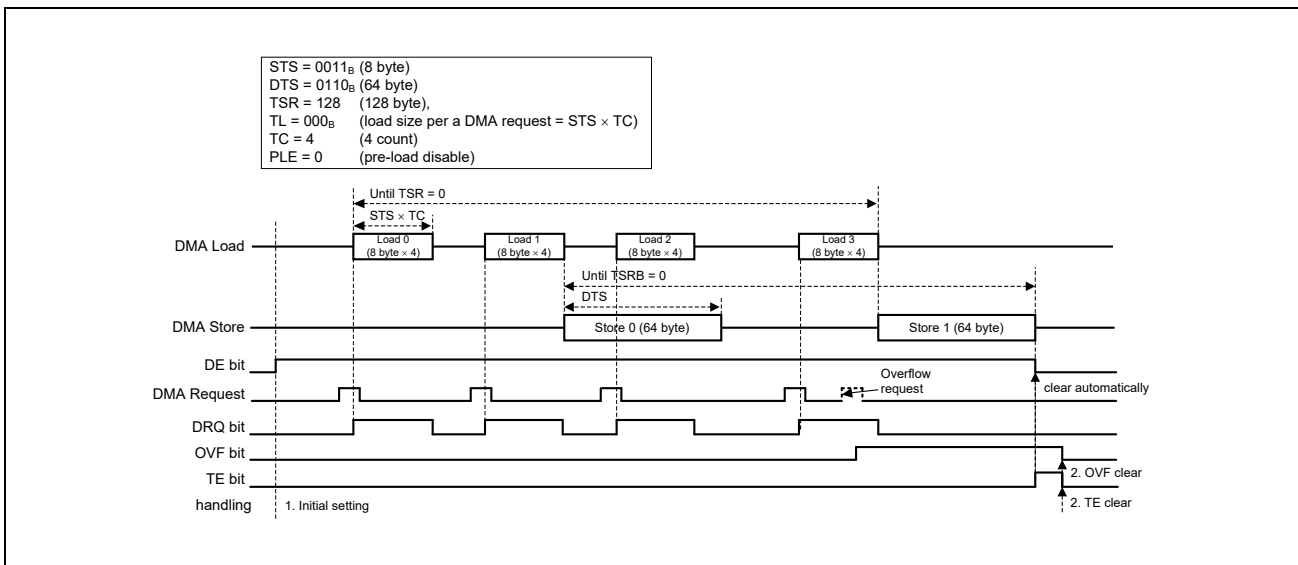


Figure 7.4 Timing Chart for Transferring Data from Peripheral Module

(b) Example 2: Hardware Request Operation to Transfer Data to a Peripheral Module

Figure 7.5, Figure 7.6, Figure 7.7, and Figure 7.8 show examples of a peripheral module requesting sDMAC transfer from on-chip memory to peripheral module memory. When the pre-load function is enabled ($\text{DMAjRS}_n.\text{PLE} = 1$), a hardware transfer request triggers a store data transfer. The $\text{DMAjRS}_n.\text{TL}$ bit determines the store data transfer limit per hardware transfer request. Load data transfers are executed in auto request mode in this case.

In case the pre-load transfer is triggered before the load data is prepared in the source memory, the pre-load limit can be controlled by setting the ULB bit in DMAjBUFCR_n register as shown in **Figure 7.8**.

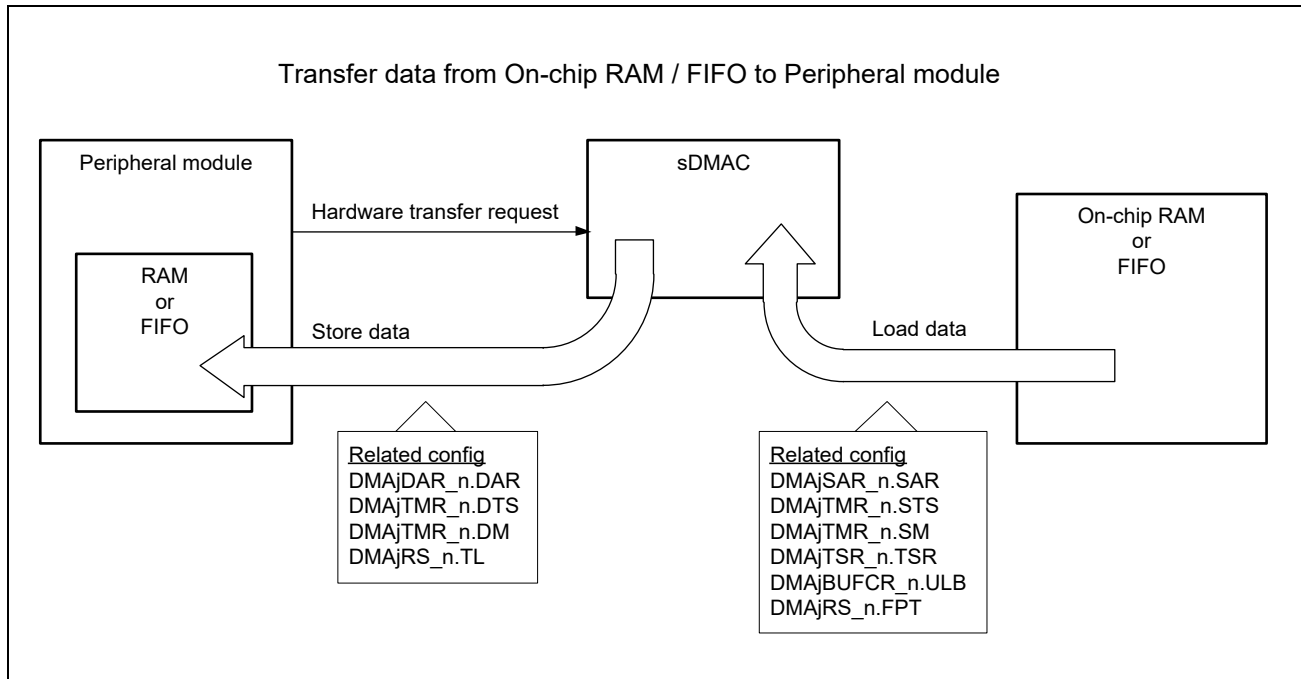


Figure 7.5 Transferring Data to a Peripheral Module in Hardware Request Mode

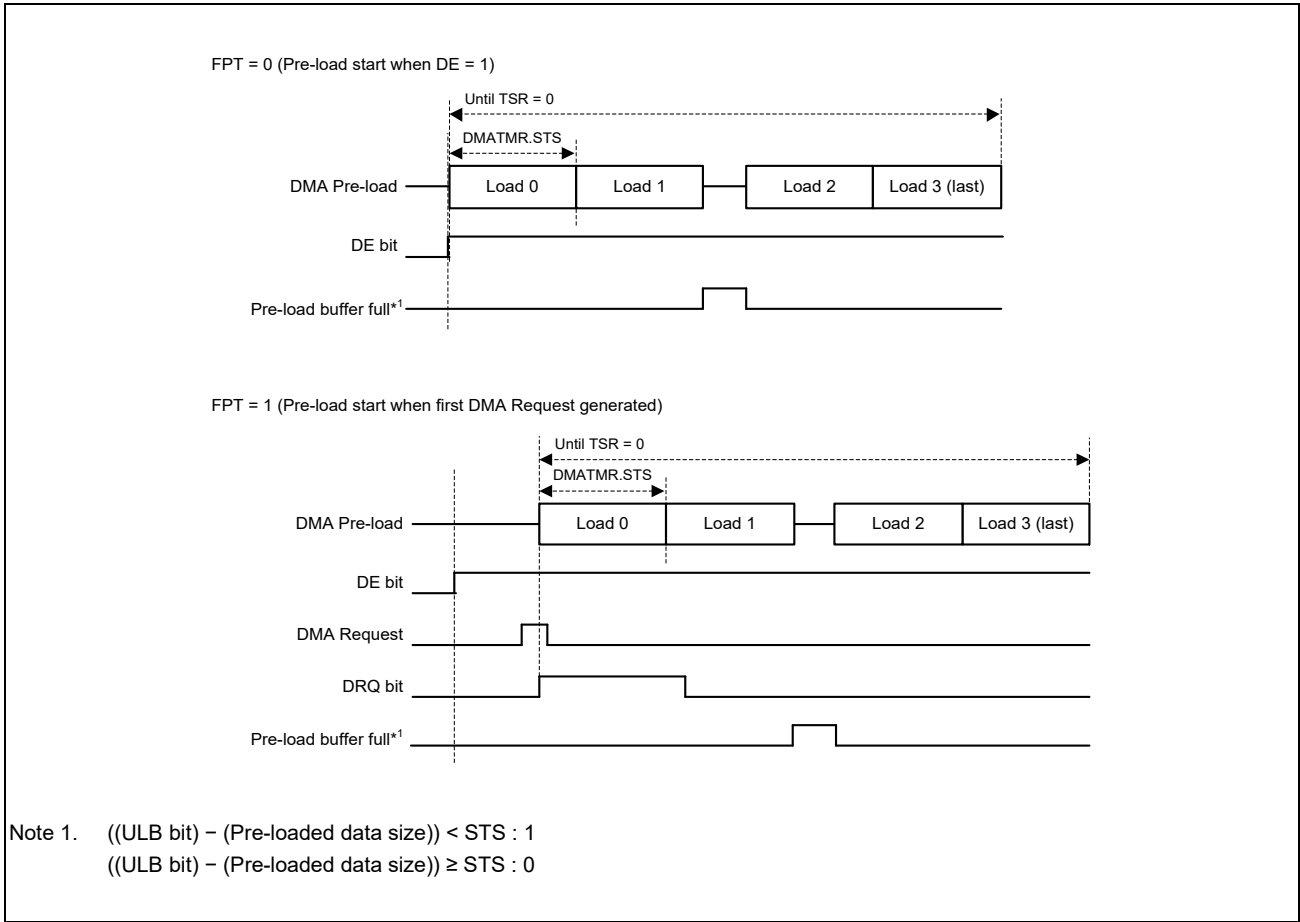


Figure 7.6 Timing Chart for Pre-Load Function in Hardware Request Mode

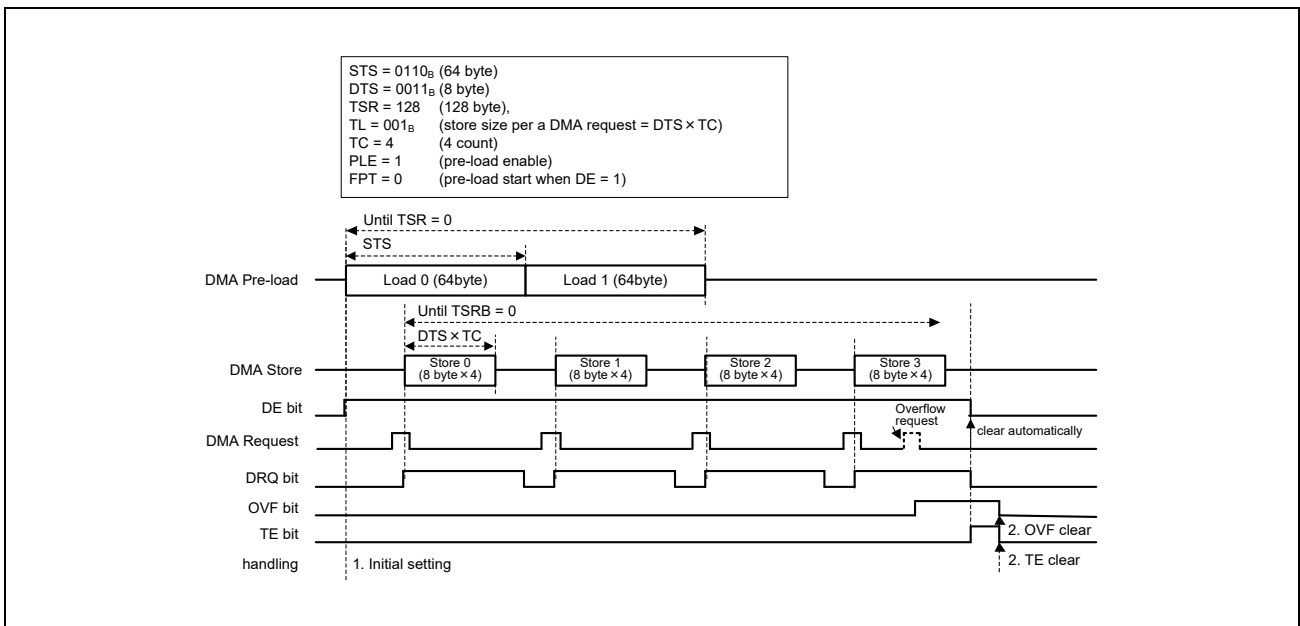


Figure 7.7 Timing Chart of Transferring Data to Peripheral Module

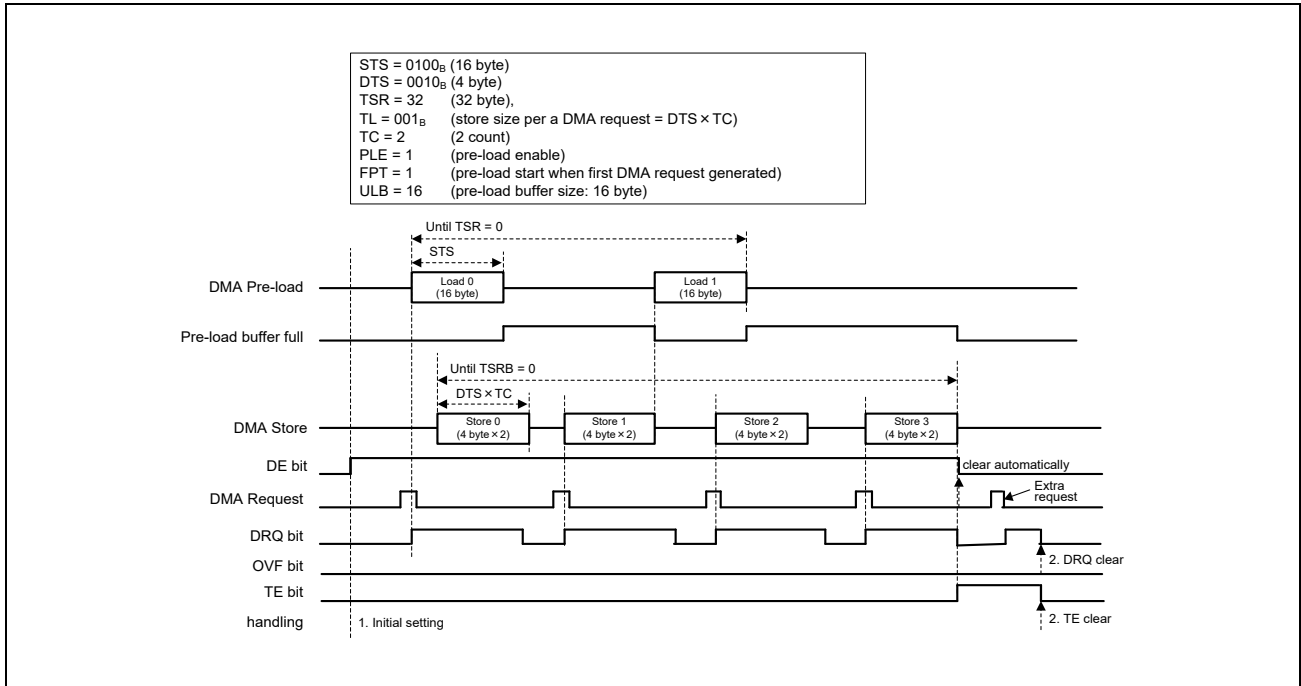


Figure 7.8 Timing Chart of Transferring Data with Buffer Control Operation

7.4.2 DMA Interrupts

The routing of sDMAC interrupts is shown in **Figure 7.9**. sDMAC interrupts can be masked by the DMA channel registers.

A channel interrupt request is raised by sDMAC if any of these status bits is set (OR function):

- Transfer end (TE),
- Descriptor step end (DSE)
- Channel address error (CAE).
- Address error notifications are also routed to the CPU by another path.

The status of channel interrupt requests and address errors is indicated in global registers DMAjISTA and DMAjESTA.

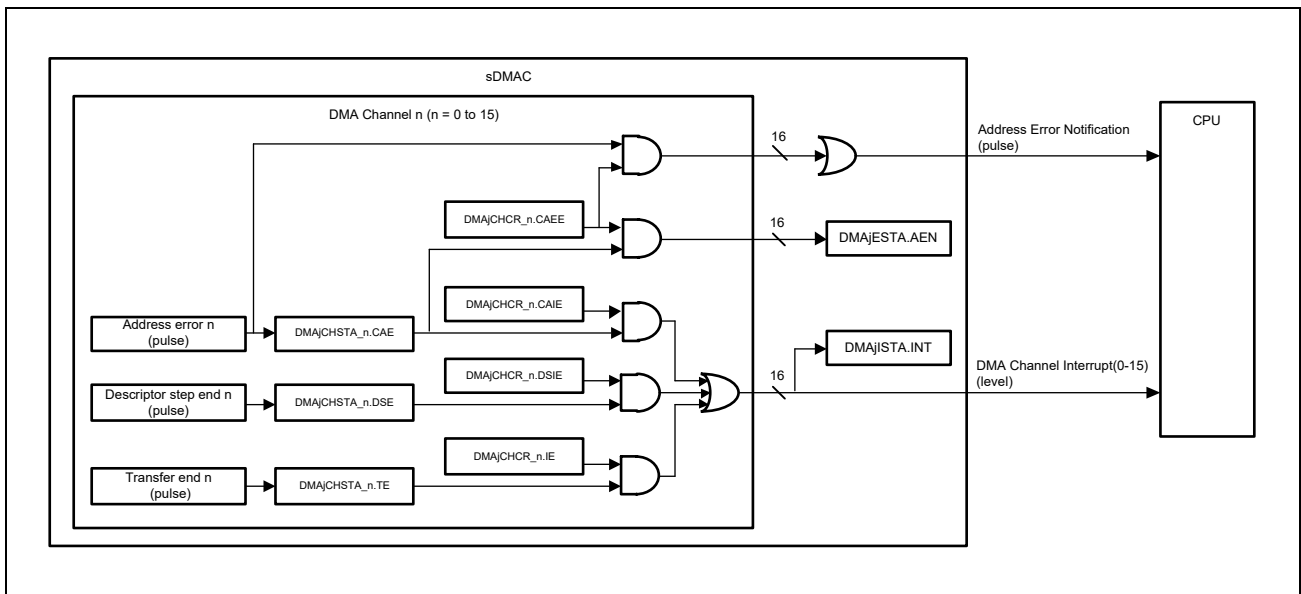


Figure 7.9 Interrupt Routing in sDMAC

7.4.3 Channel Priority

When sDMAC receives transfer requests from multiple channels, it transfers data according to a predetermined priority. One of two modes (fixed mode or round-robin mode) can be selected (DMAjOR.PR[1:0]).

(1) Fixed Mode

In this mode, the priority levels among the channels remain fixed.

CH0 > CH1 > ... > CH14 > CH15

(2) Round-Robin Mode

In round-robin mode, each time data of one transfer unit (1 to 64 byte units) is transferred on one channel, the priority is rotated. The channel on which the transfer has just finished rotates to the bottom of the priority. The priority of round-robin mode is CH0 > CH1 > ... > CH14 > CH15 immediately after reset.

(3) Channel Request Priority

To increase the priority of the specified channel, use the PRI bits in the DMA transfer mode register (DMAjTMR_n). This function is given priority over the PR bits in the DMA operation register (DMAjOR). If a channel request is not accepted, the channel priority counter increases by the count set by the PRICNT bits in the DMA channel request priority register (DMAjCHPRI). The PRICNT bits in the DMA channel request priority register (DMAjCHPRI) are counted up by SCLK. When this priority counter reaches the maximum value (= 7), the highest priority request is accepted. However, if more than one channel has high priority, the requests are arbitrated according to the setting of the PR bits in DMAjOR.

Figure 7.10 shows the channel request priorities.

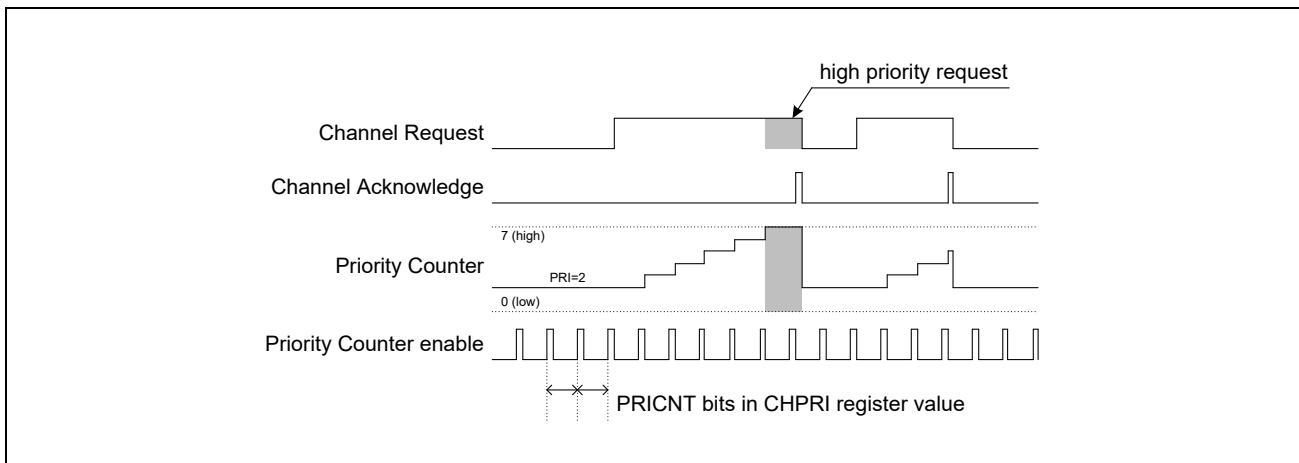


Figure 7.10 Channel Request Priorities

7.4.4 Slow Speed Mode

In slow speed mode, there is a configurable delay between consecutive DMA transfers. The delay (number of clock cycles) is specified by the SLM bit in the DMAjTMR_n register. Slow speed mode can be selected independently for each DMA channel. When one DMA channel operating in slow speed mode is waiting between DMA transfers, another DMA channel operating in slow speed mode can perform DMA transfer.

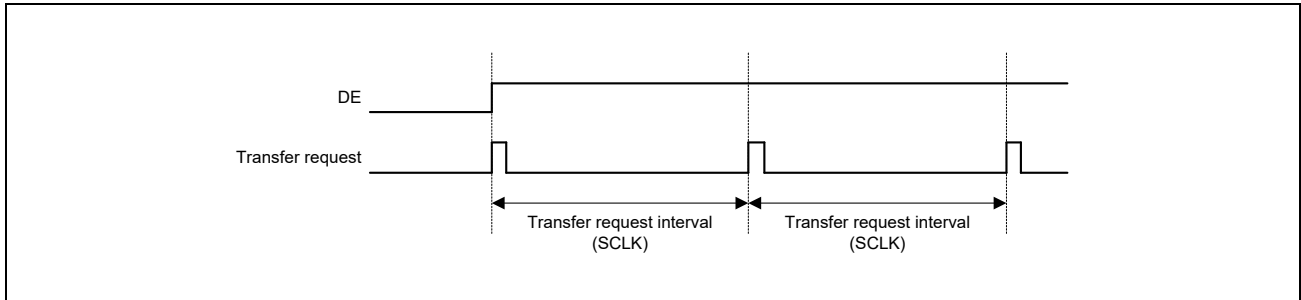


Figure 7.11 Slow Speed Mode

7.4.5 Scatter Gather Transfer

The DMAC supports gathering on the source side and scattering on the destination side.

The gather flow is shown in **Figure 7.12**. It is enabled if the GEN bit in DMAjSGCR_n for the given DMA channel is set to 1. The gather flow supports an inner and an outer loop. The number of iterations of the inner loop is specified by the GIRPT bit in DMAjSGCR_n register. The total of inner and outer loop transfers equals the number of TSR bytes. So the number of iterations of the outer loop is indirectly defined by the TSR register. The inner loop always transfers a data unit the size of an STS byte from the source to the data RAM and the SAR register is updated in the same way as transfers with gathering disabled. Additionally, the source address (SAR bits in DMAjSAR_n register) is increased by the inner address increment (GIAI bits in DMAjGIAI_n register). This procedure is repeated until the value of TSR bits in DMAjTSR_n or the value of GICNT bits in DMAjSGST_n register reaches zero, which concludes the inner loop. Before the inner loop is repeated, the source address (SAR bits in DMAjSAR_n register) is increased by the outer address increment (GOAI bits in DMAjGOAI_n register) and the GICNT bits in DMAjSGST_n register are restored with the value of GIRPT bits in DMAjSGCR_n register. The outer loop is repeated until the value of TSR bits in DMAjTSR_n register reaches zero. **Figure 7.14** shows an example with gathering and scattering enabled. If a transaction error occurs while the gather flow is active, the gather flow restarts at the beginning after the error condition is cleared.

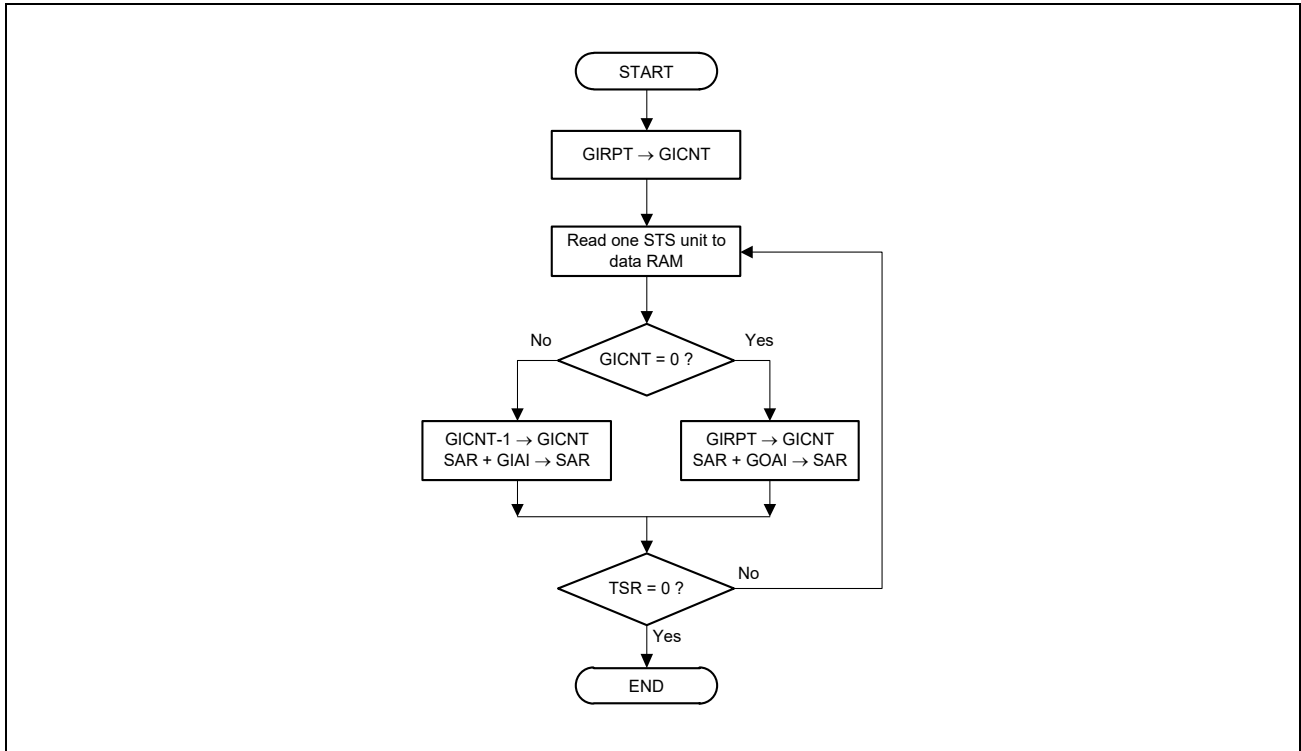


Figure 7.12 Gather Flow at Source Side If GEN = 1

The scatter flow is shown in **Figure 7.13**. It is enabled if the SEN bit in DMAjSGCR_n for the given DMA channel is set to 1. The scatter flow also supports an inner and outer loop. Additionally, zero filling is supported in the inner loop. The number of iterations of the inner loop is specified by the SIRPT register. The total of inner and outer loop transfers equals the number of TSRB bytes. So the number of iterations of the outer loop is indirectly defined by the TSRB register. The inner loop always transfers one data unit of size specified by DTS bits in DMAjTMR_n from the data RAM to the destination address and the DAR bits in DMAjDAR_n register is updated in the same way as for transfers with scattering disabled. If zero filling is disabled by setting ZF bit in DMAjSGCR_n = 0, the destination address (DAR bits in DMAjDAR_n register) is additionally increased by the inner address increment (SIAI bits in DMAjSIAI_n register), resulting in the destination address for the next data unit (the value of DAR bits in DMAjDAR_n register plus the value of SIAI bits in DMAjSIAI_n register). If zero filling is enabled by setting ZF bit in DMAjSGCR_n register = 1, value zero with data units specified by SIAI bits in DMAjSIAI_n register are written to the destination address and the DAR bits in DMAjDAR_n register is updated. This procedure also uses internal register ZFTSR to calculate the remaining zero fill transfer size. This procedure is repeated until the value of TSRW bits in DMAjTSRB_n register or the value of SICNT bits in DMAjSGST_n register reaches zero, which concludes the inner loop. Before the inner loop is repeated, the destination address (DAR bits in DMAjDAR_n register) is increased by the outer address increment (SOAI bits in DMAjSOAI_n register) if zero fill is disabled and the SICNT bits in DMAjSGST_n register is restored with the value of SIRPT bits in DMAjSGCR_n register. The outer loop is repeated until the value of TSRW bits in DMAjTSRB_n reaches zero. **Figure 7.14** shows an example with gathering and scattering enabled. If a transaction error occurs while the scatter flow is active, the scatter flow restarts at the beginning after the error condition is cleared.

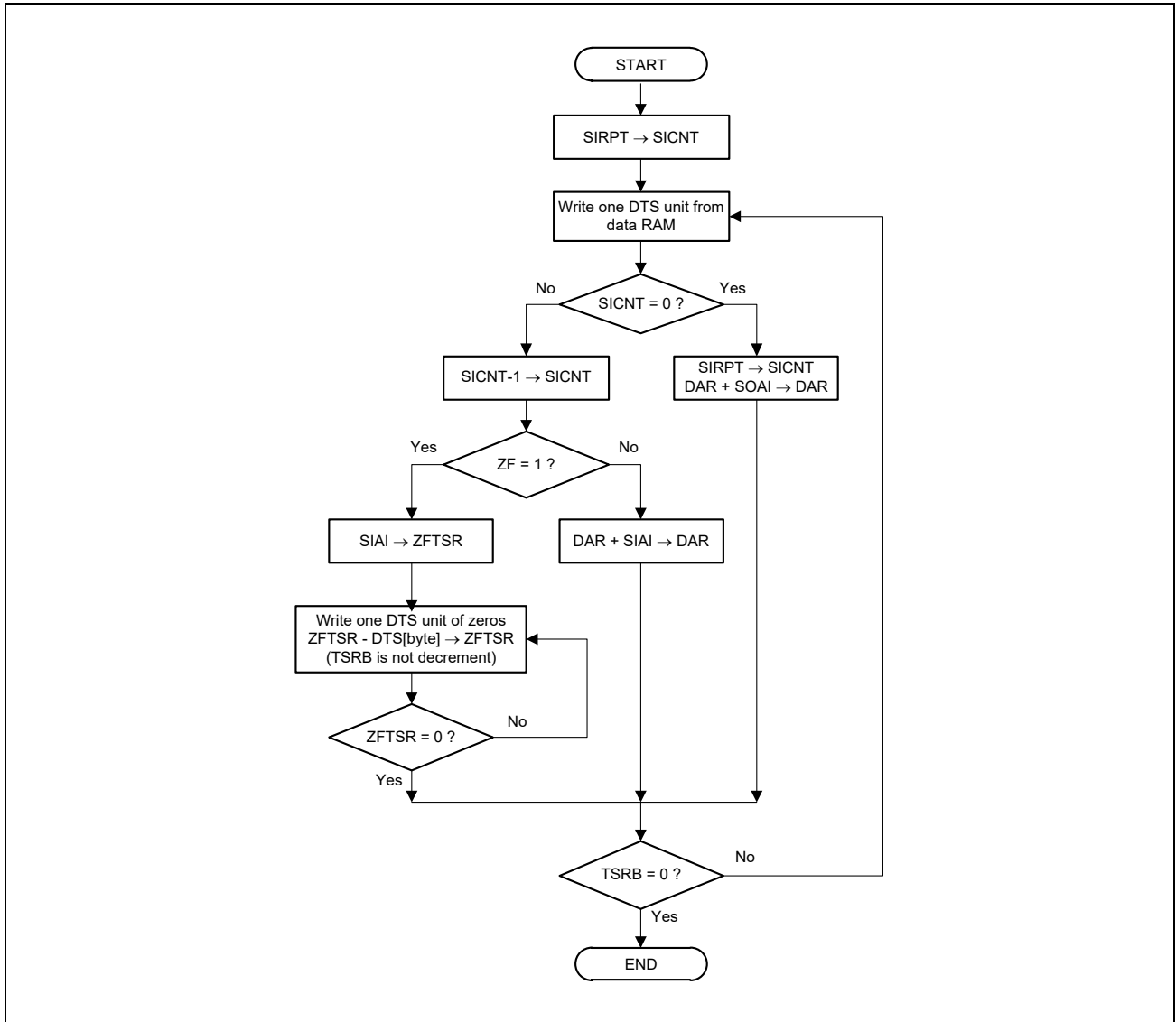


Figure 7.13 Scatter Flow at Destination Side If SEN = 1

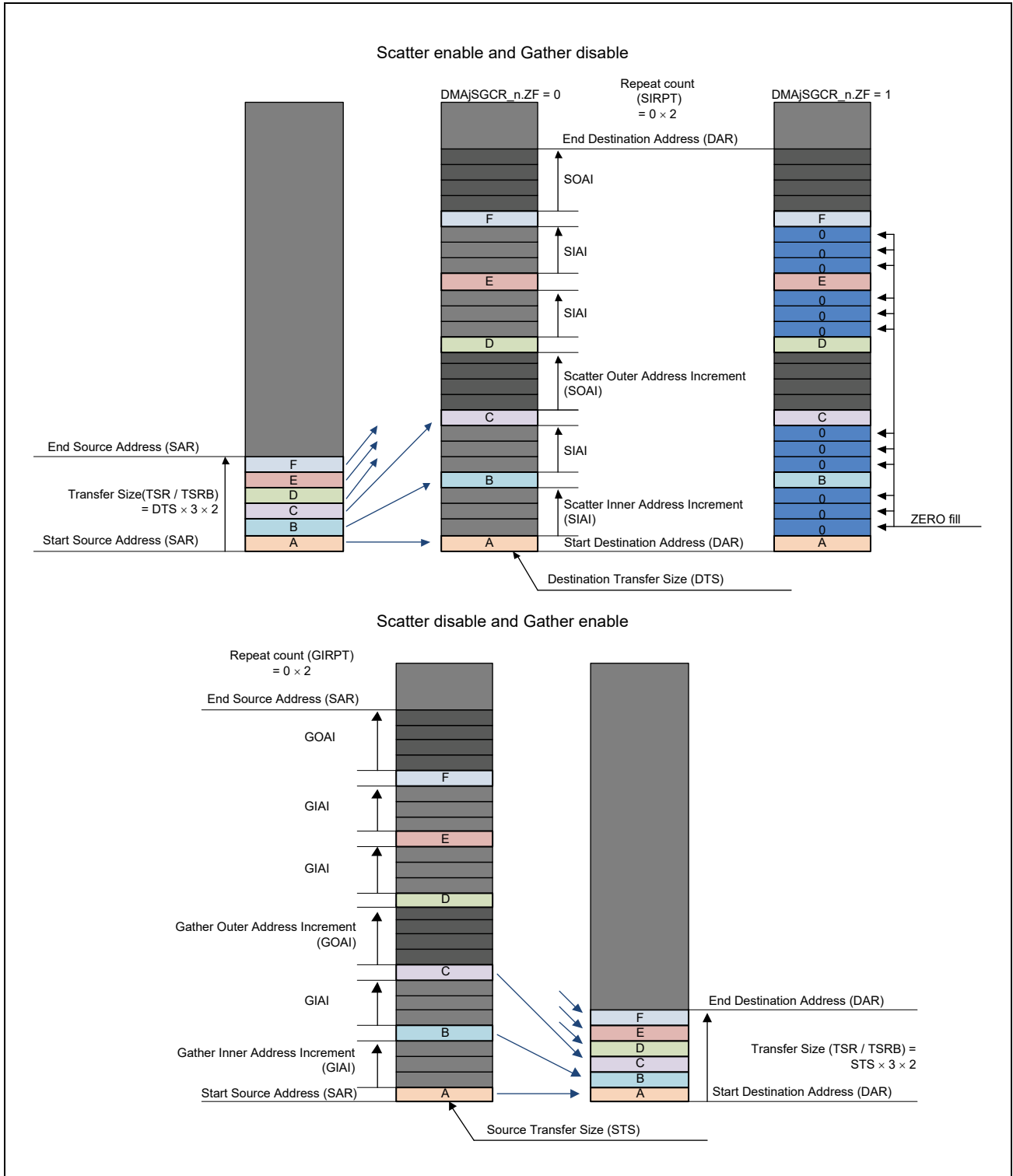


Figure 7.14 Examples That Show the Operation of Scatter and Gather Flows

The scatter gather features can be used to realize re-sorting. Transposing a column vector from a source memory address to a row vector at the destination memory address is realized by using gathering at the source side. The inner loop (GIRPT bits in DMAjSGCR_n register) is set to the number of samples in the vector. The data unit size (STS bits or DTS bits in DMAjTMR_n register) is set to the size of one vector element. The inner address offset (GIAI bits in DMAjGIAI_n register) is set to $k \times STS$, where k is the width of the source matrix in the number of elements and STS is

the value of STS bits in DMAjTMR_n register. The same principle can be used to transpose a row vector to a column vector. Here, scattering is used instead of gathering. Gathering and scattering can also be used to read every m-th source address and write it to every n-th destination address. Since gathering and scattering can be enabled independently, any combination is possible. At the destination side, zero filling can be used to set the words between the write addresses to zero.

Finally, the outer loop feature for both gathering and scattering allows the realization of inner and outer DMA loops.

7.4.6 Descriptors

A descriptor contains all the necessary register bits that describe a single transfer task on a channel. A channel can execute multiple descriptors in a chain by reloading the register settings from the descriptor memory. Descriptor operations are enabled by setting DPE bit to 1_B in DMAjCHCR_n. When the DMA transfer defined by the current register setting is finished, the next descriptor is loaded from the descriptor memory. The address of the descriptor to load is defined by the DMAjDPPTR_n.PTR bits. The new descriptor also updates the DMAjDPPTR_n register. This enables an arbitrary chain of descriptors.

Updating of the register list by descriptor operations is shown in **Table 7.26**.

The bit format in the descriptor memory matches the bit format in the corresponding DMA channel registers.

The DMAjSAR_n, DMAjDAR_n and DMAjTSR_n registers are updated during DMA transfer. If the DMAjSAR_n and DMAjDAR_n registers are not updated, the next DMA transfer is started from the final value of the previous DMA transfer. The DMAjTSR_n register is set to 0 after DMA transfer, so DMAjTSR_n register must be updated.

Table 7.26 Register List of Channels to Be Updated

No.	Name	Abbreviation	Bit	DMADPCR.UPF bit
1	DMA source address register	DMAjSAR_n	All bits	UPF[0]
2	DMA destination address register	DMAjDAR_n	All bits	UPF[1]
3	DMA transfer size register	DMAjTSR_n	All bits	UPF[2]
4	DMA transfer mode register	DMAjTMR_n	All bits	UPF[3]
5	DMA gather inner address increment register	DMAjGIAI_n	All bits	UPF[4]
6	DMA gather outer address increment register	DMAjGOAI_n	All bits	UPF[5]
7	DMA scatter inner address increment register	DMAjSIAI_n	All bits	UPF[6]
8	DMA scatter outer address increment register	DMAjSOAI_n	All bits	UPF[7]
9	DMA scatter gather control register	DMAjSGCR_n	All bits	UPF[8]
10	DMA resource select register	DMAjRS_n	All bits	UPF[9]
11	DMA buffer control register	DMAjBUFCR_n	All bits	UPF[10]

Set as follows for each channel that should use the descriptor function:

- Specify the register to be updated in the UPF bits of the DMAjDPCR_n register.
- Set the DMA transfer setting in the descriptor memory.
- If DMA transfer should start when a configuration is copied from the descriptor memory to DMA channel registers
 - set the DMAjDPPTR_n.PTR bits to the first transfer configuration in the descriptor memory
 - set DMAjCHCR_n.DPB to 1
- If DMA transfer should start according to a register setting and continue with the descriptor function.
 - configure the channel registers
 - set the DMAjDPPTR_n.PTR bits to the address of the next transfer configuration in the descriptor memory
 - set DMAjDPPTR_n.CF to 1
 - set DMAjCHCR_n.DPB to 0

The descriptor memory is shared by all DMA channels. Therefore is it important to avoid overlaps of areas used by different DMA channels in common descriptor memory.

(1) Configuration of Descriptor Memory

A DMA transfer configuration can start from any address in the descriptor memory. Register configurations that are not updated should not be written.

Examples of configurations in descriptor memory are shown in the figure below.

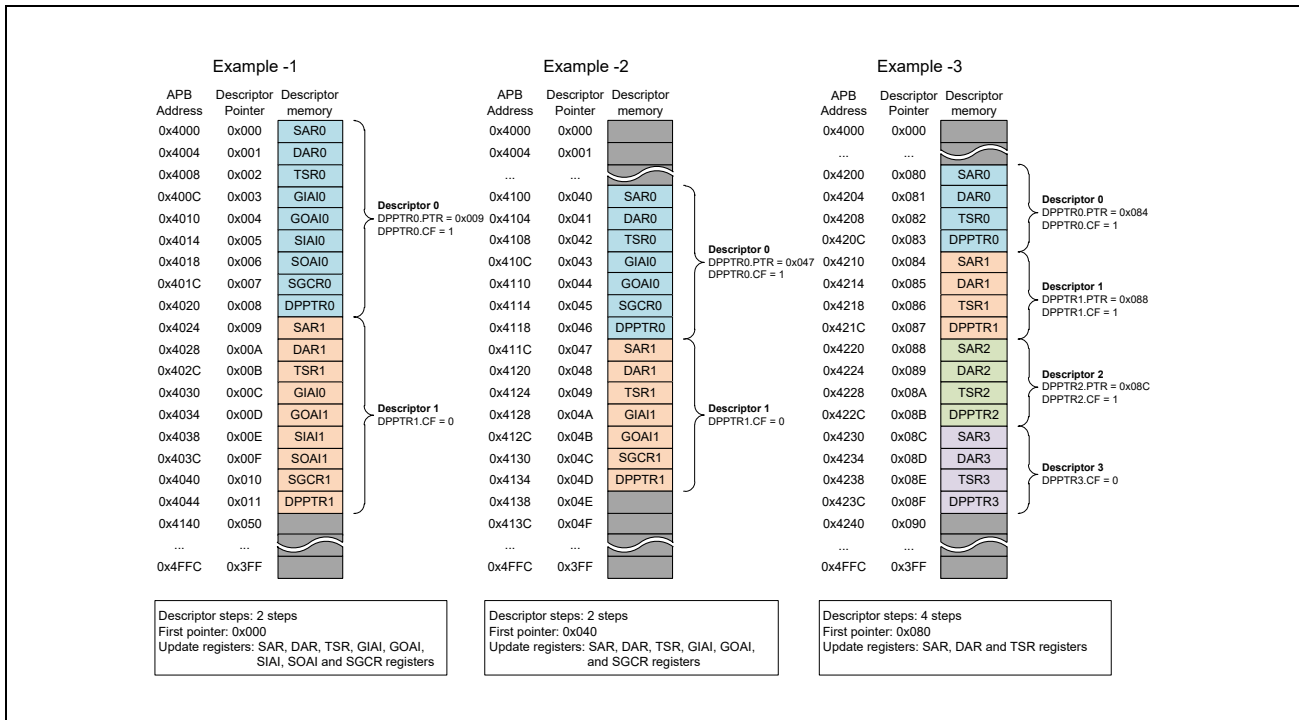


Figure 7.15 Built-in Descriptor Memory Configuration

(2) Flow of Setting Updating by Descriptor Operation

Updating a configuration by using the descriptor function is specified by the UPF bits in the DMAjDPCR_n register. If a register is updated by a descriptor operation, the PTR bit in DMAjDPPTR_n is incremented. Finally the DMAjDPPTR_n register is updated. Therefore, the DMAjDPPTR_n register setting must be always written into the descriptor memory.

This flow is automatically processed by hardware.

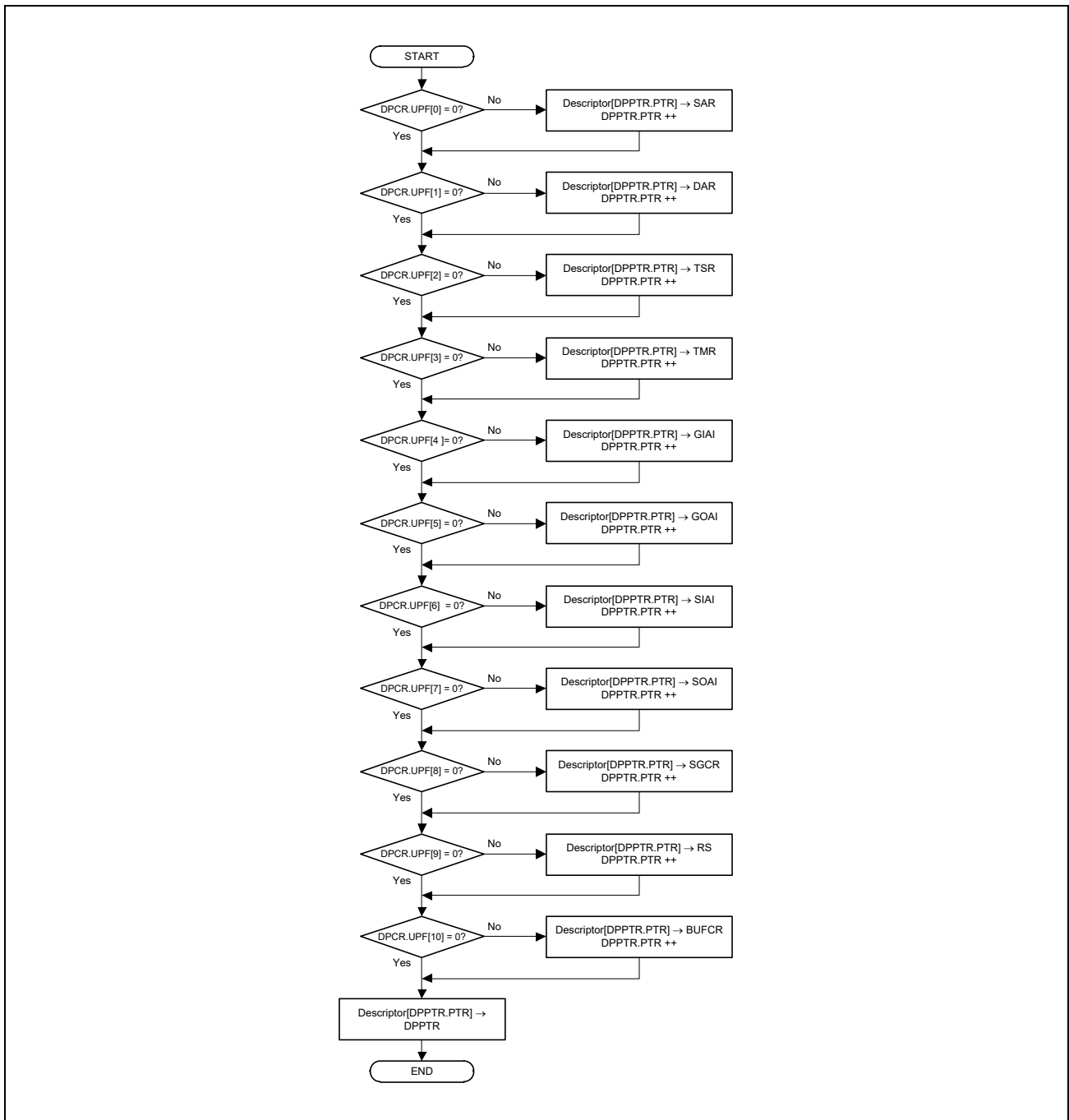


Figure 7.16 Flow of Setting Updating by Descriptor Operation

(3) Descriptor Operational Flow

Figure 7.17 shows the descriptor operational flow.

When the first transfer setting is written in the descriptor memory, the DPB bit in the DMAjCHCR_n register is set to 1. When a DSE interrupt is issued after a step of DMA transfer completion, the DSIE bit in the DMAjCHCR_n register and the DIE bit in DMAjDPPTR_n in the descriptor memory are set to 1. A DSE interrupt can be issued at the optional timing by using this setting. When DSE bit in DMAjCHSTA_n register is set already at the time of a step of DMA transfer completion, the next transfer setting is not read from the descriptor memory. There is a possibility that the next setting is not written in the descriptor memory because the first DSE interrupt has not been processed. The descriptor can be continued by clearing the first DSE interrupt. When the DPE bit in DMAjCHCR_n register is set to 0 or the CF bit in DMAjDPPTR_n register in the Descriptor memory is set to 0 after a step of DMA transfer completion, TE interrupt is issued, and Descriptor is ended.

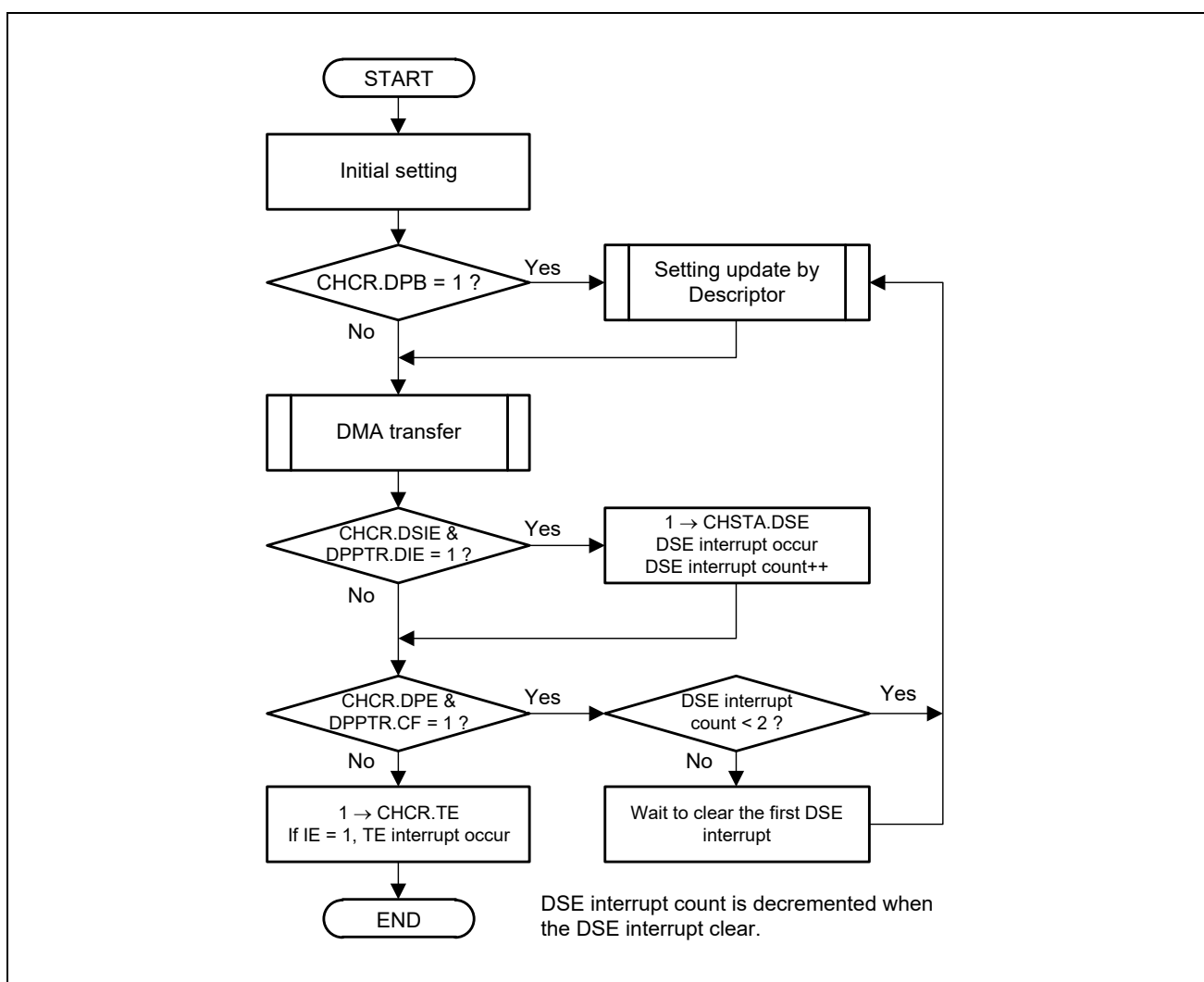


Figure 7.17 Descriptor Operational Flow

(4) Example of descriptor usage

Figure 7.18 to Figure 7.25 show examples of using descriptors operations.

(a) Example 1: Normal Descriptor Operations

The DMAC transfers descriptors in the listed order and generates an interrupt after all of the transfers are complete.

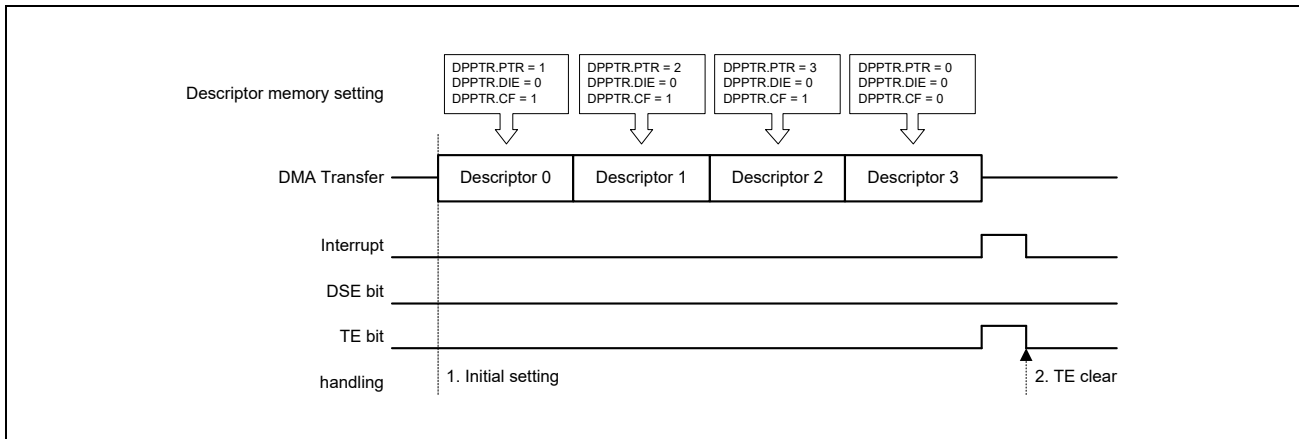


Figure 7.18 Normal Descriptor Operations (Example 1)

No.	Handling	Detail
1	Initial setting	<p>Specify the register to be updated. Write to the DPCR register. Descriptors 0 to 3 are written to the descriptor memory.</p> <p>Descriptor[0].DPPTR CF = 1, DIE = 0, PTR = Descriptor 1 pointer</p> <p>Descriptor[1].DPPTR CF = 1, DIE = 0, PTR = Descriptor 2 pointer</p> <p>Descriptor[2].DPPTR CF = 1, DIE = 0, PTR = Descriptor 3 pointer</p> <p>Descriptor[3].DPPTR CF = 0, DIE = 0, PTR = Any value</p> <p>The first descriptor pointer is written to the DPPTR register. DPPTR.PTR = 0</p> <p>Start the descriptor operation (write to CHCR) Descriptor enable (CHCR.DPE = 1) TE interrupt enable (CHCR.IE = 1) DSE interrupt disable (CHCR.DSIE = 0) To start the transfer after reading descriptor memory (CHCR.DPB = 1)</p>
2	TE clear	<p>Check the interrupt status (read CHSTA) Disable DMA transfer (CHCR.DE = 0) Clear the transfer end flag (CHFCR.TEC = 1)</p>

(b) Example 2: Specific Order Operations

The DMAC transfers descriptors in the specified order and generates an interrupt after all of the transfers are complete.

Descriptor 0 → Descriptor 1 → Descriptor 3 → Descriptor 2

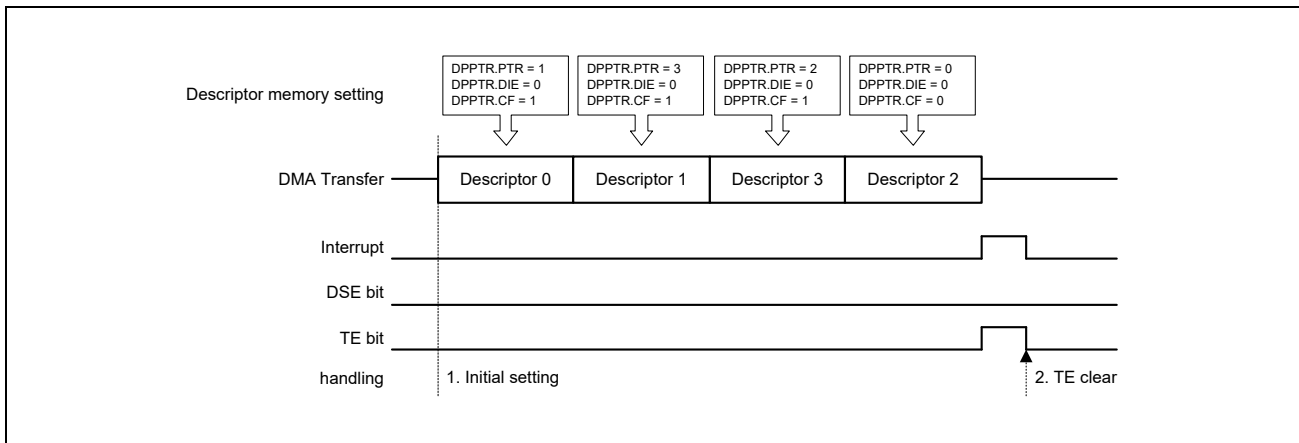


Figure 7.19 Specific Order Operation (Example 2)

No.	Handling	Detail
1	Initial setting	<p>Specify the register to be updated. Write to the DPCR register. Descriptors 0 to 3 are written to the descriptor memory.</p> <p>Descriptor[0].DPPTTR CF = 1, DIE = 0, PTR = Descriptor 1 pointer</p> <p>Descriptor[1].DPPTTR CF = 1, DIE = 0, PTR = Descriptor 3 pointer</p> <p>Descriptor[2].DPPTTR CF = 0, DIE = 0, PTR = Any value</p> <p>Descriptor[3].DPPTTR CF = 1, DIE = 0, PTR = Descriptor 2 pointer</p> <p>The first descriptor pointer is written to the DPPTTR register. DPPTTR.PTR = 0</p> <p>Start the descriptor operation (write to CHCR) Descriptor enable (CHCR.DPE = 1) TE interrupt enable (CHCR.IE = 1) DSE interrupt disable (CHCR.DSIE = 0) After reading descriptor memory (CHCR.DPB = 1)</p>
2	TE clear	<p>Check the interrupt status (read CHSTA) Disable DMA transfer. (CHCR.DE = 0) Clear the transfer end flag. (CHFCR.TEC = 1)</p>

(c) Example 3: Descriptor Step End Interrupt

The DMAC transfers the descriptors in order, and generates an interrupt after the specified descriptor is complete.

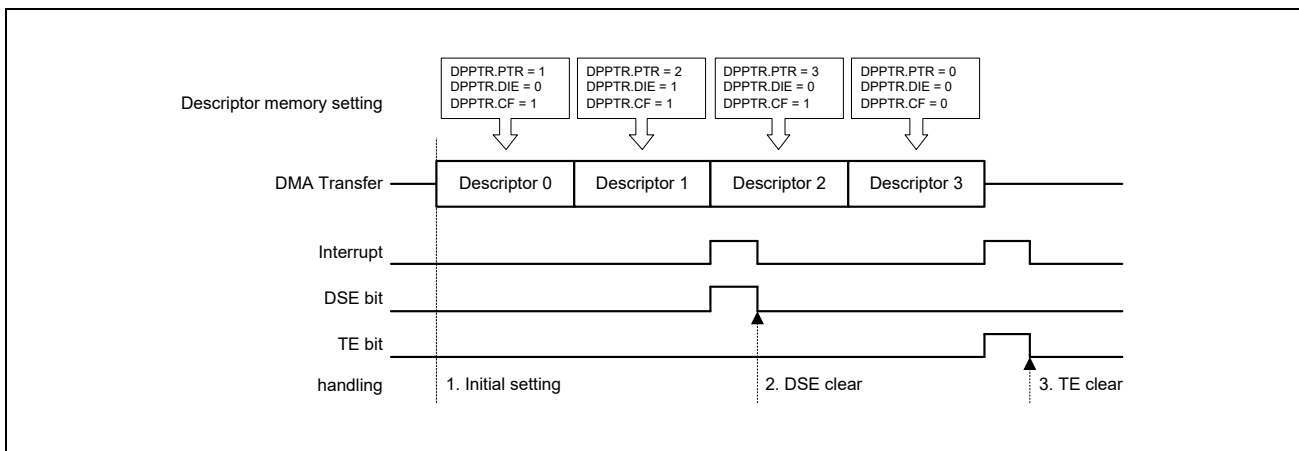


Figure 7.20 Descriptor Step End Interrupt (Example 3)

No.	Handling	Detail
1	Initial setting	<p>Specify the register to be updated. Write to the DPCR register. Descriptors 0 to 3 are written to the descriptor memory.</p> <p>Descriptor[0].DPPTR CF = 1, DIE = 0, PTR = Descriptor 1 pointer</p> <p>Descriptor[1].DPPTR CF = 1, DIE = 1, PTR = Descriptor 2 pointer</p> <p>Descriptor[2].DPPTR CF = 1, DIE = 0, PTR = Descriptor 3 pointer</p> <p>Descriptor[3].DPPTR CF = 0, DIE = 0, PTR = Any value</p> <p>The first descriptor pointer is written to the DPPTR register. DPPTR.PTR = 0</p> <p>Start the descriptor operation (write to CHCR) Descriptor enable (CHCR.DPE = 1) TE interrupt enable (CHCR.IE = 1) DSE interrupt enable (CHCR.DSIE = 1) After reading descriptor memory (CHCR.DPB = 1)</p>
2	DSE clear	<p>Check the interrupt status (read CHSTA) Clear Descriptor step end flag. (CHFCR.DSEC = 1)</p>
3	TE clear	<p>Check the interrupt status (read CHSTA) Disable DMA transfer (CHCR.DE = 0) Clear the transfer end flag (CHFCR.TEC = 1)</p>

(d) Example 4: Double Buffer Operation

Example of double buffer operation using DSE interrupt.

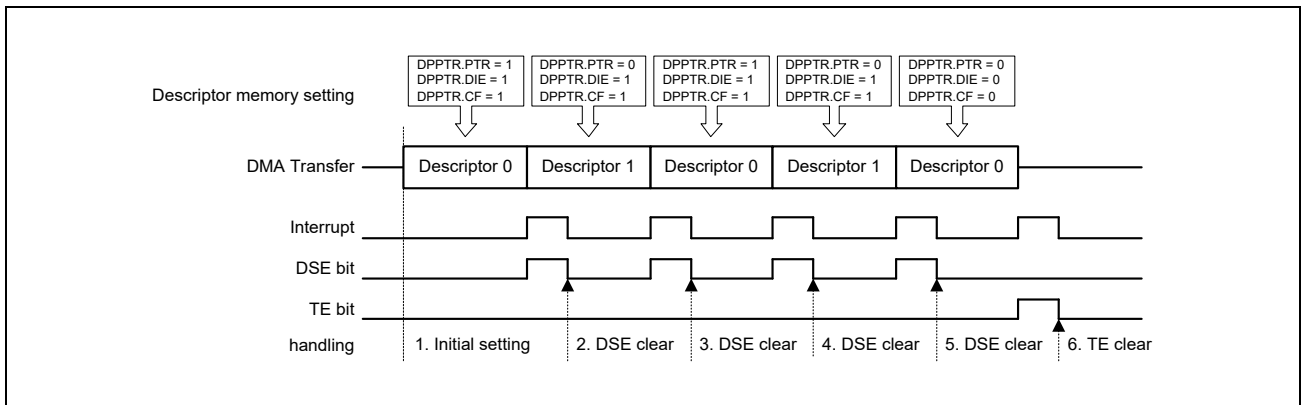


Figure 7.21 Double Buffer Operation (Example 4)

No.	Handling	Detail
1	Initial setting	Specify the register to be updated. Write to the DPCR register. Descriptor 0 to 1 is written in Descriptor Memory Descriptor[0].DPPTR CF = 1, DIE = 1, PTR = Descriptor 1 pointer Descriptor[1].DPPTR CF = 1, DIE = 1, PTR = Descriptor 0 pointer The first descriptor pointer is written to the DPPTR register. DPPTR.PTR = 0 Start the descriptor operation (write to CHCR) Descriptor enable (CHCR.DPE = 1) TE interrupt enable (CHCR.IE = 1) DSE interrupt enable (CHCR.DSIE = 1) After reading descriptor memory (CHCR.DPB = 1)
2, 3	DSE clear	Check the interrupt status (read CHSTA) Clear Descriptor step end flag. (CHFCR.DSEC = 1)
4	DSE clear	Check the interrupt status (read CHSTA) Clear continuation flag. (Descriptor[0].DPPTR.CF = 0) Clear Descriptor step end flag. (CHFCR.DSEC = 1)
5	DSE clear	Check the interrupt status (read CHSTA) Clear Descriptor step end flag. (CHFCR.DSEC = 1)
6	TE clear	Check the interrupt status (read CHSTA) Disable DMA transfer. (CHCR.DE = 0) Clear the transfer end flag. (CHFCR.TEC = 1)

(e) Example 5: Double Buffer Operation with Interrupt Processing Delay

When the next DSE interrupt occurs without clearing the DSE interrupt and without reading the next setting from descriptor memory, transmission is suspended. When the DSE interrupt is cleared after writing the next setting to the descriptor memory, transmission resumes.

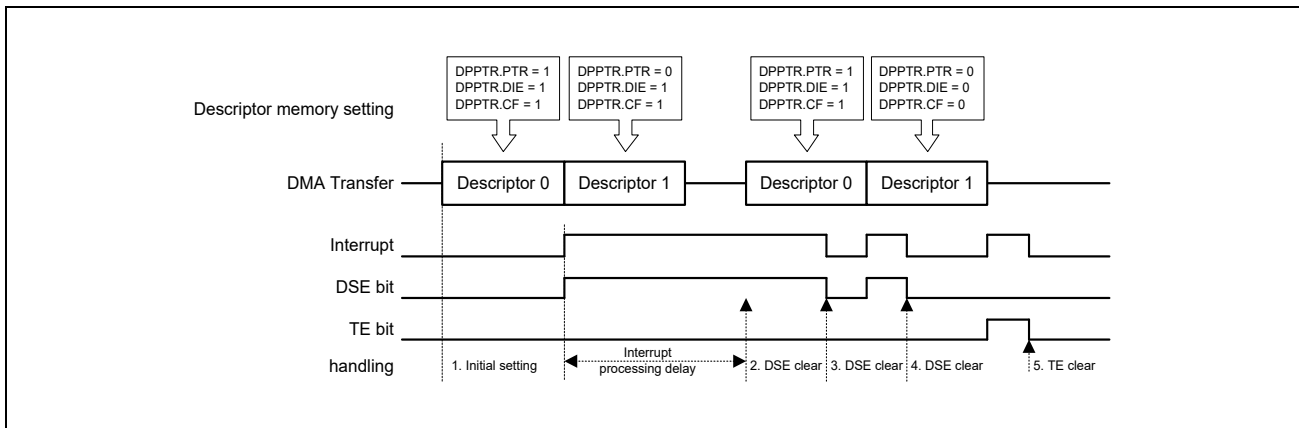


Figure 7.22 Double Buffer Operation with Interrupt Processing Delay (Example 5)

No.	Handling	Detail
1	Initial setting	Specify the register to be updated. Write to the DPCR register. Descriptor 0 to 1 is written in Descriptor Memory Descriptor[0].DPPTR CF = 1, DIE = 1, PTR = Descriptor 1 pointer Descriptor[1].DPPTR CF = 1, DIE = 1, PTR = Descriptor 0 pointer The first descriptor pointer is written to the DPPTR register. DPPTR.PTR = 0 Start the descriptor operation (write to CHCR) Descriptor enable (CHCR.DPE = 1) TE interrupt enable (CHCR.IE = 1) DSE interrupt enable (CHCR.DSIE = 1) After reading descriptor memory (CHCR.DPB = 1)
2	DSE clear	Check the interrupt status (read CHSTA) Clear Descriptor step end flag. (CHFCR.DSEC = 1)
3	DSE clear	Check the interrupt status (read CHSTA) Clear continuation flag. (Descriptor[1].DPPTR.CF = 0) Clear Descriptor step end flag. (CHFCR.DSEC = 1)
4	DSE clear	Check the interrupt status (read CHSTA) Clear Descriptor step end flag. (CHFCR.DSEC = 1)
5	TE clear	Check the interrupt status (read CHSTA) Disable DMA transfer. (CHCR.DE = 0) Clear the transfer end flag. (CHFCR.TEC = 1)

(f) Example 6: Repeat Operation

The continuation flag remains set and DMA transfer is repeated.

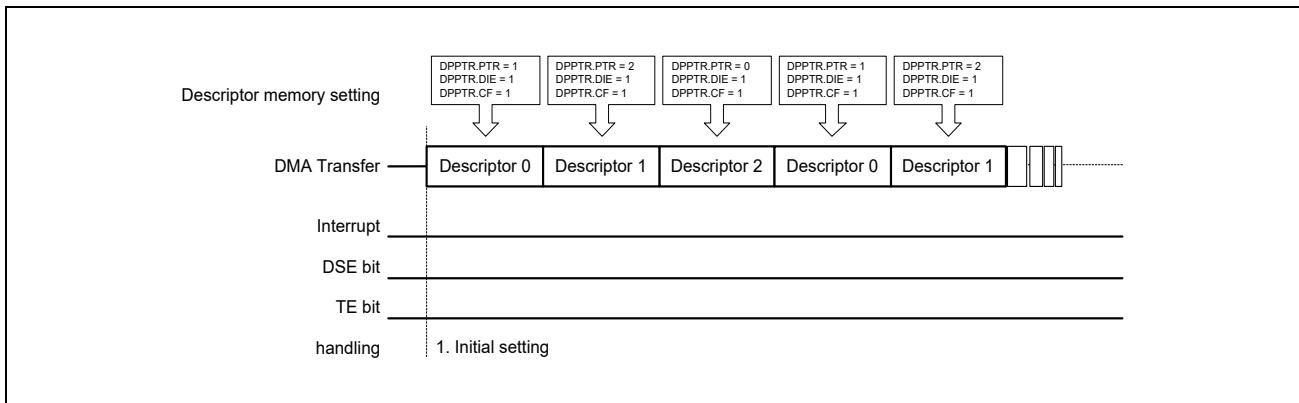


Figure 7.23 Repeat Operation (Example 6)

No.	Handling	Detail
1	Initial setting	<p>Specify the register to be updated. Write to the DPCR register.</p> <p>Descriptors 0 to 2 are written to the descriptor memory.</p> <p>Descriptor[0].DPPTTR CF = 1, DIE = 1, PTR = Descriptor 1 pointer</p> <p>Descriptor[1].DPPTTR CF = 1, DIE = 1, PTR = Descriptor 2 pointer</p> <p>Descriptor[2].DPPTTR CF = 1, DIE = 1, PTR = Descriptor 0 pointer</p> <p>The first descriptor pointer is written to the DPPTTR register. DPPTTR.PTR = 0</p> <p>Start the descriptor operation (write to CHCR) Descriptor enable (CHCR.DPE = 1) TE interrupt enable (CHCR.IE = 0) DSE interrupt disable (CHCR.DSIE = 0) After reading descriptor memory (CHCR.DPB = 1)</p>

(g) Example 7: Hardware Request Operation

When hardware transfer request mode is set, the descriptor operation is executed by a hardware request.

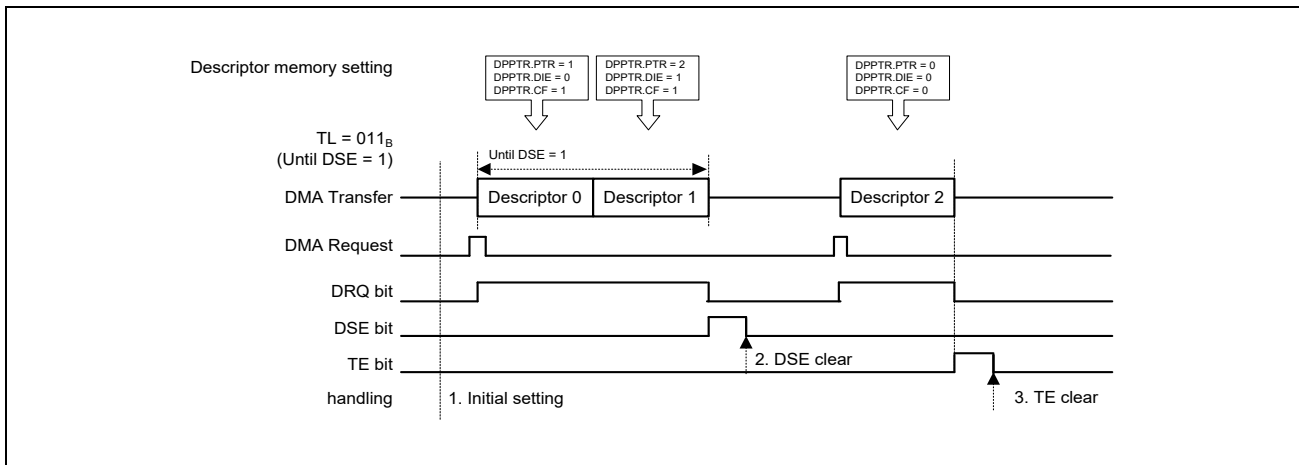


Figure 7.24 Hardware Request Operation (Example 7)

No.	Handling	Detail
1	Initial setting	<p>Specify the register to be updated.</p> <p>Write to the DPCR register.</p> <p>Descriptors 0 to 3 are written to the descriptor memory.</p> <p>Descriptor[0].DPPTR CF = 1, DIE = 0, PTR = Descriptor 1 pointer</p> <p>Descriptor[1].DPPTR CF = 1, DIE = 1, PTR = Descriptor 2 pointer</p> <p>Descriptor[2].DPPTR CF = 0, DIE = 0, PTR = Any value</p> <p>The first descriptor pointer is written to the DPPTR register. DPPTR.PTR = 0</p> <p>Setting hardware transfer request in each Descriptor n (n = 0~2)</p> <p>Hardware request mode (Descriptor[n].DMAjTMR_n.TRS = 1)</p> <p>Transfer limit by DSE is asserted (Descriptor[n].DMAjRS_n.TL = 3'b011)</p> <p>Start the descriptor operation (write to CHCR)</p> <p>Descriptor enable (CHCR.DPE = 1)</p> <p>TE interrupt enable (CHCR.IE = 0)</p> <p>DSE interrupt disable (CHCR.DSIE = 0)</p> <p>After reading descriptor memory (CHCR.DPB = 1)</p>
2	DSE clear	<p>Check the interrupt status (read CHSTA)</p> <p>Clear Descriptor step end flag. (CHFCR.DSEC = 1)</p>
3	TE clear	<p>Check the interrupt status (read CHSTA)</p> <p>Disable DMA transfer. (CHCR.DE = 0)</p> <p>Clear the transfer end flag. (CHFCR.TEC = 1)</p>

(h) Example 8: Hardware Request Operation with Different Request Source Settings

If a hardware request source setting is different in chained descriptors, the generated DMA request is automatically cleared when the descriptor memory setting is loaded.

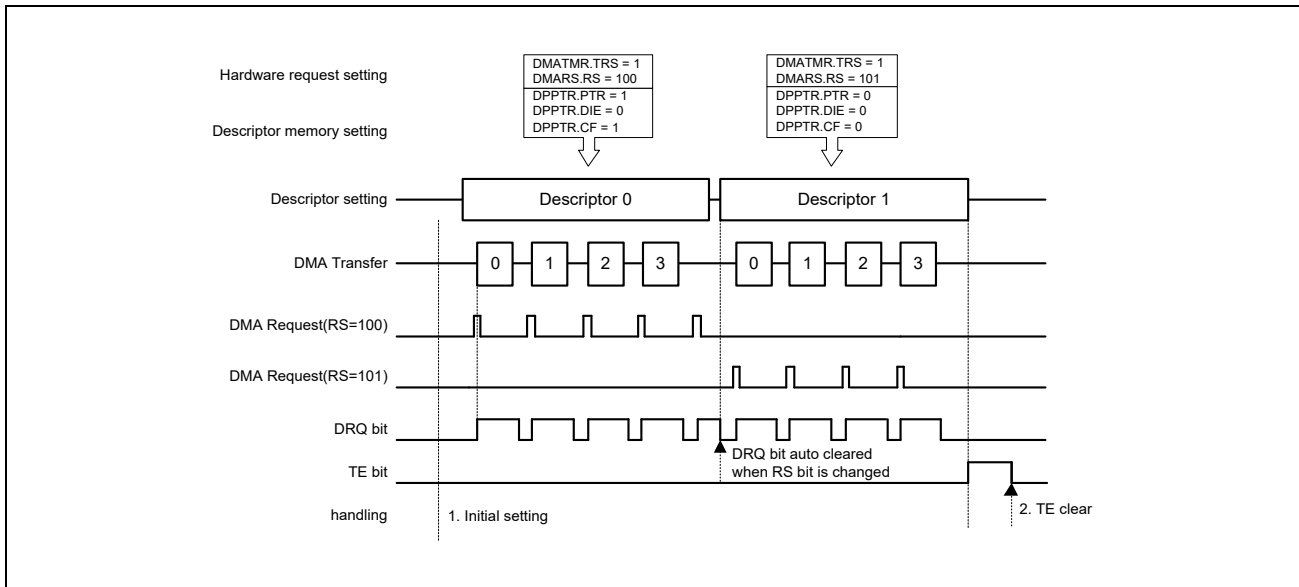


Figure 7.25 Hardware Request Operation with Different Request Source Settings (Example 8)

No.	Handling	Detail
1	Initial setting	<p>Specify the register to be updated. Write to the DPCR register. Descriptors 0 to 3 are written to the descriptor memory. Descriptor[0].DPPTR CF = 1, DIE = 0, PTR = Descriptor 1 pointer Descriptor[1].DPPTR CF = 0, DIE = 0, PTR = Any value The first descriptor pointer is written to the DPPTR register. DPPTR.PTR = 0 Setting hardware transfer request in each Descriptor Descriptor[0].DMAjTMR_n TRS = 1 Descriptor[0].DMAjRS_n RS = 100 (hardware request resource channel) Descriptor[1].DMAjTMR_n TRS = 1 Descriptor[1].DMAjRS_n RS = 101 (hardware request resource channel) Start the descriptor operation (write to CHCR) Descriptor enable (CHCR.DPE = 1) TE interrupt enable (CHCR.IE = 0) DSE interrupt disable (CHCR.DSIE = 0) After reading descriptor memory (CHCR.DPB = 1)</p>
3	TE clear	<p>Check the interrupt status (read CHSTA) Disable DMA transfer. (CHCR.DE = 0) Clear the transfer end flag. (CHFCR.TEC = 1)</p>

(5) Stopping a Descriptor Operation

Three methods can be used to stop a descriptor operation.

Method 1: Descriptor updating is stopped after completion of the specific DMA transfer.

The specific CF bit in DMAjDPPTR_n in the descriptor memory is set to 0.

The TE interrupt is issued after the end of the specified DMA transfer.

Method 2: Descriptor updating is stopped after the current DMA transfer is complete.

The DPE bit in the DMAjCHCR_n register is set to 0 by writing 1 to the DPEC bit in the DMAjCHFCR_n register. The TE interrupt is issued after the end of the current DMA transfer even if the CF bit is set to 1.

It is not recommended to use the CF bit of the DMAjDPPTR_n register to stop DMA transfer. This bit is updated by the descriptor function.

Method 3: Immediate stop without completing the current DMA transfer.

See **Section 7.5(2), DMA Transfer Suspension** and **Section 7.5(3), Stopping and Restarting DMA Transfer**.

7.4.7 Transfer Flow

Write the transfer condition settings to the DMA source address register (DMAjSAR_n), DMA destination address register (DMAjDAR_n), DMA transfer size register (DMAjTSR_n), DMA transfer mode register (DMAjTMR_n), DMA Channel control register (DMAjCHCR_n), DMA resource select register (DMAjRS_n) and DMA operation register (DMAjOR). The DMAC transfers data in following order.

(1) Auto Request Mode

- A transfer request occurs and the DMAC confirms whether transfer is enabled (DE = 1, DME = 1, TE = 0, CAE = 0). Transfer starts automatically.
- If a transfer request is generated, one transfer unit (specified by STS [3:0] or DTS[3:0]) will be transferred. DMAjTSR_n will be decremented by the transfer unit every time DMA transfer completes.
- When the specified number of transfers are complete (the value of DMAjTSR_n and DMAjTSRB_n is 0), transfer ends normally. At this time if the IE bit of DMAjCHSTA_n is set to 1, a TE interrupt will be issued to the CPU.
- Transfer is terminated if an address error occurs in DMA transfer. In addition, transfer is terminated even if the DE bit of DMAjCHCR_n or the DME bit of DMAjOR is set to 0.

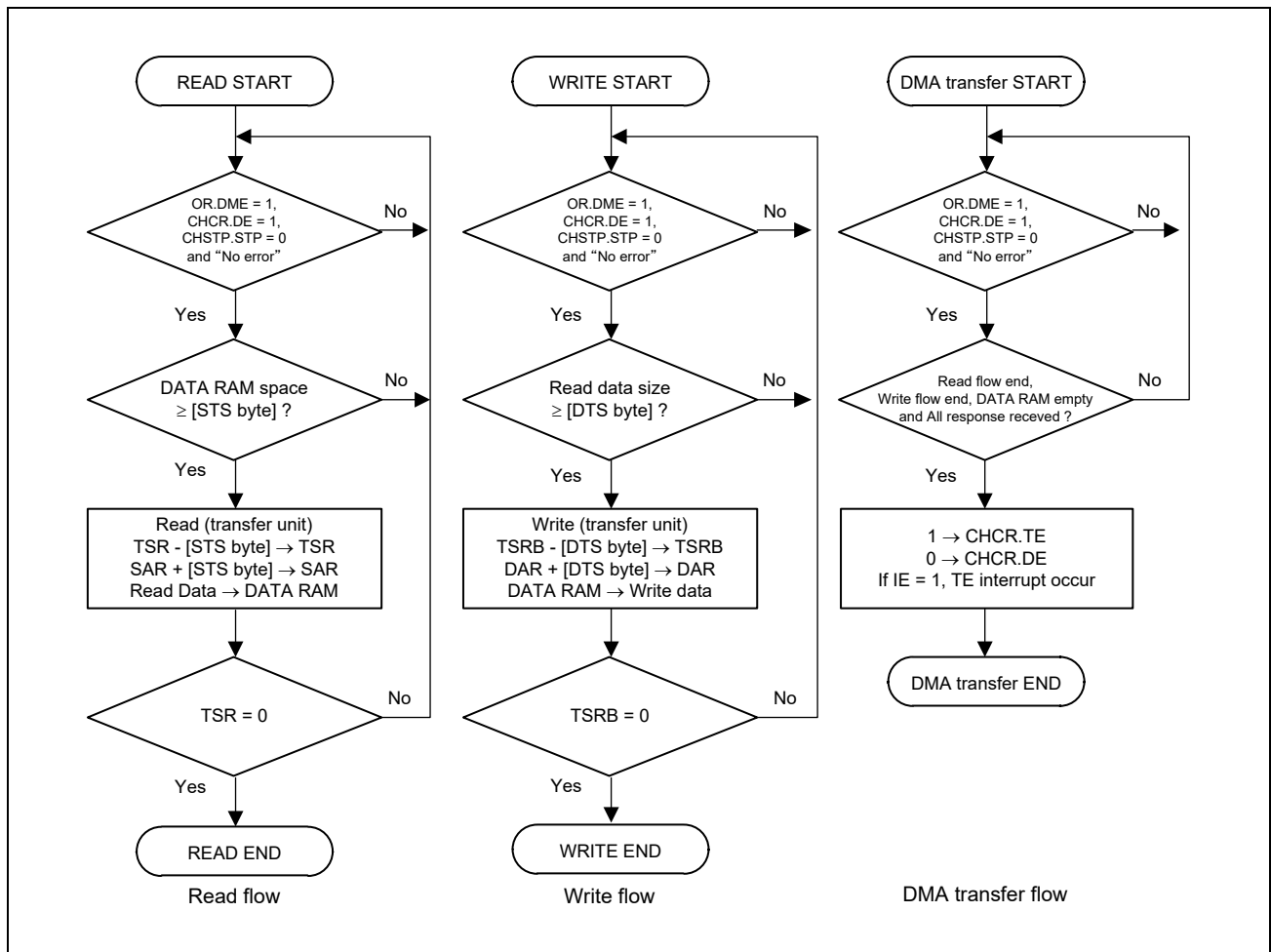


Figure 7.26 Transfer Flow in Auto Request Mode

(2) Hardware Request Mode

- **Figure 7.27** shows the flow in hardware request mode.
- A hardware request can be received when the DMAC confirms whether transfer is enabled (DE = 1, DME = 1, TE = 0, CAE = 0).
- If PLE is set to 1 and FPT is set to 0, pre-loading data from the source memory to the data RAM of sDMAC starts automatically. DMAjTSR_n will be decremented by the transfer unit every time pre-load transfer completes. If FPT is set to 1, pre-loading occurs after the first hardware transfer request is generated and the internal status register (FPS) is set to 0.
- When a hardware transfer request is generated, one transfer unit (specified by TL[2:0]) will be transferred. DMAjTSRB_n will be incremented by the transfer unit every time transfer completes.
- When the specified number of transfers are complete (the value of DMAjTSR_n and DMAjTSRB_n is 0), transfer ends normally. At this time if the IE bit of DMAjCHSTA_n is set to 1, a TE interrupt will be issued to the CPU.
- Transfer is terminated if an address error occurs in DMA transfer. In addition, transfer is terminated even if the DE bit of DMAjCHCR_n or the DME bit of DMAjOR is set to 0.

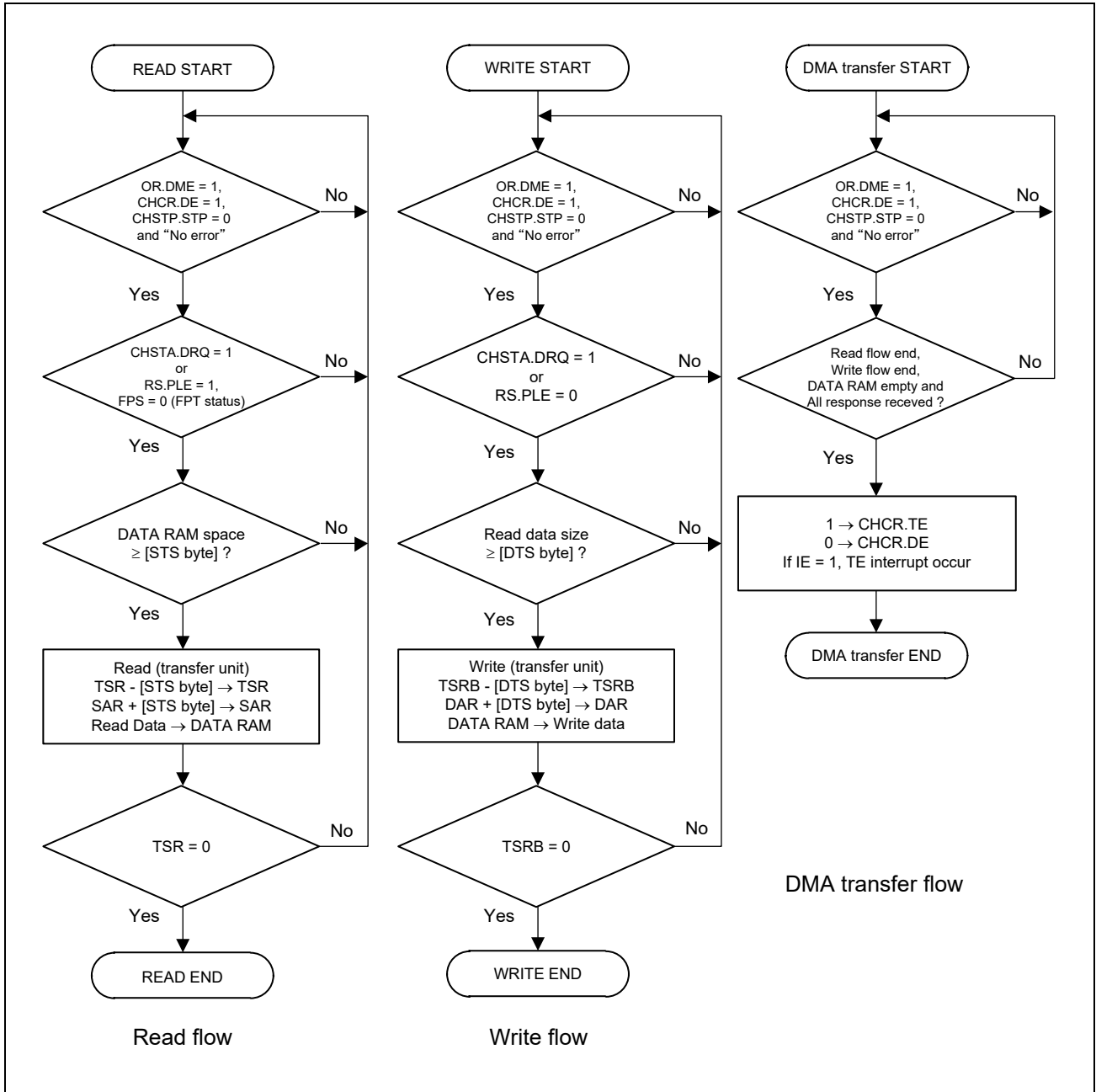


Figure 7.27 Transfer Flow in Hardware Request Mode

7.4.8 Performance

The DMA controller operates on a CLK_HBUS. Read and write transaction requests are issued in parallel on both system interconnect read and write interfaces to enable flyby transfers. Transaction requests are issued without waiting for a reply. So multiple outstanding requests can be in flight on the bus. This is used to hide latencies on the path from the DMA to the slave. The DMA supports 16 outstanding transactions per system interconnect interface (16 reads and 16 writes).

All channels operate in parallel. Arbitration on the read/write interfaces is performed as defined in the operation register DMAJOR. The bus data width is 64 bits (8 bytes). If the unit transfer size (STS/DTS) is larger than 8 bytes, the transaction requests multiple replies in sequence (also called “burst”). The maximum burst size is 8, resulting in a maximum transfer unit size of 64 bytes.

The maximum number of outstanding requests in each DMA channel changes depending on the unit transfer size (STS/DTS). Each DMA channel can use 128 bytes in data RAM as outstanding transfer data buffer. If the unit transfer size (STS/DTS) in a DMA channel is 16 bytes or more, the number of outstanding requests in the DMA channel is 8 or less and the maximum number of outstanding request is 16 or less. Each DMA channel can issue a new transaction request every three clock cycles at best. The frequency at which each DMA channel issues a new transaction request changes depending on the number of outstanding requests in the DMA channel or the number of channels using DMA transfer. If the unit transfer size (STS/DTS) in a DMA channel is 64 bytes, the number of outstanding requests in the DMA channel is 2. In this case, if DMA transfer is operated with one DMA channel and a transaction request is issued twice, a subsequent read transaction request is not issued before previous read and write transaction requests are processed, and the DMA channel cannot issue a new transaction request every three clock cycles.

To utilize the read and write data channels every cycle, two approaches can be used:

- Use three DMA channels. Each channel transfers one third of the required data. The DMAC issues a new request transaction every cycle on the read and write interfaces in parallel.
- Use one DMA channel with a STS/DTS value of 8 bytes. The read/write transaction requests are issued every three cycles.

Both approaches fully utilize the read and write data channels if the data RAM is sufficiently utilized.

7.5 Usage Notes

Pay attention to the following notes when using the DMAC.

(1) Address Errors

When a DMA address error occurs, set the register of the erroneous channel again after DMA transfer suspend shown in **Section 7.5(2), DMA Transfer Suspension** and then start transfer. Address errors are caused by the factors indicated in **Table 7.27**.

Table 7.27 Address Error Factors

Factor	Description
DMA transfer error	The transfer source or transfer destination is invalid space.
	The transfer source or transfer destination is a GUARD error.
Prohibited setting error	DMA source transaction size setting is prohibited. (DMATMR.STS > 0110: 64 bytes)
	DMA destination transaction size setting is prohibited. (DMATMR.DTS > 0110: 64 bytes)
	DMA transfer count is zero in hardware request mode (DMARS.TC = 0 when DMARS.TL = 000, 001)

(2) DMA Transfer Suspension

When suspension DMA transfer, clear the DE bit in the DMA channel control register (DMAjCHCR_n) by using the DEC bit in the DMA channel flag clear register (DMAjCHFCR_n) to disable DMA transfer. After receiving the entire response packet, clear the BUSY bit in the DMA channel status register (DMAjCHSTA_n) to 0. There is a possibility that a TE/DSE interrupt may occur to asynchronous timing after suspension transfer, so it is necessary to assume the following case and take appropriate measures:

- After DMA transfer is suspended, the TE/DSE bit is set to 1 and the BUSY bit is cleared. If the IE/DSIE bit is set to 1 at that time, the DMA interrupt is asserted.

NOTE

If DMA transfer is initialized while the last transfer is executing, this causes a TE/DSE generation delay. In this case, perform a dummy read.

Figure 7.28 shows a processing example when DMA transfer is suspended.

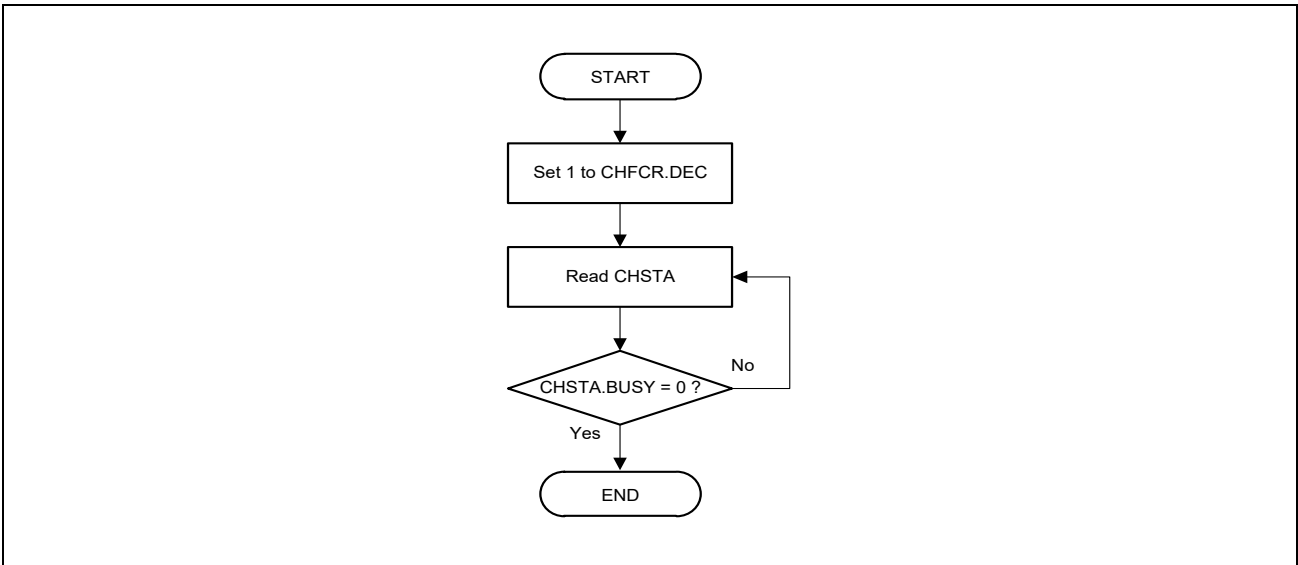


Figure 7.28 Example of DMAC Transfer Suspension Processing

(3) Stopping and Restarting DMA Transfer

To stop DMA transfer temporarily, write 1 to the STP bit in the DMA channel stop register (DMA_jCHSTP_n). Writing to the STP bit stops DMA transfer requests immediately. After receiving the entire response packet, the BUSY bit is cleared to 0. When restarting, set the STP bit to 0.

Figure 7.29 shows a processing example when stopping and restarting DMA transfer.

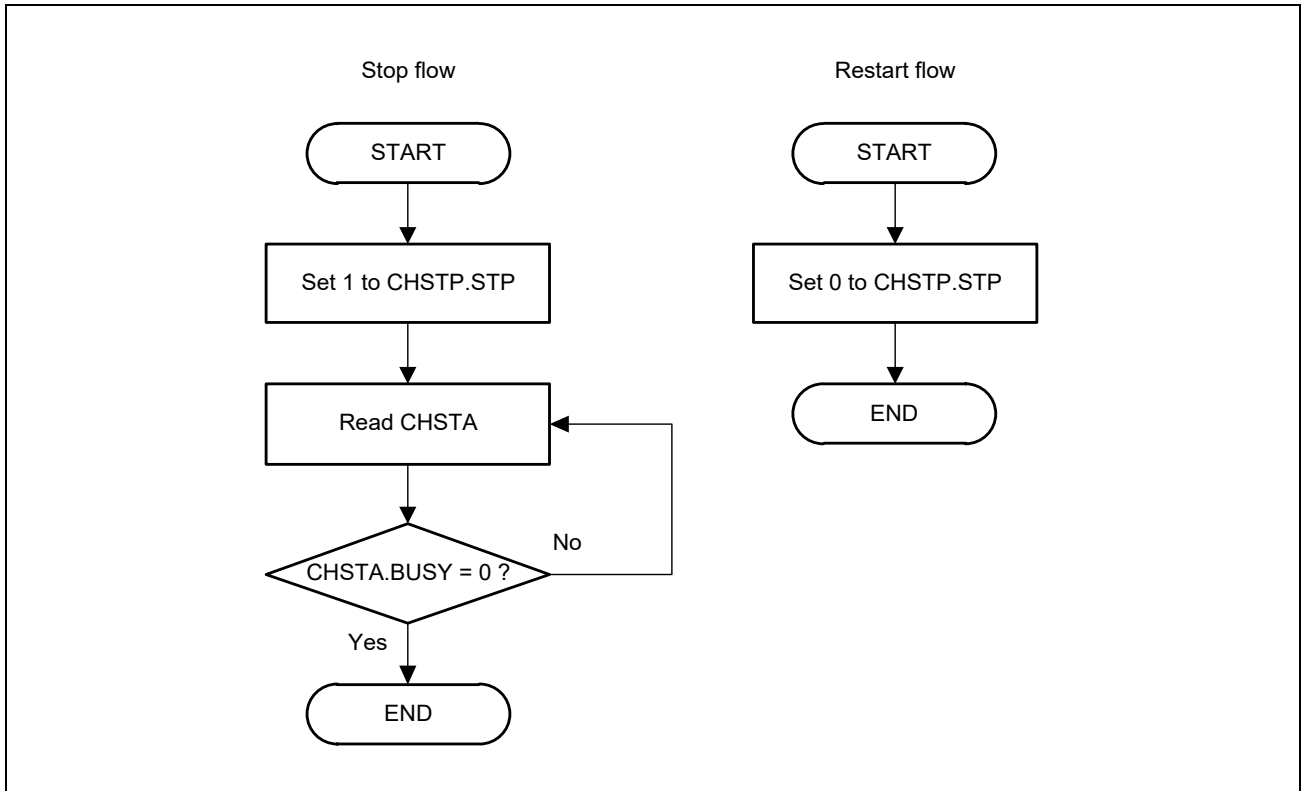


Figure 7.29 Example of DMA Transfer Stop and Restart Processing

(4) Impact of Transfer Unit Size on Addresses and Offsets

The general recommendation is to set the base addresses (SAR, DAR) and address increments (GIAI, GOAI, SIAI, SOAI) to a multiple of the related transfer unit size (STS, DTS). If this recommendation is not followed, byte transfers are used instead. This reduces the performance significantly.

7.6 Reliability Function

7.6.1 Overview

In this product, the DMAC is a resource used by multiple masters (CPU0 or CPU1). In order for the DMAC to support a multi-core configuration, the following reliability functions are offered.

- Register access protection function
Refer to **Section 38, Functional Safety**.
- Master information inheritance function

7.6.2 Master Information Inheritance Function

In this product, DMA access inherits master information is set by DMAjCM_n register.

The master information that is output from the DMAC is as shown in **Table 7.28** for access guard.

Table 7.28 Master Information That Is Output from DMAC

Meaning	Value that is output from DMAC
UM	Same as the UM bit value in the channel master setting register
SEC	Non secure
SPID	Same as the SPID bit value in the channel master setting register

7.7 Setting up DMA Transfer

7.7.1 Overview of Setting up DMA Transfer

Table 7.29 Overview of Setting up DMA Transfer

No.	Description	Register		Necessity of the Setting
1	Transfer request group selection setting	DMACSELj_m	DMA transfer request group selection register m	Mandatory (if using hardware transfer request mode)
2	Overall DMA operation setting	DMAjCM_n	DMA channel master setting register	Mandatory
3	Channel setting	DMAjSAR_n	DMA source address register	Mandatory
4		DMAjDAR_n	DMA destination address register	Mandatory
5		DMAjTSR_n	DMA transfer size register	Mandatory
6		DMAjTMR_n	DMA transfer mode register	Mandatory
7		DMAjRS_n	DMA resource select register	Mandatory (if using hardware transfer request mode)
8	Status clear	DMAjCHFCR_n	DMA channel flag clear register	Mandatory
9	Channel operation enable	DMAjCHCR_n	DMA channel control register	Mandatory

NOTE

If a transfer request is used to change the transfer request group, it is necessary to set up the registers from the beginning without configuring an overall DMA operation setting.

7.7.2 Setting up the Transfer Request Group Selection

Set up the transfer request group selection before using the sDMAC. The following registers must be set up to configure a transfer request group.

- sDMACj Transfer Request Group Selection Register m (DMACSELj_m) (j = 0 to 1, m = 0 to 15)
This register configures the transfer request group selection

7.7.3 Setting up the Overall DMA Operation

Set up the overall DMA operation before you start using DMA.

The following register must be set up to configure the overall DMA operation.

DMAC channel master setting register (DMAjCM_n)

This register configures the channel assignment. (For details, see **Section 7.6, Reliability Function.**)

If the DMAC channel master setting registers are not properly set, DMA channel setting and DMA transfer cannot be executed properly.

7.7.4 Setting up the DMA Channel Setting

The DMA channel setting defines the transfer information and transfer source for each DMAC channel.

To configure the DMA channel setting, the master allowed to access each channel needs to set up the channel registers.

Follow the procedure below to set up the DMAC channel setting and use the DMAC.

(1) Disabling the DMAC channel operation

If the channel operation enable bit (DE) in the DMA channel control register (DMAjCHCR_n) is set, channel setting up registers must not to be set up. Wait for the previous DMA transfer to finish or be suspended, and then disable the DMAC channel operation.

(2) Setting up the transfer information

When setting up the transfer information for the DMAC, the following registers need to be set up.

- DMA source address register (DMAjSAR_n)
- DMA destination address register (DMAjDAR_n)
- DMA transfer size register (DMAjTSR_n)
- DMA transfer mode register (DMAjTMR_n)

(3) Setting up the DMA transfer request

While setting the transfer information, set the DMA transfer request select (TRS) bit in the DMA transfer mode register (DMAjTMR_n) to define whether a hardware or software DMA transfer request is used.

The hardware and software DMA transfer request cannot be used for the same channel at the same time.

If the hardware DMA transfer request is used, select one source to be used as the hardware DMA transfer request out of the 256 available hardware DMA transfer sources using the resource selection (RS) bit in the DMA resource select register (DMAjRS_n).

The DMA channel may retain a hardware DMA transfer request that came before the hardware DMA transfer source is selected. Clear the hardware DMA transfer request (DMACHSTA_n.DRQ) and overflow flag (DMACHSTA_n.OVF) by using the DMA channel flag clear register (DMAjCHFCR_n) before enabling channel operation.

If the software DMA transfer request is used, disable the hardware DMA transfer request select (TRS) bit in the DMA transfer mode register (DMAjTMR_n).

(4) Clearing the transfer status

The DMA channel status register (DMAjCHSTA_n) may retain the result of the previous DMA transfer. Clear the flags in the DMA transfer status register using the DMA channel flag clear register (DMAjCHFCR_n).

(5) Enabling the DMAC channel operation

Set the DMA channel operation enable bit (DE) in the DMA channel control register (DMAjCHCR_n) to enable the channel operation. After the channel operation enable bit is set, the DMAC can accept a DMA transfer request and start DMA transfer.

Section 8 DTS Controller

This section provides a generic description of the Data Transfer Service (DTS) controller. The first part of this section contains information about product properties such as number of units and register base addresses. The second part of this section describes functions and registers of DTS.

8.1 Features of DTS

8.1.1 Number of Units and Channels

This product has the following number of DTS units.

Table 8.1 Number of Units

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pin
Number of Units	1	1
Number of Channels	128	128
Name	DTSCNT	DTSCNT

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Units	1	1
Number of Channels	128	128
Name	DTSCNT	DTSCNT

Table 8.2 Definition of Indexes

Index	Meaning
nnn	The channel number is identified by the index "nnn". (nnn = 0 to 127)
m	The individual trigger group selection registers are identified by the index "m" (m = 0 to 15).

8.1.2 Register Base Addresses

The DTS base addresses are listed in **Table 8.3**.

The DTS register addresses are specified as offsets from the base addresses.

Table 8.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<DTS_base>	FFF8 8000 _H	Peripheral Group 0
<DMATRGSSEL_base>	FF09 0100 _H	Peripheral Group 9

8.1.3 Clock Supplies

The DTS clock supplies are shown in the following table.

Table 8.4 Clock Supplies

Unit Name	Clock for the Unit	Supply Clock Name
DTSCNT	clk	CLK_HBUS
DMATRGSEL	PCLK	CLK_HBUS

8.1.4 Interrupt Requests

The DTS interrupt requests are listed in the following table.

Table 8.5 Interrupt Requests

Interrupt symbol name	Unit Interrupt Signal	Outline	Interrupt Number
INTDTS31TO0	INTDTS[31:0]	DTS ch31-0 transfer end	37
INTDTS63TO32	INTDTS[63:32]	DTS ch63-32 transfer end	38
INTDTS95TO64	INTDTS[95:64]	DTS ch95-64 transfer end	39
INTDTS127TO96	INTDTS[127:96]	DTS ch127-96 transfer end	40
INTDTSCT31TO0	INTDTSCT[31:0]	DTS ch31-0 transfer count match	41
INTDTSCT63TO32	INTDTSCT[63:32]	DTS ch63-32 transfer count match	42
INTDTSCT95TO64	INTDTSCT[95:64]	DTS ch95-64 transfer count match	43
INTDTSCT127TO96	INTDTSCT[127:96]	DTS ch127-96 transfer count match	44
INTDTSERR	DTSERR	DTS transfer error	30

8.1.5 DTS Transfer Requests for DTS

The DTS transfer request table lists all DTS transfer request factors for DTS. The DTS transfer request group for DTS can be selected by the DTSSSELm register.

8.1.6 Reset Sources

DTS reset sources are listed in the following table. DTS is initialized by the following reset sources.

Table 8.6 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
DTSCNT	All registers	✓	✓	✓	✓	—	—
DTSSSEL	All registers	✓	✓	✓	✓	—	—

8.1.7 External Input/Output Signals

This module has no signals mapped to external input/output pins.

8.2 Overview

8.2.1 Functional Overview

The Data Transfer Service (DTS) is a DMA controller used to access data without going through the CPU.

DTS stores transfer information in a dedicated RAM (DTSRAM).

Table 8.7 shows the configuration of DTS.

Table 8.7 Configuration of DTS

Module Name	Number of Units	Number of Channels
DTSCNT	1	128 ch

Functions of DTSFSL are arbitration of incoming the DTS requests, triggering the DTS controller and monitoring ongoing DTS transfers. DTSFSL can handle 128 DTS transfer sources.

The address space that can be used for DTS transfer is a 4 GB address space represented by a 32-bit address. For information about which resource is assigned to a particular area in the 4 GB address space and which area is accessible from DTS, refer to **Section 4, Address Map**.

The DTS is connected to System Bus directly. If a conflict occurs with another bus master in System Bus, the transfer is arbitrated by round-robin arbitration.

8.2.2 Definition of Terms

Table 8.8 shows the terms used in this section.

Table 8.8 List of Term Definitions

Term	Meaning
DTS transfer	A term for data transfer carried out by the DTS controller
DTS cycle	A series of actions that consist of reading an amount of data specified by the transfer size (8/16/32/64/128 bits) from the address specified by the source address and writing it to the address specified by the destination address. The first half of the DTS cycle (reading part) is called a read cycle, and the second half (writing part) is called a write cycle.
Hardware DTS transfer source	A trigger for a DTS transfer request issued by an internal peripheral device
Hardware DTS transfer request	A DTS transfer request issued by a hardware DTS transfer source
Software DTS transfer request	A DTS transfer request issued by writing to a register via software
DTS transfer request	A trigger to start DTS transfer with DTS
Transfer information (TI)	The information required for DTS transfer, including the source address, destination address, transfer data size, and transfer count. The transfer information for DTS is referred to as TI.
DTSRAM	RAM used by DTS to store the transfer information
Single transfer	A DTS transfer consisting of one DTS cycle started by one DTS transfer request
Block transfer 1	A DTS transfer consisting of the number of DTS cycles specified by the transfer count in the transfer information, started by one DTS transfer request
Block transfer 2	A DTS transfer consisting of the number of DTS cycles specified by the address reload count in the transfer information, started by one DTS transfer request
Block transfer	A general term for both block transfer 1 and block transfer 2
Last transfer	The DTS cycle carried out when the transfer count in the transfer information is 1
Address reload transfer	The DTS cycle carried out when the address reload count in the transfer information is 1 if reload function 2 is used
Suspend	An action of pausing a DTS transfer during block transfer. The DTS transfers can be resumed after suspension.
Resume	An action of resuming a suspended DTS transfer
Transfer abort	An action of aborting a DTS transfer before it is finished. Aborted DTS transfers cannot be resumed.

8.3 Registers

8.3.1 List of Registers

Table 8.9 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
DTSCNT	DTS Control Register 1	DTSCCTL1	<DTS_base> + 0010 _H	32	—
	DTS Control Register 2	DTSCCTL2	<DTS_base> + 0014 _H	32	—
	DTS Status Register	DTSSTS	<DTS_base> + 0018 _H	32	—
	DTS Error Register	DTSER	<DTS_base> + 0024 _H	32	—
	DTS Channel Priority Setting Register 0	DTSPR0	<DTS_base> + 0060 _H	32	—
	DTS Channel Priority Setting Register 1	DTSPR1	<DTS_base> + 0064 _H	32	—
	DTS Channel Priority Setting Register 2	DTSPR2	<DTS_base> + 0068 _H	32	—
	DTS Channel Priority Setting Register 3	DTSPR3	<DTS_base> + 006C _H	32	—
	DTS Channel Priority Setting Register 4	DTSPR4	<DTS_base> + 0070 _H	32	—
	DTS Channel Priority Setting Register 5	DTSPR5	<DTS_base> + 0074 _H	32	—
	DTS Channel Priority Setting Register 6	DTSPR6	<DTS_base> + 0078 _H	32	—
	DTS Channel Priority Setting Register 7	DTSPR7	<DTS_base> + 007C _H	32	—
	DTS Channel Master Setting Register*1	DTSnmmCM	<DTS_base> + "0200 _H + 4 _H × [DTS channel number] (0200 _H to 03FC _H)"	32	—
	DTS Source Address Register	DTSAmm	<DTS_base> + 1000 _H + 40 _H × [channel number]	32	—
	DTS Destination Address Register	DTDAmm	<DTS_base> + 1004 _H + 40 _H × [channel number]	32	—
	DTS Transfer Count Register	DTTCmm	<DTS_base> + 1008 _H + 40 _H × [channel number]	32	—
	DTS Transfer Control Register	DTTCmm	<DTS_base> + 100C _H + 40 _H × [channel number]	32	—
	DTS Reload Source Address Register	DTRSAmm	<DTS_base> + 1010 _H + 40 _H × [channel number]	32	—
	DTS Reload Destination Address Register	DTRDAmm	<DTS_base> + 1014 _H + 40 _H × [channel number]	32	—
	DTS Reload Transfer Count Register	DTRTCmm	<DTS_base> + 1018 _H + 40 _H × [channel number]	32	—
	DTS Transfer Count Compare Register	DTTCmm	<DTS_base> + 101C _H + 40 _H × [channel number]	32	—
	DTSFSL Operation Setting Register	DTFSLmm	<DTS_base> + 1020 _H + 40 _H × [channel number]	32	—
	DTSFSL Transfer Status Register	DTFSTmm	<DTS_base> + 1024 _H + 40 _H × [channel number]	32	—
	DTSFSL Transfer Request Set Register	DTFSSmm	<DTS_base> + 1028 _H + 40 _H × [channel number]	32	—
	DTSFSL Transfer Status Clear Register	DTFSCmm	<DTS_base> + 102C _H + 40 _H × [channel number]	32	—
	DMATRGSEL	DTS Transfer Request Group Selection Register	DTSELm*2	<DMATRGSEL_base> + 100 _H + (m × 4 _H)	8, 16, 32

Note 1. The [channel number] in the offset address is a number from 0 to 127. The “nnn” in the register symbols is a 3-digit number from 000 to 127.

Note 2. m index

8.3.2 Description of Global Registers

8.3.2.1 DTSCCTL1 — DTS Control Register 1

DTS Control Register 1 (DTSCCTL1) is a 32-bit read/write register that controls the DTS global suspension setting.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTSUS T
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.10 DTSCCTL1 Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved. When read, the value after reset is returned. When writing, write the value after reset.
0	DTSUST	DTS suspend This bit shows whether DTS transfers are suspended. If this bit is set to 1, a DTS suspend is requested. 0: DTS is not suspended 1: DTS is suspended or DTS suspension is requested

8.3.2.2 DTSTCTL2 — DTS Control Register 2

DTS Control Register 2 (DTSTCTL2) is a 32-bit write only register that controls DTS transfer abort.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTSTIT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 8.11 DTSTCTL2 Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved. When writing, write the value after reset.
0	DTSTIT	DTS transfer abort request Setting this bit to 1 while DTS transfer is suspended aborts the suspended DTS transfer. Writing to this bit while DTS transfer is in progress is ignored. When the suspended DTS transfer is aborted, the DTSSTS.DTSACT bit is cleared. This bit is write only.

8.3.2.3 DTSSTS — DTS Status Register

The DTS Status Register (DTSSTS) is a 32-bit read only register that indicates the DTS status.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DTSCY C	DTSACH[6:0]						DTSAC T	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.12 DTSSTS Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved. When read, the value after reset is returned.
8	DTSCYC	DTS cycle execution state This bit shows whether a DTS cycle is in progress in DTS. 0: DTS cycle is not in progress. 1: DTS cycle is in progress.
7 to 1	DTSACH	DTS transfer channel If there is a channel in DTS executing a DTS transfer, the channel number is shown. If there is no channel in DTS executing a DTS transfer, the channel number of the last DTS transfer is shown.
0	DTSACT	DTS transfer status This bit shows whether there is a channel in DTS executing a DTS transfer. 0: There is no channel in DTS executing DTS transfer. 1: There is a channel in DTS executing DTS transfer. If DTS is in the suspended state while there is a channel executing DTS transfer, this bit remains 1. If a DTS transfer abort request is made using the DTSCCTL2.DTSTIT bit, the suspended DTS transfer is aborted, and this bit is cleared to 0. When a DTS transfer error occurs and the DTS transfer is aborted, this bit is cleared.

8.3.2.4 DTSER — DTS Error Register

The DTS Error Register (DTSER) is 32-bit read only register that indicates the error status of DTS.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	DTSERCH[6:0]						—	—	—	—	—	—	DTSERWR	DTSER	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.13 DTSER Register Contents

Bit Position	Bit Name	Function
31 to 15	—	Reserved. When read, the value after reset is returned.
14 to 8	DTSERCH	DTS error channel These bits show the DTS channel number of the smallest channel number that has a DTS transfer error. These bits are read-only and cannot be cleared.
7 to 2	—	Reserved. When read, the value after reset is returned.
1	DTSERWR	DTS transfer error cycle This bit indicates that a DTS transfer error occurred on the channel indicated by DTSERCH[6-0] in the read cycle or the write cycle. These bits are read-only and cannot be cleared. 0: A DTS transfer error occurred in the read cycle. 1: A DTS transfer error occurred in the write cycle.
0	DTSER	DTS transfer error flag This bit shows whether a DTS transfer error is detected in DTS. 0: A DTS transfer error was not detected in all DTS channels. 1: A DTS transfer error was detected in one of the DTS channels.

Note: In order to clear DTSER, a DTS transfer error of all DTS channels need to be cleared.

8.3.2.5 DTSPRn — DTS Channel Priority Setting Register (n = 0 to 7)

The DTS Channel Priority Setting Register (DTSPRn) is a 32-bit read/write register that defines the priorities for each DTS channel. The priorities are configured from 0 to 3, with 0 being the highest.

• DTSPR0

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS15PR[1:0]		DTS14PR[1:0]		DTS13PR[1:0]		DTS12PR[1:0]		DTS11PR[1:0]		DTS10PR[1:0]		DTS9PR[1:0]		DTS8PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS7PR[1:0]		DTS6PR[1:0]		DTS5PR[1:0]		DTS4PR[1:0]		DTS3PR[1:0]		DTS2PR[1:0]		DTS1PR[1:0]		DTS0PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.14 DTSPR0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=0 to 15)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest.

• DTSPR1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS31PR[1:0]		DTS30PR[1:0]		DTS29PR[1:0]		DTS28PR[1:0]		DTS27PR[1:0]		DTS260PR[1:0]		DTS25PR[1:0]		DTS24PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS23PR[1:0]		DTS22PR[1:0]		DTS21PR[1:0]		DTS20PR[1:0]		DTS19PR[1:0]		DTS18PR[1:0]		DTS17PR[1:0]		DTS16PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.15 DTSPR1 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=16 to 31)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest.

• DTSPR2

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS47PR[1:0]		DTS46PR[1:0]		DTS45PR[1:0]		DTS44PR[1:0]		DTS43PR[1:0]		DTS42PR[1:0]		DTS41PR[1:0]		DTS40PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS39PR[1:0]		DTS38PR[1:0]		DTS37PR[1:0]		DTS36PR[1:0]		DTS35PR[1:0]		DTS34PR[1:0]		DTS33PR[1:0]		DTS32PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.16 DTSPR2 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=32 to 47)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest

• DTSPR3

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS63PR[1:0]		DTS62PR[1:0]		DTS61PR[1:0]		DTS60PR[1:0]		DTS59PR[1:0]		DTS58PR[1:0]		DTS57PR[1:0]		DTS56PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS55PR[1:0]		DTS54PR[1:0]		DTS53PR[1:0]		DTS52PR[1:0]		DTS51PR[1:0]		DTS50PR[1:0]		DTS49PR[1:0]		DTS48PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.17 DTSPR3 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=48 to 63)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest.

• DTSPR4

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS79PR[1:0]		DTS78PR[1:0]		DTS77PR[1:0]		DTS76PR[1:0]		DTS75PR[1:0]		DTS74PR[1:0]		DTS73PR[1:0]		DTS72PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS71PR[1:0]		DTS70PR[1:0]		DTS69PR[1:0]		DTS68PR[1:0]		DTS67PR[1:0]		DTS66PR[1:0]		DTS65PR[1:0]		DTS64PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.18 DTSPR4 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=64 to 79)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest.

• DTSPR5

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS95PR[1:0]		DTS94PR[1:0]		DTS93PR[1:0]		DTS92PR[1:0]		DTS91PR[1:0]		DTS90PR[1:0]		DTS89PR[1:0]		DTS88PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS87PR[1:0]		DTS86PR[1:0]		DTS85PR[1:0]		DTS84PR[1:0]		DTS83PR[1:0]		DTS82PR[1:0]		DTS81PR[1:0]		DTS80PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.19 DTSPR5 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=80 to 95)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest.

• DTSPR6

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS111PR[1:0]		DTS110PR[1:0]		DTS109PR[1:0]		DTS108PR[1:0]		DTS107PR[1:0]		DTS106PR[1:0]		DTS105PR[1:0]		DTS104PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS103PR[1:0]		DTS102PR[1:0]		DTS101PR[1:0]		DTS100PR[1:0]		DTS99PR[1:0]		DTS98PR[1:0]		DTS97PR[1:0]		DTS96PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.20 DTSPR6 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=96 to 111)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest.

• DTSPR7

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS127PR[1:0]		DTS126PR[1:0]		DTS125PR[1:0]		DTS124PR[1:0]		DTS123PR[1:0]		DTS122PR[1:0]		DTS121PR[1:0]		DTS120PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS119PR[1:0]		DTS118PR[1:0]		DTS117PR[1:0]		DTS116PR[1:0]		DTS115PR[1:0]		DTS114PR[1:0]		DTS113PR[1:0]		DTS112PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.21 DTSPR7 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=112 to 127)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest.

8.3.2.6 DTSnnnCM — DTS Channel Master Setting Register (nnn = 000 to 127)

The DTS Channel Master Setting Register (DTSnnnCM) is a 32-bit read/write register that defines the channel master configuration for each DTS channel. The contents of this register is used for the master information inheritance function. For more information about the function, see **Section 8.7, Reliability Functions**.

This register is placed in the DTSRAM and the lower 16-bit field of this register is shared with the DTS transfer count compare register (DTTCCnnn), one of DTS channel register. If this register is written, DTTCCnnn register is updated as well.

Value after reset: 0000 0000_H*¹ RAM initialization is executed.

Undefined*¹ RAM initialization is not executed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	CHAIN RESTRICT	CHAIN_SPID[4:0]				CHAIN UM	—	—	SPID[4:0]				UM	—				
Value after reset								Undefined* ¹										
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	CMC[15:0]																	
Value after reset	Undefined* ¹																	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 10.3.6, RAM Initialization** for details.

Table 8.22 DTSnnnCM Register Contents

Bit Position	Bit Name	Function
31	CHAIN_RESTRICT	Chain function restriction setting 0: The chain function is not restricted. A chain from any channel is possible. 1: The chain function is restricted. A chain is permitted only when SPID set by CHAIN_SPID[4:0] is the same as DTSnnnCM.SPID[4:0] of the current channel in the chain and UM set by CHAIN_UM is the same as DTSnnnCM.UM of the current channel in the chain.
30 to 26	CHAIN_SPID[4:0]	Chain permission SPID setting These bits specify the SPID information that permits the chain function.
25	CHAIN_UM	Chain permission UM setting This bit specifies the UM information that permits the chain function.
24 to 23	—	Reserved When read, the value returned is undefined. The write value should be 0.
22 to 18	SPID[4:0]	Channel master SPID setting These bits specify the SPID information used by the master assigned to the channel.
17	UM	Channel master UM setting This bit specifies the UM information of the master assigned to the channel.
16	—	Reserved When read, the value returned is undefined. The write value should be 0.
15 to 0	CMC[15:0]	Transfer count compare In terms of contents, this field is the same as bits[15:0] in Section 8.3.3.2(8), DTTCCnnn — DTS Transfer Count Compare Register .

Note: DTS000CM to DTS127CM configure the channel master information of DTS channels 0 to 127 respectively. For information about the functions this register provides, see **Section 8.7, Reliability Functions**.

The lowest 16 bits of this register are shared with the DTS Transfer Count Compare Register, one of DTS channel registers.

When writing to this register, the DTS Transfer Count Compare Register is updated as well.

Bits 24-23 and 16 of this register are reserved, but these bits are readable and writable. It is recommended to write 0 to these bits and to ignore the value read from these bits.

8.3.3 Description of DTS Channel Registers

8.3.3.1 Transfer Information of DTS (TI)

(1) Structure of TI

The transfer information of DTS is called TI. One TI consists of 8 sets which are called TI-A, TI-B, TI-C, TI-D, TI-E, TI-F, TI-G, and TI-H. Each set consists of a 32-bit word. Eight sets of TI are assigned to each channel.

The following figure shows the structure of TI.

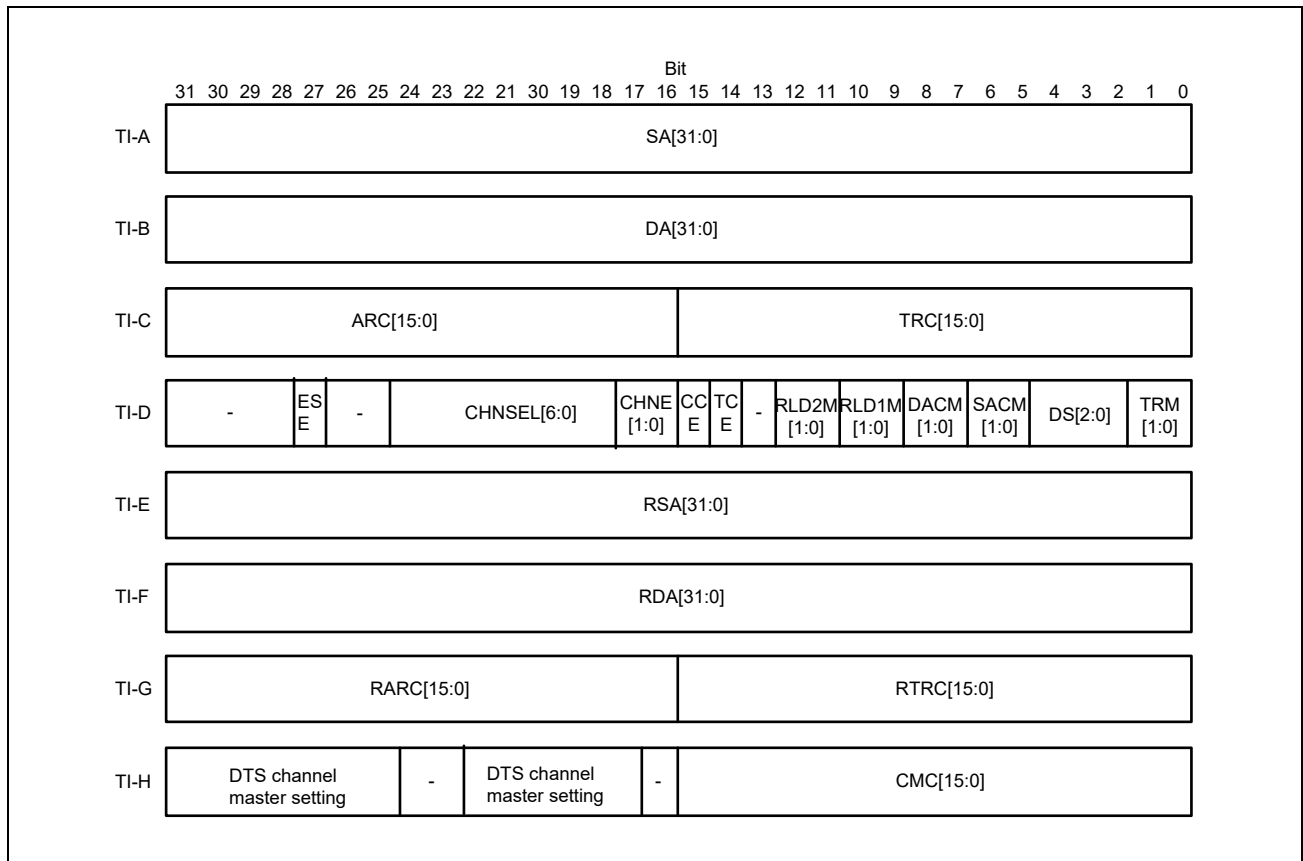


Figure 8.1 Structure of TI

(2) Organization of TI in DTSRAM

The user can indirectly access DTSRAM through the DTS channel registers for each channel and DTS channel master setting registers.

Therefore, direct access to TI structures in DTSRAM is usually not required.

As an exception, when an ECC error occurs while DTSRAM is being accessed, the address at which error occurred is stored in the DTSRAM 1st 1-bit Error Address Register (DR_00SEADR) or the DTSRAM 1st Fatal Error Address Register (DR_00DEADR). It is helpful to understand the address organization of TI in DTSRAM to distinguish which channel/TI has generated the error.

Figure 8.2 shows the address organization of TI in DTSRAM.

DTSRAM address	TI location	
FFC _H	channel 127	TI-H
FF8 _H		TI-G
FF4 _H		TI-F
FF0 _H		TI-E
FEC _H		TI-D
FE8 _H		TI-C
FE4 _H		TI-B
FE0 _H		TI-A
FDC _H	channel 126	TI-H
040 _H	channel 2	TI-A
03C _H	channel 1	TI-H
038 _H		TI-G
034 _H		TI-F
030 _H		TI-E
02C _H		TI-D
028 _H		TI-C
024 _H		TI-B
020 _H		TI-A
01C _H	channel 0	TI-H
018 _H		TI-G
014 _H		TI-F
010 _H		TI-E
00C _H		TI-D
008 _H		TI-C
004 _H		TI-B
000 _H		TI-A

Figure 8.2 Organization of TI in DTSRAM

(3) How to Access TI

TI-A can be accessed through the DTS Source Address Register (DTSAnnn) for each channel.

TI-B can be accessed through the DTS Destination Address Register (DTDAnnn) for each channel.

TI-C can be accessed through the DTS Transfer Count Register (DTTCnnn) for each channel.

TI-D can be accessed through the DTS Transfer Control Register (DTTCTnnn) for each channel.

TI-E can be accessed through the DTS Reload Source Address Register (DTRSAnnn) for each channel.

TI-F can be accessed through the DTS Reload Destination Address Register (DTRDAnnn) for each channel.

TI-G can be accessed through the DTS Reload Transfer Count Register (DTRTCnnn) for each channel.

TI-H can be accessed through either the global Channel Master Setting Register (DTSnnnCM) or the channel specific Transfer Count Compare Register (DTTCCnnn).

(4) Cautions on Accessing TI

The value of the DTS Channel Master Setting Register and the value of the DTS Transfer Count Compare Register are both stored in TI-H.

Accessing the DTS Channel Master Setting Register (DTSnnnCM) is performed via a 32-bit access to the entire TI-H.

When writing to the DTS Channel Master Setting Register, the lower 16 bits (DTS transfer count compare or CMC) are also updated. When reading from the DTS Channel Master Setting Register, the lower 16 bits represent the CMC value.

When reading from the DTS Transfer Count Compare Register (DTTCCnnn), hardware performs a 32-bit read access from TI-H but only the lower 16 bits are used for the result of the register read. When writing to the DTS Transfer Count Compare Register (DTTCCnnn), hardware performs a read-modify-write access to TI-H.

Since the contents of TI after reset are undefined, writing to the DTS Transfer Count Compare Register (DTTCCnnn) before writing to the DTS Channel Master Setting Register can result in an ECC error due to the read part of the read-modify-write access.

Some bits of TI-H are reserved. It is recommended to write 0 to these bits and to ignore the value read from these bits.

After reset, the values of TI in DTSRAM are undefined. The first access to TI after reset should be a write access. Otherwise an ECC error will occur.

- DTS Source Address Register (DTSAnnn)
- DTS Destination Address Register (DTDAnnn)
- DTS Transfer Count Register (DTTCnnn)
- DTS Transfer Control Register (DTTCTnnn)
- DTS Reload Source Address Register (DTRSAnnn)
- DTS Reload Destination Address Register (DTRDAnnn)
- DTS Reload Transfer Count Register (DTRTCnnn)
- Channel master setting registers (DTSnnnCM)

After reset a write access to the DTS Transfer Count Compare Register (DTTCCnnn) must follow a write access to the DTS Channel Master Setting Register (DTSnnnCM).

When accessing TI contents while a DTS transfer is in progress, the following implications should be considered.

- TI of the channel should not be updated by the CPU during the ongoing transfer. Otherwise a mismatch between the DTS transfer and the contents of TI may occur.
- If a TI access is requested from the CPU while a TI fetch or TI write back is being executed, the TI access is executed after the completion of the TI fetch or TI write back.
- If a TI fetch or TI write back is requested while a TI access request from the CPU is being processed, the TI fetch or TI write back is executed after the completion of the TI access.

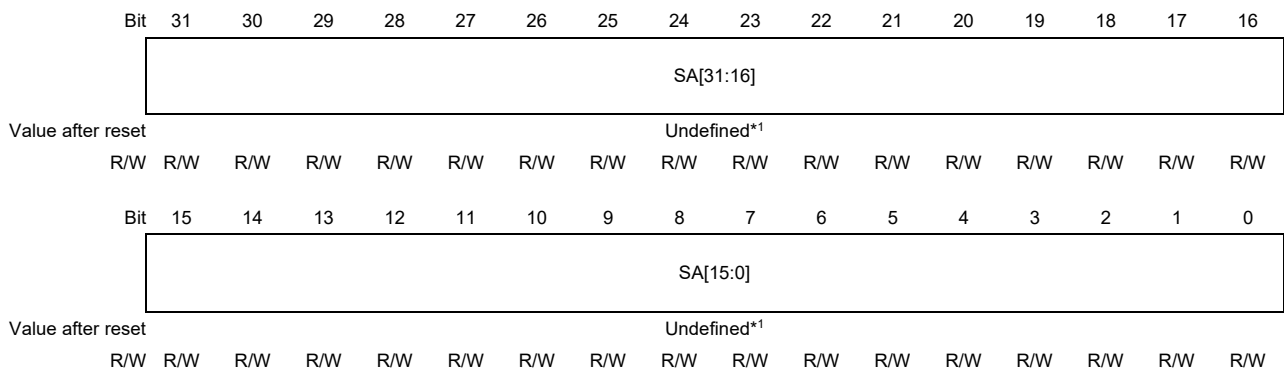
8.3.3.2 Details of DTS Channel Registers

The “nnn” in the register symbols indicates the DTS channel number (nnn = 000 to 127).

(1) DTSA_{nnn} — DTS Source Address Register

The DTS Source Address Register (DTSA_{nnn}) is a 32-bit read/write register that specifies the DTS transfer source address of each channel.

Value after reset: 0000 0000_H*¹ RAM initialization is executed.
 Undefined*¹ RAM initialization is not executed.



Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 10.3.6, RAM Initialization** for details.

Table 8.23 DTSA_{nnn} Register Contents

Bit Position	Bit Name	Function
31 to 0	SA[31:0]	Source address These bits specify the DTS transfer source address. This register is updated at the timing of TI writeback and retains the DTS transfer source address for the next DTS transfer.

NOTE

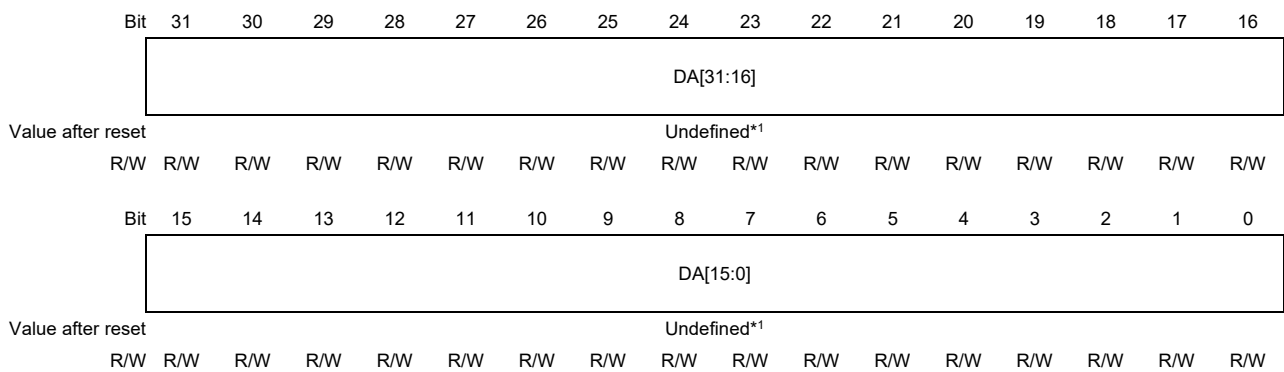
DTS transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size are as follows. The correct operation is not guaranteed if any other setting is specified.

Data Size	SA3	SA2	SA1	SA0
8 bits	0/1	0/1	0/1	0/1
16 bits	0/1	0/1	0/1	0
32 bits	0/1	0/1	0	0
64 bits	0/1	0	0	0
128 bits	0	0	0	0

(2) DTDAnnn — DTS Destination Address Register

The DTS Destination Address Register (DTDAnnn) is a 32-bit read/write register that specifies the DTS transfer destination address of each channel.

Value after reset: 0000 0000_H*¹ RAM initialization is executed.
Undefined*¹ RAM initialization is not executed.



Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 10.3.6, RAM Initialization** for details.

Table 8.24 DTDAnnn Register Contents

Bit Position	Bit Name	Function
31 to 0	DA[31:0]	Destination address These bits specify the DTS transfer destination address. This register is updated at the timing of TI writeback and retains the DTS transfer destination address for the next DTS transfer.

NOTES

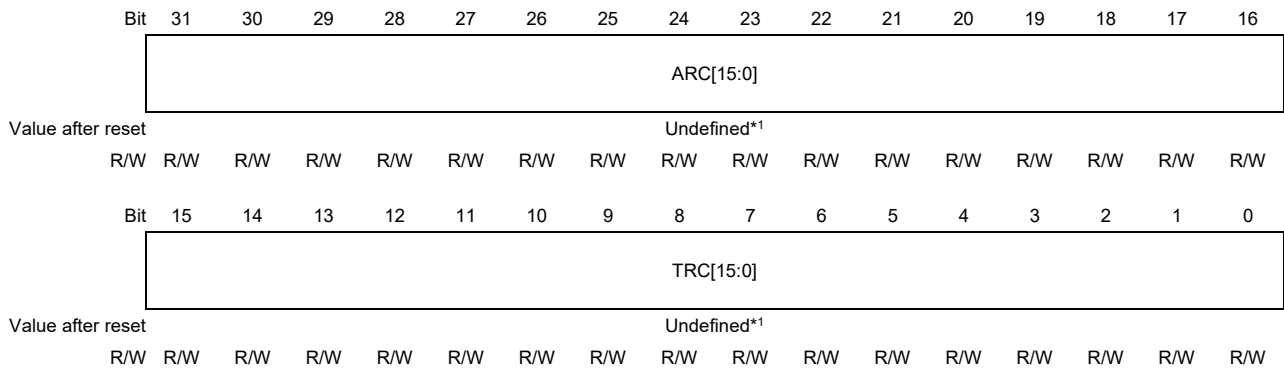
- If a DTS transfer error occurs in the read cycle of a DTS transfer, the write cycle is not executed, but the destination address is updated.
- DTS transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size are as follows. The correct operation is not guaranteed if any other setting is specified.

Data Size	DA3	DA2	DA1	DA0
8 bit	0/1	0/1	0/1	0/1
16 bit	0/1	0/1	0/1	0
32 bit	0/1	0/1	0	0
64 bit	0/1	0	0	0
128 bit	0	0	0	0

(3) DTTCnnn — DTS Transfer Count Register

The DTS Transfer Count Register (DTTCnnn) is a 32-bit read/write register that specifies the DTS transfer count of each channel.

Value after reset: 0000 0000_H*1 RAM initialization is executed.
Undefined*1 RAM initialization is not executed.



Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 10.3.6, RAM Initialization** for details.

Table 8.25 DTTCnnn Register Contents

Bit Position	Bit Name	Function										
31 to 16	ARC[15:0]	<p>Address reload count</p> <p>These bits specify the transfer count for the address reload function. The number is used for both reload function 2 and block transfer 2. When reload function 2 or block transfer 2 is used, ARC[15:0] are decremented by one for every DTS cycle and updated at the timing of TI writeback. When neither reload function 2 nor block transfer 2 is used, ARC[15:0] are not updated.</p> <p>If ARC[15:0] are 0000_H, address reload is not performed and ARC[15:0] are not updated.</p>										
15 to 0	TRC[15:0]	<p>Transfer count</p> <p>These bits configure the transfer count. TRC[15:0] are decremented by one whenever a DTS cycle is executed. These bits are updated at the timing of TI writeback. If the reload function is not used, the value at completion (0000_H) is retained after the completion of the last transfer. If 0000_H is set, DTS transfer is not executed even if a DTS transfer request is accepted.</p> <table border="1"> <thead> <tr> <th>TRC[15:0]</th> <th>Operation</th> </tr> </thead> <tbody> <tr> <td>0000_H</td> <td>Transfer is disabled or complete.</td> </tr> <tr> <td>0001_H</td> <td>The number of transfers or remaining transfers is 1.</td> </tr> <tr> <td>:</td> <td>:</td> </tr> <tr> <td>FFFF_H</td> <td>The number of transfers or remaining transfers is 65535.</td> </tr> </tbody> </table>	TRC[15:0]	Operation	0000 _H	Transfer is disabled or complete.	0001 _H	The number of transfers or remaining transfers is 1.	:	:	FFFF _H	The number of transfers or remaining transfers is 65535.
TRC[15:0]	Operation											
0000 _H	Transfer is disabled or complete.											
0001 _H	The number of transfers or remaining transfers is 1.											
:	:											
FFFF _H	The number of transfers or remaining transfers is 65535.											

NOTE

If a transfer error occurs in the read cycle of DTS transfer, the write cycle is not executed, but the transfer count and address reload count are updated.

(4) DTTCTnnn — DTS Transfer Control Register

The DTS Transfer Control Register (DTTCTnnn) is a 32-bit read/write register that defines DTS transfer configuration of the channel.

Value after reset: 0000 0000_H*¹ RAM initialization is executed.
Undefined*¹ RAM initialization is not executed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	ESE	—	—	CHNSEL[6:0]						CHNE[1:0]		
Value after reset	Undefined* ¹							Undefined* ¹								
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCE	TCE	—	RLD2M[1:0]	RLD1M[1:0]	DACM[1:0]	SACM[1:0]	DS[2:0]		TRM[1:0]						
Value after reset	Undefined* ¹							Undefined* ¹								
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 10.3.6, RAM Initialization** for details.

Table 8.26 DTTCTnnn Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 28	—	Reserved When read, the value returned is undefined. The write value should be 0.
27	ESE	Setting of DTS error transfer disable This bit specifies whether to abort DTS transfer when a DTS transfer error occurs. If this bit is cleared to 0, DTS transfer continues when a DTS transfer error occurs. If this bit is set to 1, DTS transfer is aborted when a DTS transfer error occurs. 0: DTS transfer continues when a DTS transfer error occurs. 1: DTS transfer is aborted when a DTS transfer error occurs.
26 to 25	—	Reserved When read, the value returned is undefined. The write value should be 0.
24 to 18	CHNSEL[6:0]	Next channel to chain These bits specify the next channel to chain. The next channel must be another channel in DTS. Setting the same channel for the next channel is not allowed. (Correct operation is not guaranteed in this case.)
17 to 16	CHNE[1:0]	Chain enable This bit selects the chain function. 00: Disabled 01: Chain at the last transfer A chain request is generated at the completion of the DTS cycle at which the remaining transfer count is 1. 10: (Setting prohibited) 11: Always chain A chain request is generated at the completion of every DTS cycle.
15	CCE	Transfer count match interrupt enable If this bit is set, a transfer count match interrupt is generated at the completion of the DTS cycle in which the Transfer Count Compare Register and remaining transfer count have the same value.

Table 8.26 DTTCTnnn Register Contents (2/3)

Bit Position	Bit Name	Function															
14	TCE	Transfer completion interrupt enable If this bit is set, a transfer completion interrupt is generated at the completion of the last transfer.															
13	—	Reserved When read, the value returned is undefined. The write value should be 0.															
12, 11	RLD2M[1:0]	Reload function 2 setting These bits configure reload function 2. 00: Reload function 2 is disabled. 01: Reload function 2 is enabled. The source address and address reload count are reloaded at the completion of the DTS cycle in which the address reload count is 1. 10: Reload function 2 is enabled. The destination address and address reload count are reloaded at the completion of the DTS cycle in which address reload count is 1. 11: Reload function 2 is enabled. The source address, destination address, and address reload count are reloaded at the completion of the DTS cycle in which address reload count is 1.															
10 to 9	RLD1M[1:0]	Reload function 1 setting These bits configure reload function 1. 00: Reload function 1 is disabled. 01: Reload function 1 is enabled. The source address and transfer count are reloaded at the completion of the DTS cycle in which remaining transfer count is 1. (If reload function 2 is enabled, the address reload count is also reloaded.) 10: Reload function 1 is enabled. The destination address and transfer count are reloaded at the completion of the DTS cycle in which remaining transfer count is 1. (If reload function 2 is enabled, the address reload count is also reloaded.) 11: Reload function 1 is enabled. The source address, destination address, and transfer count are reloaded at the completion of the DTS cycle in which remaining transfer count is 1. (If reload function 2 is enabled, the address reload count is also reloaded.)															
8 to 7	DACM[1:0]	Destination address count direction These bits specify the count direction of the destination address. <table border="1"> <thead> <tr> <th>DACM1</th> <th>DACM0</th> <th>Direction of Count</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Increment</td> </tr> <tr> <td>0</td> <td>1</td> <td>Decrement</td> </tr> <tr> <td>1</td> <td>0</td> <td>Fixed</td> </tr> <tr> <td>1</td> <td>1</td> <td>Prohibited (Operation not guaranteed)</td> </tr> </tbody> </table>	DACM1	DACM0	Direction of Count	0	0	Increment	0	1	Decrement	1	0	Fixed	1	1	Prohibited (Operation not guaranteed)
DACM1	DACM0	Direction of Count															
0	0	Increment															
0	1	Decrement															
1	0	Fixed															
1	1	Prohibited (Operation not guaranteed)															
6 to 5	SACM[1:0]	Source address count direction These bits specify the count direction of the source address. <table border="1"> <thead> <tr> <th>SACM1</th> <th>SACM0</th> <th>Direction of Count</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Increment</td> </tr> <tr> <td>0</td> <td>1</td> <td>Decrement</td> </tr> <tr> <td>1</td> <td>0</td> <td>Fixed</td> </tr> <tr> <td>1</td> <td>1</td> <td>Prohibited (Operation not guaranteed)</td> </tr> </tbody> </table>	SACM1	SACM0	Direction of Count	0	0	Increment	0	1	Decrement	1	0	Fixed	1	1	Prohibited (Operation not guaranteed)
SACM1	SACM0	Direction of Count															
0	0	Increment															
0	1	Decrement															
1	0	Fixed															
1	1	Prohibited (Operation not guaranteed)															

Table 8.26 DTTCTnnn Register Contents (3/3)

Bit Position	Bit Name	Function																												
4 to 2	DS[2:0]	Transfer data size These bits specify the transfer data size.																												
		<table border="1"> <thead> <tr> <th>DS2</th> <th>DS1</th> <th>DS0</th> <th>Transfer Data Size</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>8 bits</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>16 bits</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>32 bits</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>64 bits</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>128 bits</td> </tr> <tr> <td colspan="3">Other than the above</td> <td>Prohibited (Operation not guaranteed)</td> </tr> </tbody> </table>	DS2	DS1	DS0	Transfer Data Size	0	0	0	8 bits	0	0	1	16 bits	0	1	0	32 bits	0	1	1	64 bits	1	0	0	128 bits	Other than the above			Prohibited (Operation not guaranteed)
		DS2	DS1	DS0	Transfer Data Size																									
		0	0	0	8 bits																									
		0	0	1	16 bits																									
		0	1	0	32 bits																									
		0	1	1	64 bits																									
1	0	0	128 bits																											
Other than the above			Prohibited (Operation not guaranteed)																											
1 to 0	TRM[1:0]	Transfer mode These bits specify the DTS transfer mode.																												
		00: Single transfer																												
		01: Block transfer 1 (The number of transfers is specified by the transfer count.)																												
		10: Block transfer 2 (The number of transfers is specified by the address reload count.)																												
		11: Prohibited (Operation not guaranteed)																												

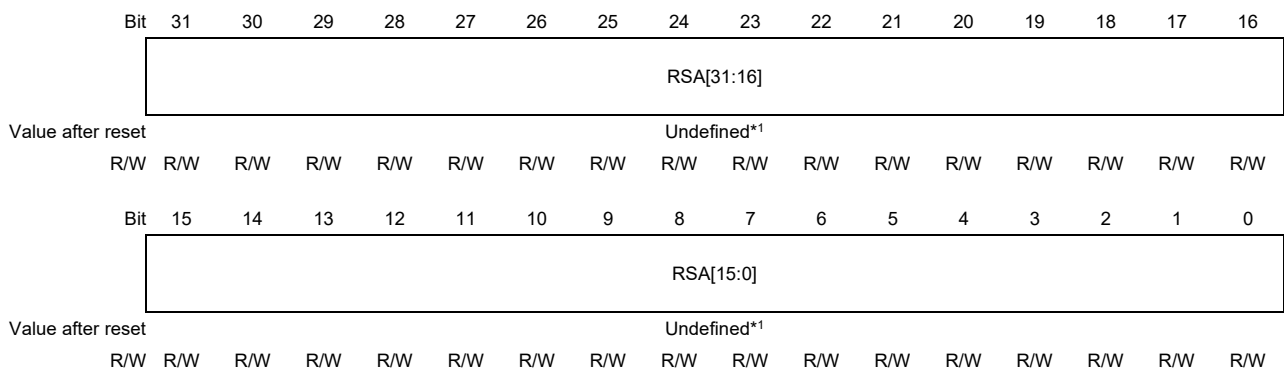
NOTES

1. If prohibited settings are used for some of the bits, the correct operation is not guaranteed.
2. Some bits in this register are unused, but these bits are readable and writable. It is recommended to write 0 to these bits and to ignore the value read from these bits.

(5) DTRSAnn — DTS Reload Source Address Register

The DTS Reload Source Address Register (DTRSAnn) is a 32-bit read/write register that specifies the DTS reload source address of each channel.

Value after reset: 0000 0000_H*¹ RAM initialization is executed.
Undefined*¹ RAM initialization is not executed.



Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 10.3.6, RAM Initialization** for details.

Table 8.27 DTRSAnn Register Contents

Bit Position	Bit Name	Function
31 to 0	RSA[31:0]	Reload source address These bits specify the source address to be reloaded when reload function 1 or reload function 2 is used.

NOTE

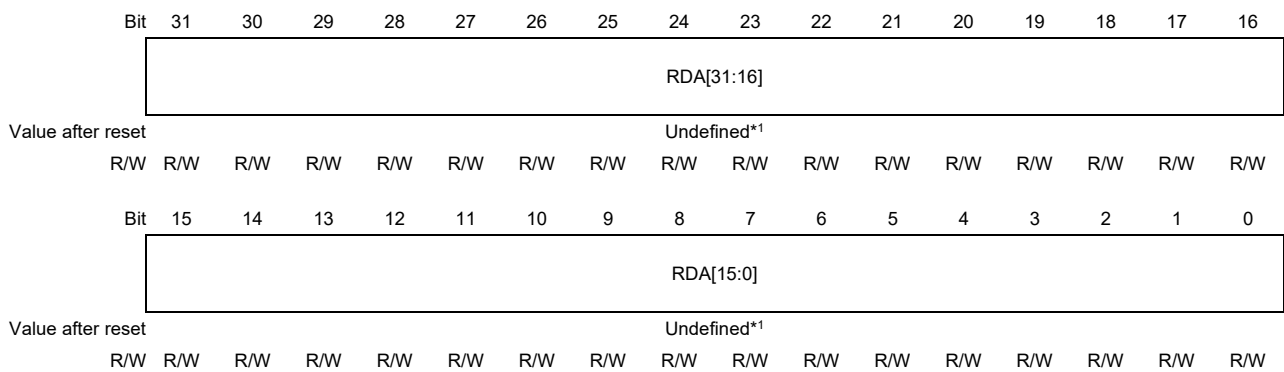
DTS transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size are as follows. The correct operation is not guaranteed if any other setting is specified.

Data Size	RSA3	RSA2	RSA1	RSA0
8 bits	0/1	0/1	0/1	0/1
16 bits	0/1	0/1	0/1	0
32 bits	0/1	0/1	0	0
64 bits	0/1	0	0	0
128 bits	0	0	0	0

(6) DTRDAnnn — DTS Reload Destination Address Register

The DTS Reload Destination Address Register (DTRDAnnn) is a 32-bit read/write register that specifies the DTS reload destination address of each channel.

Value after reset: 0000 0000_H*¹ RAM initialization is executed.
Undefined*¹ RAM initialization is not executed.



Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 10.3.6, RAM Initialization** for details.

Table 8.28 DTRDAnnn Register Contents

Bit Position	Bit Name	Function
31 to 0	RDA[31:0]	Reload destination address These bits specify the destination address to be reloaded when reload function 1 or reload function 2 is used.

NOTE

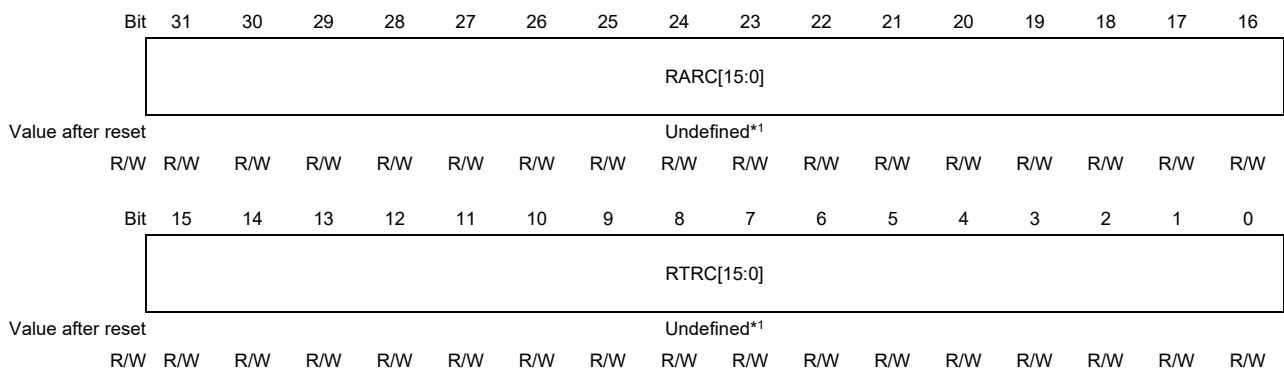
DTS transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size are as follows. The correct operation is not guaranteed if any other setting is specified.

Data Size	RDA3	RDA2	RDA1	RDA0
8 bits	0/1	0/1	0/1	0/1
16 bits	0/1	0/1	0/1	0
32 bits	0/1	0/1	0	0
64 bits	0/1	0	0	0
128 bits	0	0	0	0

(7) DTRTCnnn — DTS Reload Transfer Count Register

The DTS Reload Transfer Count Register (DTRTCnnn) is a 32-bit read/write register that specifies the DTS reload transfer count of each channel.

Value after reset: 0000 0000_H*1 RAM initialization is executed.
 Undefined*1 RAM initialization is not executed.



Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 10.3.6, RAM Initialization** for details.

Table 8.29 DTRTCnnn Register Contents

Bit Position	Bit Name	Function
31 to 16	RARC[15:0]	Reload address reload count These bits specify the value to be reloaded to the address reload count when reload function 2 is used.
15 to 0	RTRC[15:0]	Reload transfer count These bits specify the value to be reloaded to the transfer count when reload function 1 is used.
	RTRC[15:0]	Operation
	0000 _H	No DTS transfer
	0001 _H	1 transfer
	:	:
	FFFF _H	65535 transfers

(8) DTTCCnnn — DTS Transfer Count Compare Register

The DTS Transfer Count Compare Register (DTTCCnnn) is a 32-bit read/write register which configures the transfer count to be compared to transfer count register.

Value after reset: 0000 0000_H*¹ RAM initialization is executed.

Undefined*¹ RAM initialization is not executed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMC[15:0]															
Value after reset	Undefined* ¹															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The value after reset is dependent on setting of RAM initialization. See **Section 10.3.6, RAM Initialization** for details.

Table 8.30 DTTCCnnn Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	CMC[15:0]	Transfer count compare These bits configure the transfer count to be compared to the Transfer Count Register. If the transfer count match interrupt enable (DTTCTnnn.CCE) bit is 1, the transfer count match interrupt is generated at the completion of the DTS cycle in which remaining transfer count is the same as the value in this register. If CMC[15:0] is 0000H, comparison with the transfer count is disabled. In this case, the transfer count match interrupt is not generated.

Note: This register must be accessed after the DTS channel master setting register is set up. If this register is accessed without setting up the DTS channel master setting register after reset, an ECC error may occur.

(9) DTFSL_{nnn} — DTSFSL Operation Setting Register

The DTSFSL Operation Setting Register (DTFSL_{nnn}) is a 32-bit read/write register that controls the DTS transfer request of each channel.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	REQEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.31 DTFSL_{nnn} Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	REQEN	DTS transfer request enable This bit specifies whether a DTS transfer request of each channel is used as a candidate for DTS channel arbitration. 0: The DTS transfer request is not used as a candidate for DTS channel arbitration. 1: The DTS transfer request is used as a candidate for DTS channel arbitration. If this bit is 0, the request of the channel is not a candidate of DTS channel arbitration inside DTSFSL. DTS transfer request from the channel is not generated consequently, even if DTSFSL holds a request.

(10) DTFSTnnn — DTSFSL Transfer Status Register

The DTSFSL Transfer Status Register (DTFSTnnn) is a 32-bit read-only register that indicates the status of each channel.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	ERWR	—	—	—	ER	—	CC	TC	—	—	—	DRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.32 DTFSTnnn Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 12	—	Reserved When read, the value after reset is returned.
11	ERWR	DTS transfer error cycle When transfer error flag (ER) is set, this bit is also updated. This bit indicates whether a DTS transfer error has occurred in the read cycle or write cycle. If the ER bit is already set, this bit is not updated when another DTS transfer error occurs. This bit is also cleared when the ER bit is cleared. 0: A DTS transfer error occurred in the read cycle. 1: A DTS transfer error occurred in the write cycle
10 to 8	—	Reserved When read, the value after reset is returned.
7	ER	Transfer error flag This bit is set when a DTS transfer error occurs. If this bit is 1 and the DTTCTnnn.ESE bit is set, a DTS transfer is not executed when a DTS transfer request occurs. This bit is cleared by writing 1 to the DTFSCnnn.ERC bit. 0: No DTS transfer error has occurred 1: A DTS transfer error has occurred
6	—	Reserved When read, the value after reset is returned.
5	CC	Transfer count match flag This bit is set at the completion of the DTS transfer in which the remaining transfer count is the same as the value set in the transfer compare register. This bit is cleared by writing 1 to the DTFSCnnn.CCC bit. 0: No compare match with the Transfer Count Compare Register has occurred. 1: A compare match with the Transfer Count Compare Register has occurred.
4	TC	Transfer completion flag This bit is set at the completion of the last transfer and shows whether DTS transfer is complete. This bit is cleared by writing 1 to the DTFSCnnn.TCC bit. 0: DTS transfer is not complete 1: DTS transfer is complete
3 to 1	—	Reserved When read, the value after reset is returned.

Table 8.32 DTFSTnnn Register Contents (2/2)

Bit Position	Bit Name	Function
0	DRQ	<p>DTS transfer request flag</p> <p>This bit shows whether a DTS transfer request of this channel is being held.</p> <p>This bit is set when DTS transfer request is detected, or when the DTFSSnnn.DRQS bit is written by software.</p> <p>This bit is automatically cleared when DTS accepts a transfer request while DTSFSL is requesting DTS transfer for the channel in question. This bit also can be cleared by writing to the DTFSCnnn.DRQC bit.</p> <p>0: A DTS transfer request was not detected.</p> <p>1: A DTS transfer request was detected and is being held in DTSFSL.</p>

(11) DTFSSnnn — DTSFSL Transfer Request Set Register

The DTSFSL Transfer Request Set Register (DTFSSnnn) is 32-bit write-only register that sets the DRQ bit in the DTFSTnnn register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRQS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 8.33 DTFSSnnn Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When writing, write the value after reset.
0	DRQS	DTS transfer request set Writing 1 to this bit will generate a DTS transfer request.

(12) DTFSCnnn — DTSFSL Transfer Status Clear Register

The DTSFSL Transfer Status Clear Register (DTFSCnnn) is 32-bit write-only register that clears the status flag bits in the DTFSTnnn register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ERC	—	CCC	TCC	—	—	—	DRQC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	R	W	W	R	R	R	W

Table 8.34 DTFSCnnn Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When writing, write the value after reset.
7	ERC	Transfer error flag (ER) clear The DTFSTnnn.ER bit can be cleared by writing 1 to this bit.
6	—	Reserved When writing, write the value after reset.
5	CCC	Transfer count match flag (CC) clear The DTFSTnnn.CC bit can be cleared by writing 1 to this bit.
4	TCC	Transfer completion flag (TC) clear The DTFSTnnn.TC bit can be cleared by writing 1 to this bit.
3 to 1	—	Reserved When writing, write the value after reset.
0	DRQC	DTS transfer request clear The DTFSTnnn.DRQ bit can be cleared by writing 1 to this bit.

8.3.4 Detail DTSSSELm

8.3.4.1 DTSSSELm — DTS Transfer Request Group Selection Register (m = 0 to 15)

DTSSSELm register is 32-bit read/write register. The DTSSSELm register specifies the DTS transfer request group for each channel.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SEL7		—	—	SEL6		—	—	SEL5		—	—	SEL4	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEL3		—	—	SEL2		—	—	SEL1		—	—	SEL0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Table 8.35 DTSSSELm Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
29, 28	SEL7[1:0]	These bits select the DTS transfer request group for DTS channel $8 \times m + 7$.
27, 26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25, 24	SEL6[1:0]	These bits select the DTS transfer request group for DTS channel $8 \times m + 6$.
23, 22	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
21, 20	SEL5[1:0]	These bits select the DTS transfer request group for DTS channel $8 \times m + 5$.
19, 18	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
17, 16	SEL4[1:0]	These bits select the DTS transfer request group for DTS channel $8 \times m + 4$.
15, 14	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
13, 12	SEL3[1:0]	These bits select the DTS transfer request group for DTS channel $8 \times m + 3$.
11, 10	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
9, 8	SEL2[1:0]	These bits select the DTS transfer request group for DTS channel $8 \times m + 2$.
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5, 4	SEL1[1:0]	These bits select the DTS transfer request group for DTS channel $8 \times m + 1$.
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	SEL0[1:0]	These bits select the DTS transfer request group for DTS channel $8 \times m + 0$.

Table 8.35 DTSSSELm Register Contents (2/2)

Bit Name	Function
SELn[1:0]	00 _B : Select DTS transfer request group 0
	01 _B : Select DTS transfer request group 1
	10 _B : Select DTS transfer request group 2
	11 _B : Select DTS transfer request group 3

Figure 8.3 shows DTS transfer request group selection.

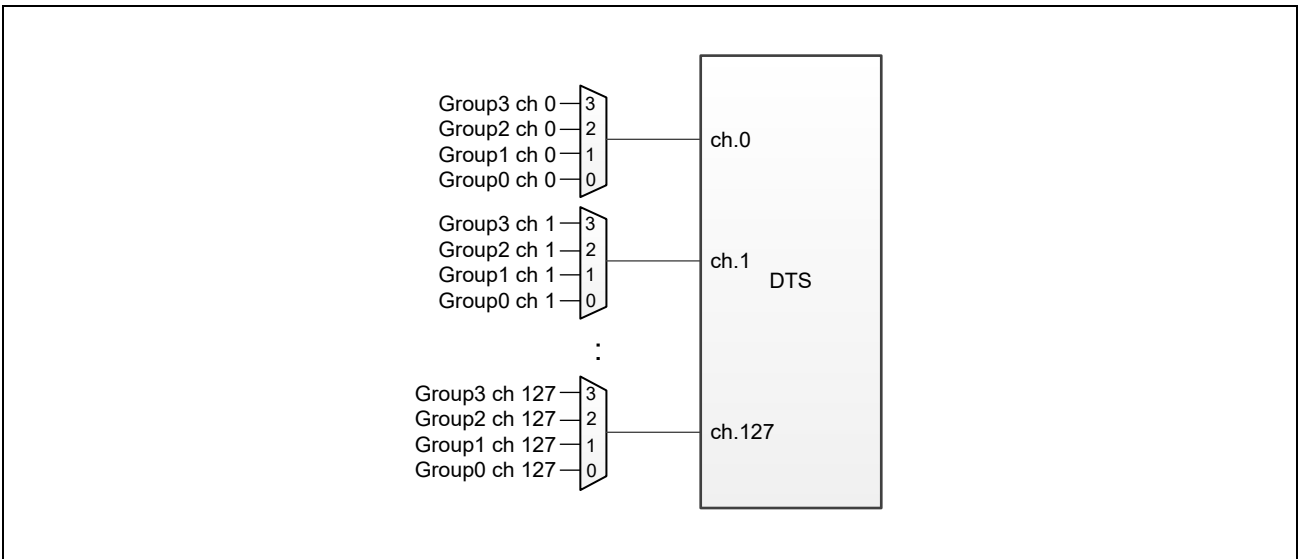


Figure 8.3 DTS Transfer Request Group Selection

8.4 Operation

8.4.1 Basic Operation of DTS Transfer

8.4.1.1 Transfer Mode

DTS has three transfer modes.

(1) Single Transfer

One DTS cycle is executed when a DTS transfer request is accepted.

(2) Block Transfer 1

The number of DTS cycles specified by the TRC bits in the DTS Transfer Count Register are executed when a DTS transfer request is accepted.

(3) Block Transfer 2

The number of DTS cycles specified by the ARC bits in the DTS Transfer Count Register are executed when a DTS transfer request is accepted. If the address reload count (ARC) is larger than the value of the transfer count (TRC), the number of DTS cycles is defined as the value of TRC.

8.4.1.2 Executing a DTS Cycle

DTS always executes the write cycle after the read cycle is complete. For example, if the transfer data size is 128 bits, the write cycle is executed after the read cycle for the 128-bit data is finished. The write cycle never starts before the read cycle is completed.

8.4.1.3 Updating Transfer Information

When a DTS cycle is executed, the DTS updates the transfer information as follows.

(1) Source Address and Destination Address

Transfer information (TI) will be updated as described in **Table 8.36** according to the settings of the DTS Transfer Control Register (DTTCTnnn) such as the source address count direction, the destination address count direction, and the transfer data size.

Table 8.36 Updating the Source Address and the Destination Address

Direction of Count	Transfer Data Size	Address after Update
Increment	8 bit	(address before update) + 0000 0001 _H
	16 bit	(address before update) + 0000 0002 _H
	32 bit	(address before update) + 0000 0004 _H
	64 bit	(address before update) + 0000 0008 _H
	128 bit	(address before update) + 0000 0010 _H
Decrement	8 bit	(address before update) – 0000 0001 _H
	16 bit	(address before update) – 0000 0002 _H
	32 bit	(address before update) – 0000 0004 _H
	64 bit	(address before update) – 0000 0008 _H
	128 bit	(address before update) – 0000 0010 _H
Fixed	—	Same as the address before update.

When the reload function is used, a specific update rule is applied other than the one described in **Table 8.36** for the last transfer and the address reload transfer. For details, see **Section 8.4.3, Reload Function**.

(2) Transfer Count/Address Reload Count

The transfer count is decremented by one for every DTS cycle.

The address reload count is decremented by one for every DTS cycle when reload function 2 or block transfer 2 is used. When neither reload function 2 nor block transfer 2 is used, it is not updated.

If the reload function is used, a specific update rule is applied for the last transfer and the address reload transfer.

For details, see **Section 8.4.3, Reload Function**.

(3) Other Transfer Information

Other transfer information is not updated during execution of a DTS cycle.

8.4.1.4 Last Transfer and Address Reload Transfer

The last transfer means the DTS cycle executed when the value in the Transfer Count Register, which shows the remaining number of transfers, is 1. The last transfer differs in operation compared to the other DTS cycles as follows.

- The transfer completion flag (DTFSTnnn.TC) is set when the last transfer is complete.
- When the transfer completion interrupt output enable is set, a transfer completion interrupt is output when the last transfer is complete.
- When reload function 1 is enabled, reload function 1 is executed at the timing of the last transfer. For details, see **Section 8.4.3, Reload Function**.

Address reload transfer means the DTS cycle executed when reload function 2 is enabled and the address reload count is 1. Reload function 2 is executed after address reload transfer. For details, see **Section 8.4.3, Reload Function**.

8.4.1.5 Transfer Completion Interrupt and Transfer Count Match Interrupt Output

DTS can output a transfer completion interrupt and a transfer count match interrupt to outside DTS.

(1) Transfer Completion Interrupt Output

When the transfer completion interrupt output enable bit (DTTCTn.TCE) is set in the Transfer Control Register, DTS requests a DTS transfer completion interrupt when the last transfer is complete.

(2) Transfer Count Match Interrupt Output

When the transfer count match interrupt enable bit (DTTCTn.CCE) is set in the Transfer Control Register, DTS requests a DTS transfer count match interrupt at the completion of the DTS cycle in which the Transfer Count Compare Register and the transfer count have the same value.

Figure 8.4 shows the operation of the transfer completion interrupt and the transfer count match interrupt.

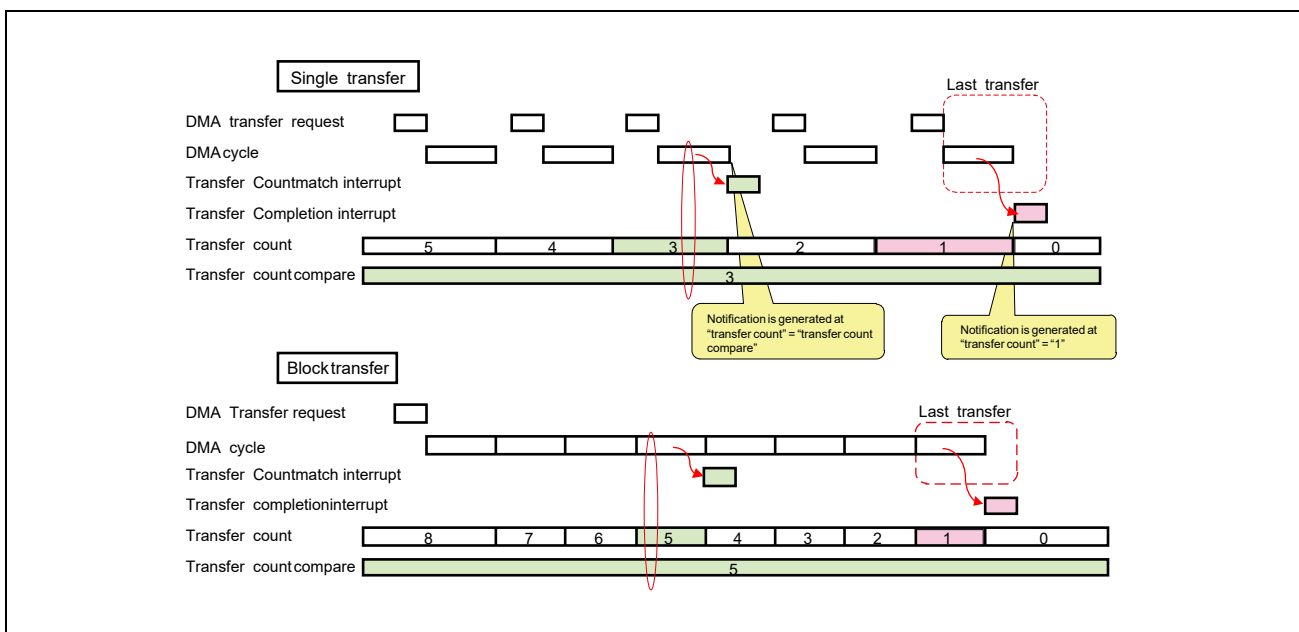


Figure 8.4 Transfer Completion Interrupt and Transfer Count Match Interrupt

8.4.1.6 Continuous Transfer

DTS does not have a configuration that supports continuous transfer.

DTS does not start DTS transfer when a DTS transfer request is generated while the transfer count is 0.

If the reload function 1 is used and the transfer count with the value other than 0 is reloaded to the transfer count bits (TRC[15:0]) in the DTS Transfer Count Register (DTTC_{nnn}) when the last transfer is complete, the DTS can start the transfer after the next DTS transfer request is accepted.

8.4.2 Channel Priority Order

This subsection explains arbitration between multiple DTS channels.

8.4.2.1 DTS Channel Arbitration

If there are DTS transfer requests from multiple DTS channels, DTSFSL arbitrates those DTS channels. For each DTS channel, a priority can be selected from four levels using the DTS channel priority setting registers.

If there are DTS transfer requests from multiple DTS channels, arbitration is executed as follows.

The channel with the higher priority level in the setting of the DTS channel priority setting register has priority.

If two channels have the same priority level specified in the DTS channel priority setting register, the channel with the smaller channel number has priority.

DTSFSL sends DTS a DTS transfer request for the channel selected by arbitration. DTS executes DTS transfer when it accepts the DTS transfer request.

DTS transfer does not allow arbitration between DTS channels in the middle of a block transfer. If a higher priority DTS transfer request is issued during a lower priority block transfer, the higher priority DTS transfer has to wait until the current block transfer is complete*1.

Note 1. Completion of block transfer means either the last transfer or the last transfer for block transfer 1 or the address reload transfer for block transfer 2 is finished.

When the DTS is executing the block transfer 1 or the block transfer 2, a DTS cycle of a DTS channel with a higher priority does not take over the ongoing block transfer until the last transfer is completed.

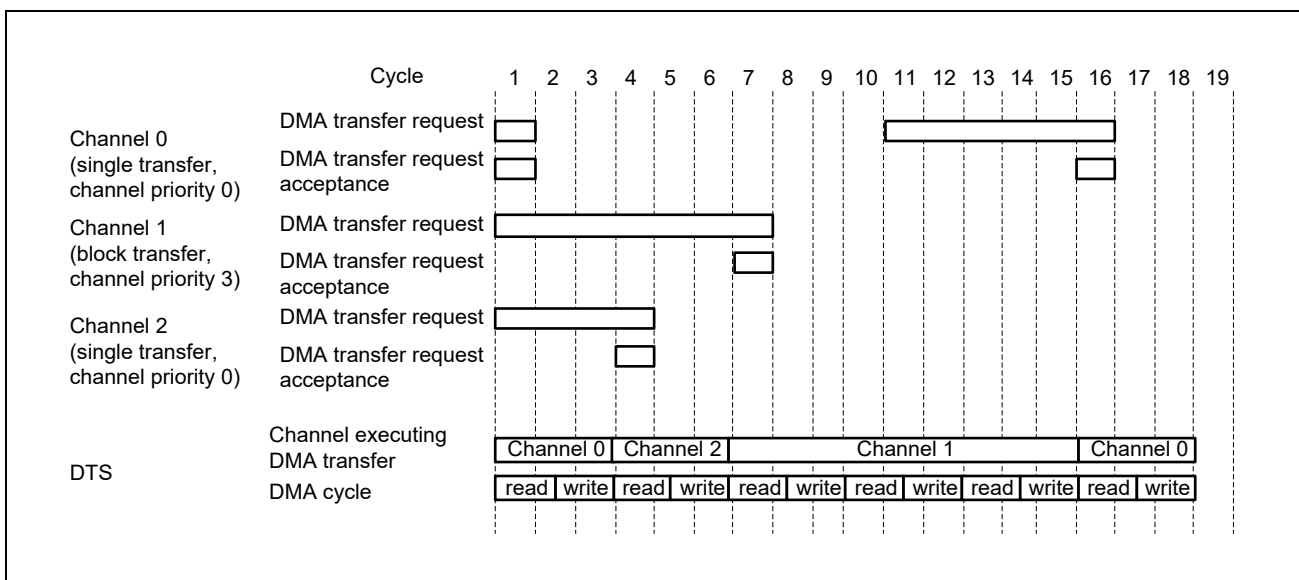


Figure 8.5 DTS Channel Arbitration

Cycle numbers shown in **Figure 8.5** are for explanation purposes only. They do not indicate the actual number of cycles necessary for executing DTS transfer.

In **Figure 8.5**, DTS transfer requests for channels 0, 1, and 2 are generated at Cycle 1. The channel priority for channels 0 and 2 is 0 and is higher than the channel priority for channel 1, which is 3. In addition, if two channels have the same priority, the channel with the smaller channel number has higher priority. Consequently, the priority order for

arbitration is “channel 0 > channel 2 > channel 1”, and a DTS cycle for channel 0 starts because its priority is the highest. As a result of arbitration between channels 1 and 2 at cycle 4, a DTS cycle for channel 2 starts. A DTS cycle for channel 1 starts at Cycle 7, and continuous cycles follow until Cycle 15 because channel 1 uses block transfer. At Cycle 11, a DTS transfer request for channel 0 is generated. The DTS cycle for channel 1 is still ongoing and no arbitration is executed until the block transfer of channel 1 is complete.

At cycle 15, the block transfer of channel 1 is complete. At cycle 16, a DTS cycle for channel 0 starts.

8.4.3 Reload Function

8.4.3.1 Overview of the Reload Function

The reload function updates a portion of the transfer information, more specifically, the source address, destination address, transfer count and address reload count, to predefined values during DTS transfer.

The reload function has two types of functions: reload function 1 and reload function 2.

8.4.3.2 Operation of Reload Function 1

When reload function 1 is enabled, the actions described in **Table 8.37** are executed at the timing of the last transfer according to the reload function 1 settings.

Table 8.37 Operation of Reload Function 1

Reload Function 1 Setting	Register	Action at the Last Transfer
00 (Reload function 1 disabled.)	Source address	Not reloaded.
	Destination address	Not reloaded.
	Transfer count	Not reloaded.
	Address reload count	Not reloaded.
01 (Reload function 1 enabled. Reloading source address and transfer count.)	Source address	The reload source address is copied to this.
	Destination address	Not reloaded.
	Transfer count	The reload transfer count is copied to this.
	Address reload count	If reload function 2 is disabled: Not reloaded. If reload function 2 is enabled: The reload address reload count is copied to this.
10 (Reload function 1 enabled. Reloading destination address and transfer count.)	Source address	Not reloaded.
	Destination address	The reload destination address is copied to this.
	Transfer count	The reload transfer count is copied to this.
	Address reload count	If reload function 2 is disabled: Not reloaded. If reload function 2 is enabled: The reload address reload count is copied to this.
11 (Reload function 1 enabled. Reloading source address, destination address, and transfer count.)	Source address	The reload source address is copied to this.
	Destination address	The reload destination address is copied to this.
	Transfer count	The reload transfer count is copied to this.
	Address reload count	If reload function 2 is disabled: Not reloaded. If reload function 2 is enabled: The reload address reload count is copied to this.

Figure 8.6 shows the operation of reload function 1.

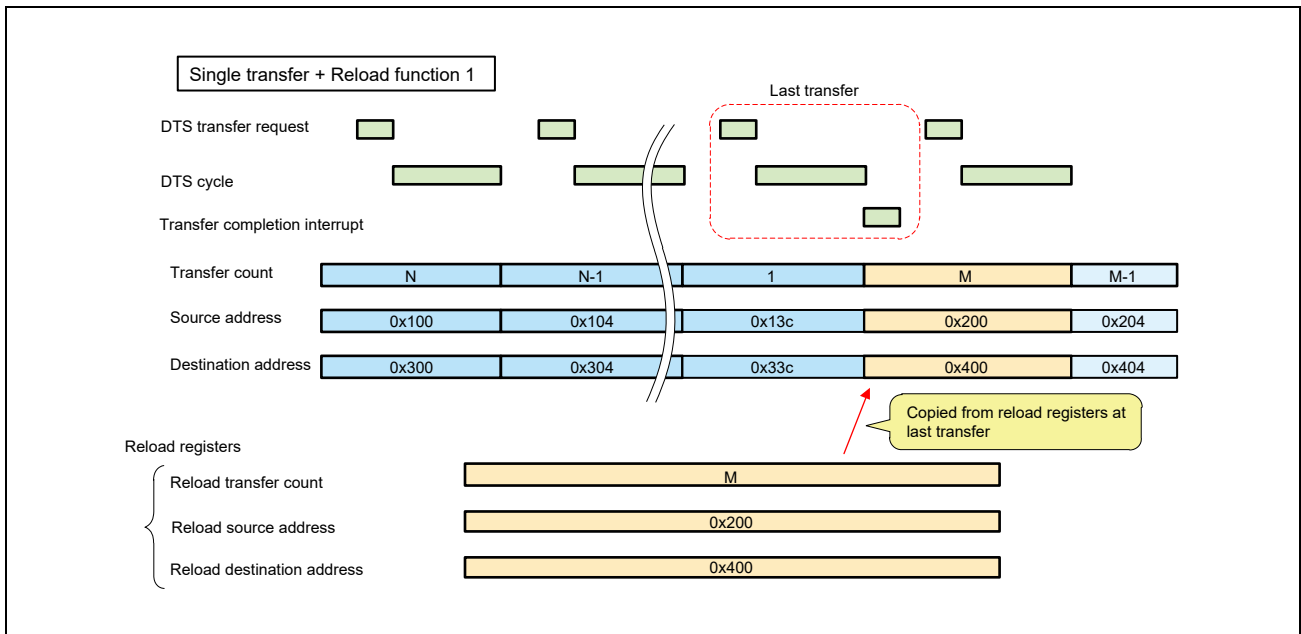


Figure 8.6 Operation of Reload Function 1

8.4.3.3 Reload Function 2

When reload function 2 is enabled, the actions described in **Table 8.38** are executed at the timing of address reload transfer according to the reload function 2 settings.

Table 8.38 Operation of Reload Function 2

Reload Function 2 Setting	Register	Action at Address Reload Transfer
00 (Reload function 2 disabled.)	Source address	Not reloaded.
	Destination address	Not reloaded.
	Address reload count	Not reloaded.
01 (Reload function 2 enabled. Reloading source address.)	Source address	The reload source address is copied to this.
	Destination address	Not reloaded.
	Address reload count	The reload address reload count is copied to this.
10 (Reload function 2 enabled. Reloading destination address.)	Source address	Not reloaded.
	Destination address	The reload destination address is copied to this.
	Address reload count	The reload address reload count is copied to this.
11 (Reload function 2 enabled. Reloading source address and destination address.)	Source address	The reload source address is copied to this.
	Destination address	The reload destination address is copied to this.
	Address reload count	The reload address reload count is copied to this.

Figure 8.7 shows the operation of reload function 2.

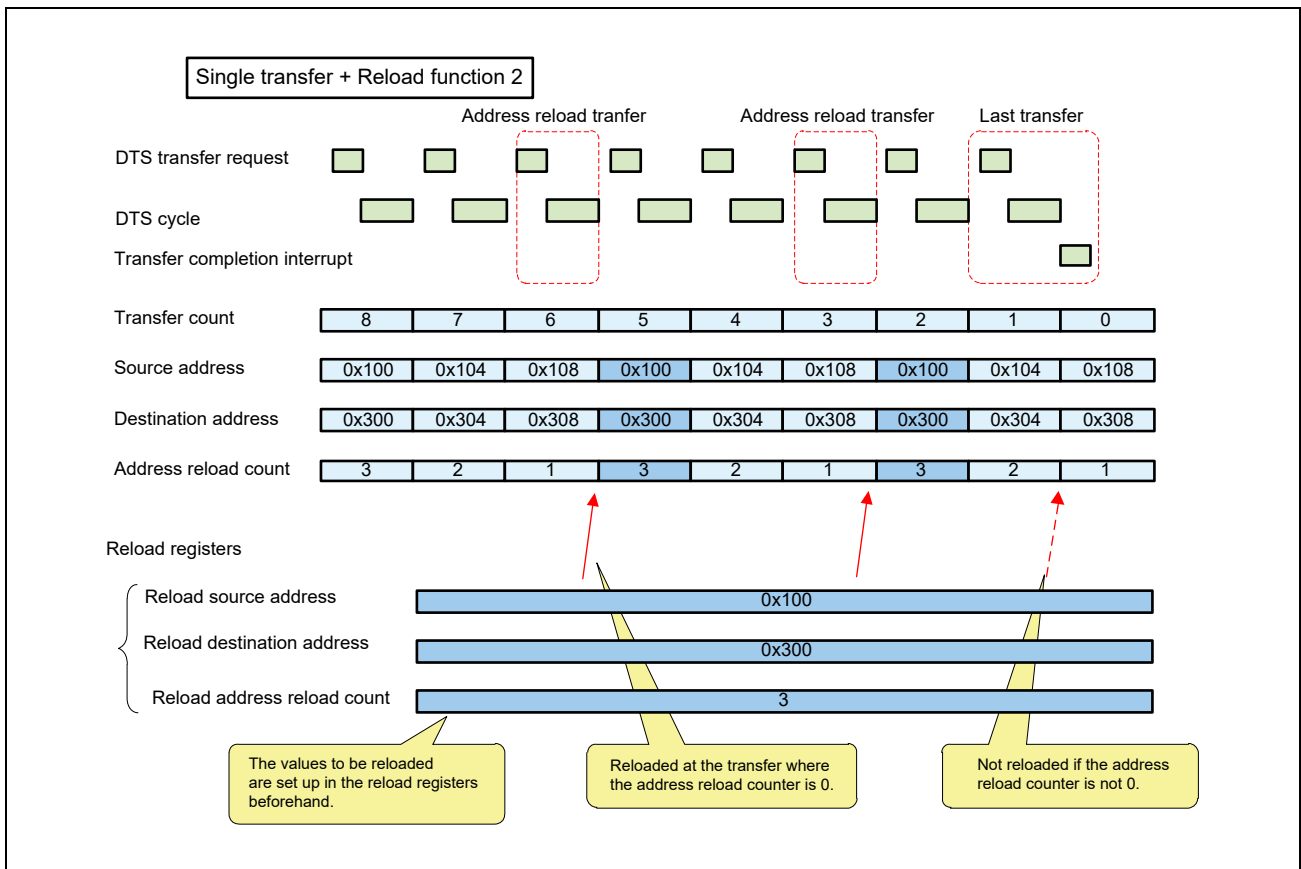


Figure 8.7 Operation of Reload Function 2

Figure 8.8 shows the operation when both reload function 1 and reload function 2 are used simultaneously.

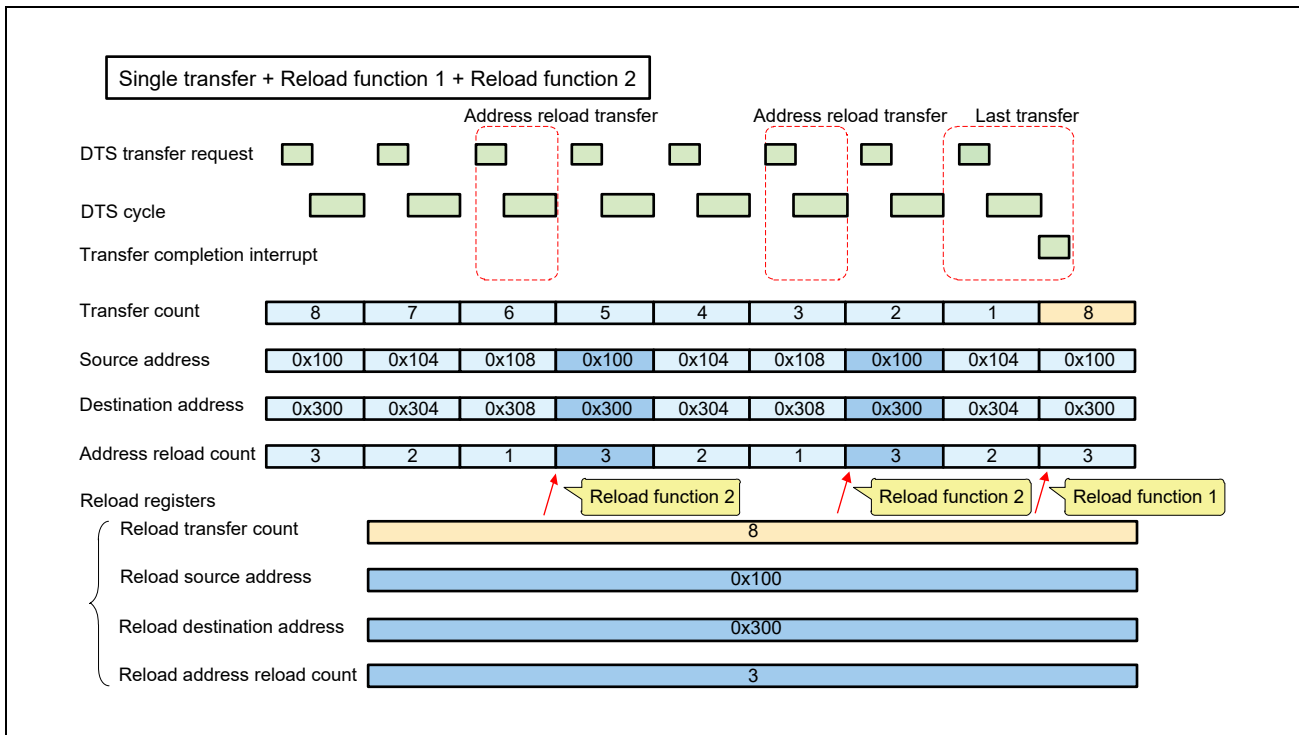


Figure 8.8 Operation When Combining Reload Function 1 and Reload Function 2

8.4.3.4 Timing of Setting DTS Reload Registers

It should be noted that the right timing of setting up the reload source address information, reload destination address information and reload transfer count information differs depending on the transfer mode.

In the single transfer mode, the TI fetched at the beginning of the last transfer or address reload transfer is used for reload at the completion of the DTS cycle. Therefore, if the reload function is used for a single transfer, the reload source address information, reload destination address information, and reload transfer count information in the TI must be set up before the beginning of the last transfer or address reload transfer.

For a block transfer, TI is fetched only at the beginning of the block transfer. The TI is used for the reload function at the last transfer or the address reload transfer. Therefore, when using the reload function for a block transfer, the reload contents in the TI must be set up before the beginning of the block transfer. If the reload contents is updated during the block transfer, those new settings cannot be reflected to the TI nor the next transfer.

8.4.4 Chain Function

8.4.4.1 Overview

DTS provides a function called chain function. When using the chain function, the completion of the DTS cycle or the last transfer for one channel can trigger a DTS transfer request for another channel. The DTS transfer request initiated by the chain function is called a chain request. You can select the condition for generating a chain request from the following two options.

- Always chain: A chain request is generated at the completion of every DTS cycle.
- Chain at the last transfer: A chain request is generated at the completion of the last transfer.

Figure 8.9 shows the operation of the “always chain” option.

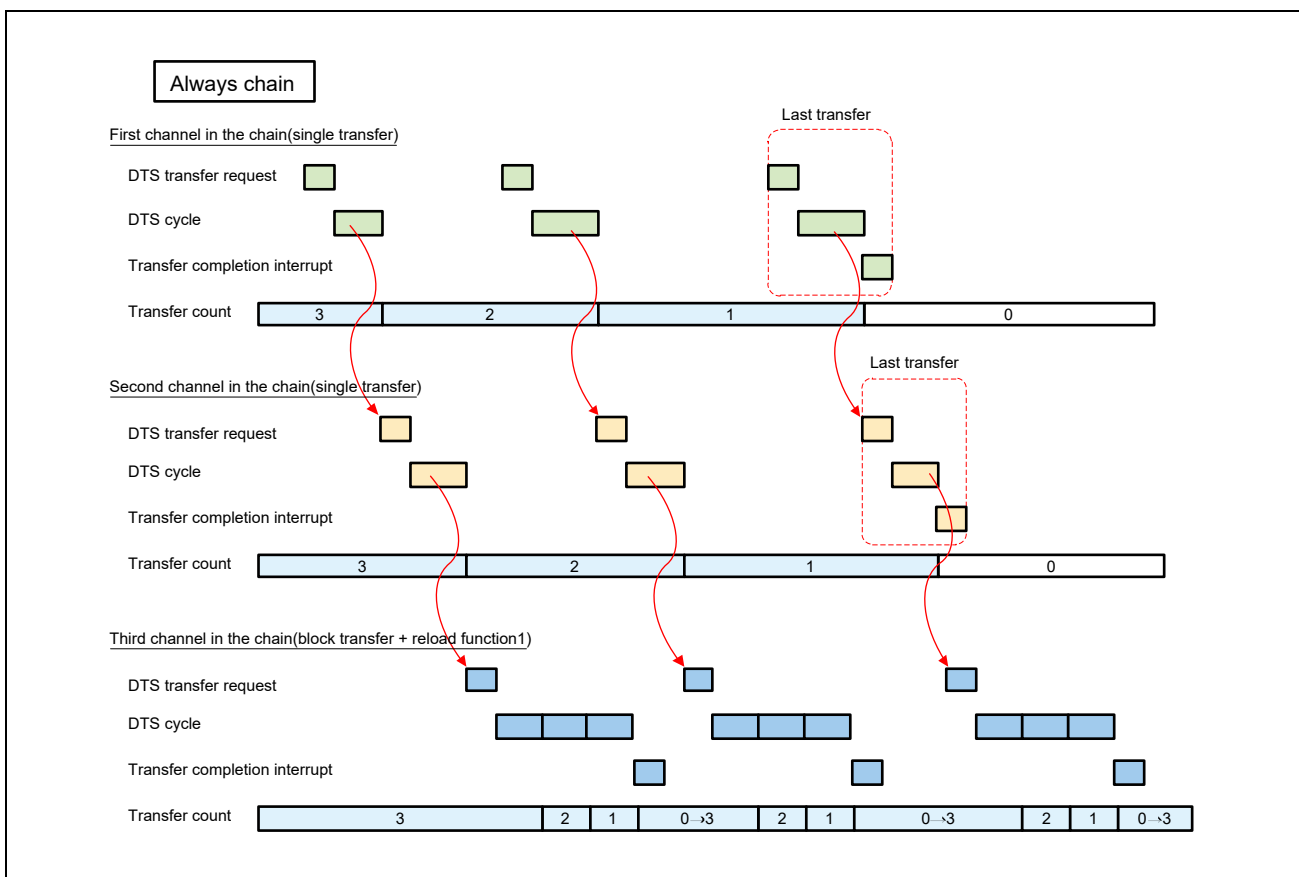


Figure 8.9 Operation of “Always Chain” Option

Figure 8.10 shows the operation of the “chain at the last transfer” option.

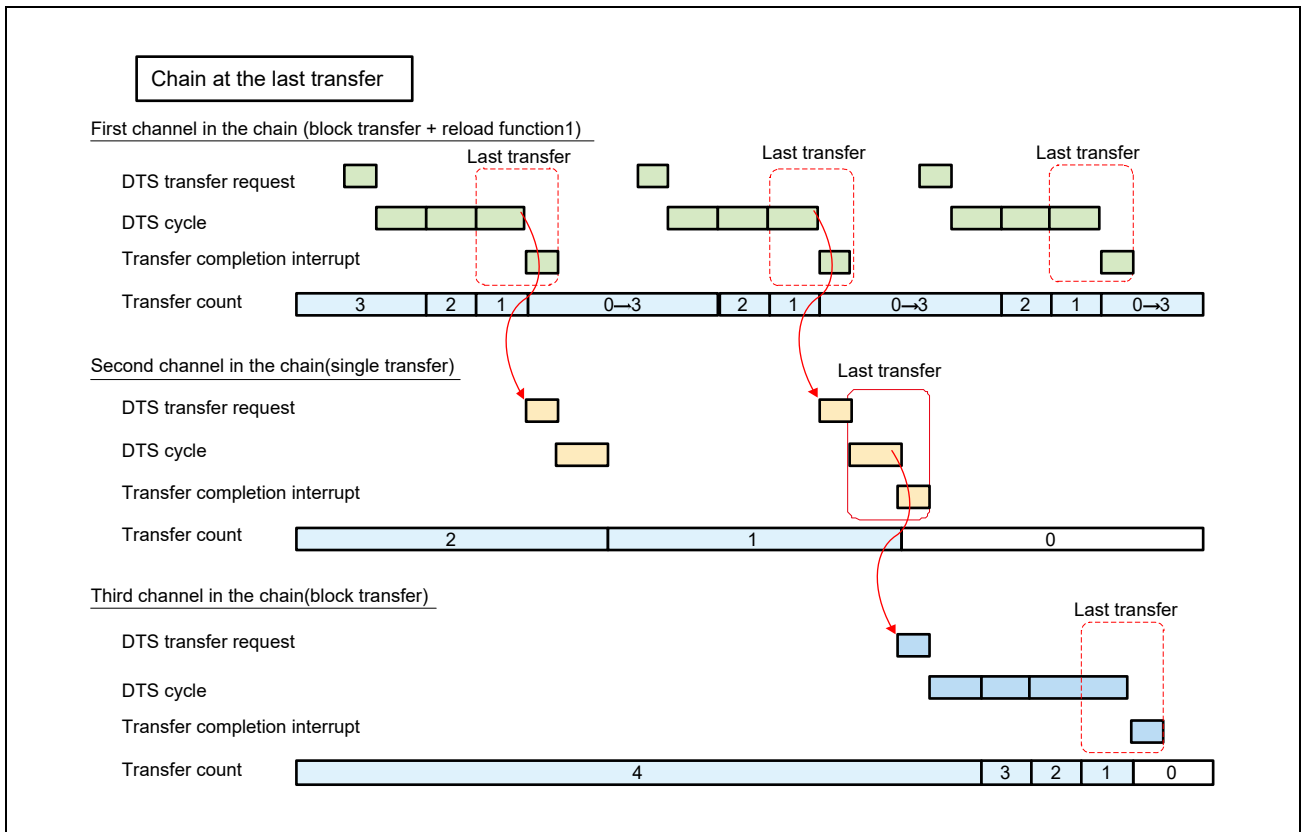


Figure 8.10 Operation of “Chain at the Last Transfer” Option

8.4.4.2 Setting Up the Chain Function

When using the chain function, it has to be enabled by the chain enable (DTTCTnnn.CHNE) and the next channel in the chain has to be specified in the chain selection (DTTCTnnn.CHNSEL) of the DTS Transfer Control Register.

If the reliability functions are required, the chain restriction (DTSnnnCM.CHAIN_RESTRICT, CHAIN_SPID[4:0], CHAIN_UM) also has to be specified in DTS Channel Master Setting Register.

For detailed information about reliability function, see **Section 8.7, Reliability Functions** for details.

8.4.5 DTS Operation

8.4.5.1 Types of DTS Transfer Requests and Assigning DTS Transfer Requests

DTS starts DTS transfer by accepting a hardware DTS transfer request or software DTS transfer request.

A transfer request for DTS is retained in the transfer request pending bit of DTSFSL for each channel.

For DTSFSL, both a hardware DTS transfer request and a software DTS transfer request are retained in the same transfer request pending bit. When executing DTS transfer, DTS does not care whether the DTS transfer request is a hardware DTS transfer request or software DTS transfer request.

8.4.5.2 Generating and Accepting a DTS Transfer Request

DTS only supports edge detection for hardware DTS transfer requests.

When DTSFSL detects a hardware DTS transfer source*¹ input, DTSFSL sets the transfer request pending bit and retains the hardware DTS transfer source as a DTS transfer request. If the transfer request pending bit is set and the transfer request enable bit (DTFSLnnn.REQEN) in the DTSFSL operation setting register is set, DTSFSL notifies DTS of the DTS transfer request.

Software can also generate a DTS transfer request by setting the transfer request pending bit (DTFSTnnn.DRQ) using the DTSFSL transfer request set register (DTFSSnnn).

DTSFSL can retain only one DTS transfer request per channel. If, while the transfer request pending bit for a channel is set, a new hardware DTS transfer source input for the same channel occurs, DTSFSL ignores the new hardware DTS transfer source input.

The transfer request pending bit is automatically cleared when DTS accepts the DTS transfer request. DTSFSL clears the transfer request pending bit automatically when DTS accepts the DTS transfer request regardless of the type of DTS transfer to be executed by DTS.

The transfer request pending bit can also be cleared using the DTSFSL transfer status clear register (DTFSCnnn). If the transfer request pending bit of a channel is cleared before DTS accepts the DTS transfer request, DTS transfer of the channel is not executed.

Note 1. For details, refer to Appendix file "DTS_Transfer_request_Table.xlsx".

8.4.5.3 Executing DTS Transfer

When DTS accepts a DTS transfer request for a channel, DTS executes DTS transfer on the channel. If there are DTS transfer requests from multiple channels, DTSFSL arbitrates the DTS channels and selects the channel for the DTS transfer request.

While DTS is executing DTS transfer, the DTS transfer status (DTSSTS.DTSACT) bit in the DTS status register is set. In addition, the channel number of the currently ongoing DTS transfer is set in the DTS transfer channel (DTSSTS.DTSACH).

When the DTS transfer is complete or aborted because of a DTS transfer error or a register write and no channel is currently executing DTS transfer, the DTS transfer status (DTSSTS.DTSACT) bit is cleared.

8.4.5.4 DTSRAM Access

DTS accesses the DTSRAM when DTS transfer starts and finishes.

The DTS action of reading transfer information from the DTSRAM when DTS transfer starts is called TI fetch.

The DTS action of updating the transfer information on the DTSRAM when DTS transfer finishes is called TI writeback.

A single transfer performs a TI fetch at the beginning of a DTS cycle and a TI writeback at the end of a DTS cycle.

A block transfer performs a TI fetch at the beginning of the first DTS cycle and a TI writeback at the end of the DTS cycle that satisfies the block transfer completion condition (the last transfer or address reload transfer).

Therefore, in the case of a single transfer, the transfer information in DTSRAM is updated during each DTS cycle. In the case of block transfer, the transfer information in DTSRAM is updated after completion of the block transfer. If software reads the transfer information from DTSRAM during execution of a block transfer, the information represents the beginning of the block transfer.

8.5 Suspension, Resume, and Transfer Abort, and Clearing a DTS Transfer Request

8.5.1 Suspend, Resume, and Transfer Abort of a DTS Transfer

A DTS transfer being executed by DTS can be suspended by setting the DTS suspend bit (DTSCTL1.DTSUST) in DTS control register 1. If a DTS cycle is in progress, the DTS transfer is suspended after the DTS cycle is finished. If the current DTS cycle is a single transfer or a transfer that completes a block transfer (the last transfer or address reload transfer), the DTS transfer is suspended after a TI writeback and completion of the DTS cycle.

If the type of the current DTS cycle is other than the above, DTS transfer is suspended after the completion of the DTS cycle without a TI writeback. To resume DTS transfer after suspension, the DTS suspend bit in DTS control register 1 has to be cleared.

To abort DTS transfer by DTS, first suspend DTS as described above and then set the DTS transfer abort request bit (DTSCTL2.DTSTIT) in DTS control register 2. If a transfer is aborted, no TI writeback is executed. In addition, aborting a DTS transfer does not change the value of the DTS suspend bit (DTSCTL1.DTSUST). If it is desired to have the DTS accept another DTS transfer request after the abort, clear the DTS suspend bit.

Figure 8.11 shows an example of suspension, resume, and transfer abort of a DTS transfer.

In **Figure 8.11**, channels 0, 1, and 2 are executing block transfer. At time tick 1, a DTS transfer request for channel 1 is accepted and DTS transfer starts. At time tick 2, DTS transfer requests for channels 0 and 2 are generated. At time tick 3, the last transfer of channel 1 is complete, and as a result of DTS channel arbitration, a DTS transfer request for channel 0 is accepted, and DTS transfer of channel 0 starts because channel 0 has a higher priority than channel 2. At time tick 4, the last transfer of channel 0 is complete, and DTS transfer of channel 2 starts. At time tick 5, DTS is put into the suspended state, and the DTS transfer of channel 2 is suspended. At time tick 6, DTS transfer requests for channels 0 and 1 are generated. At time tick 7, the suspended state for DTS is cleared, and the DTS transfer of channel 2, which was suspended in the middle of a block transfer, is resumed. If DTS transfer is suspended in the middle of a block transfer, no DTS channel arbitration is executed when it is resumed. At time tick 8, the last transfer of channel 2 is complete, and as a result of DTS channel arbitration, a DTS transfer request for channel 0 is accepted and the DTS transfer starts because channel 0 has a higher priority than channel 2. At time tick 9, DTS is put into the suspended state, and at time tick 10, the suspended DTS transfer of channel 0 is aborted. When the suspended state of DTS is cleared at time tick 11, DTS transfer of channel 1 starts because there is no currently ongoing DTS transfer and channel 1 is the only channel with a DTS transfer request.

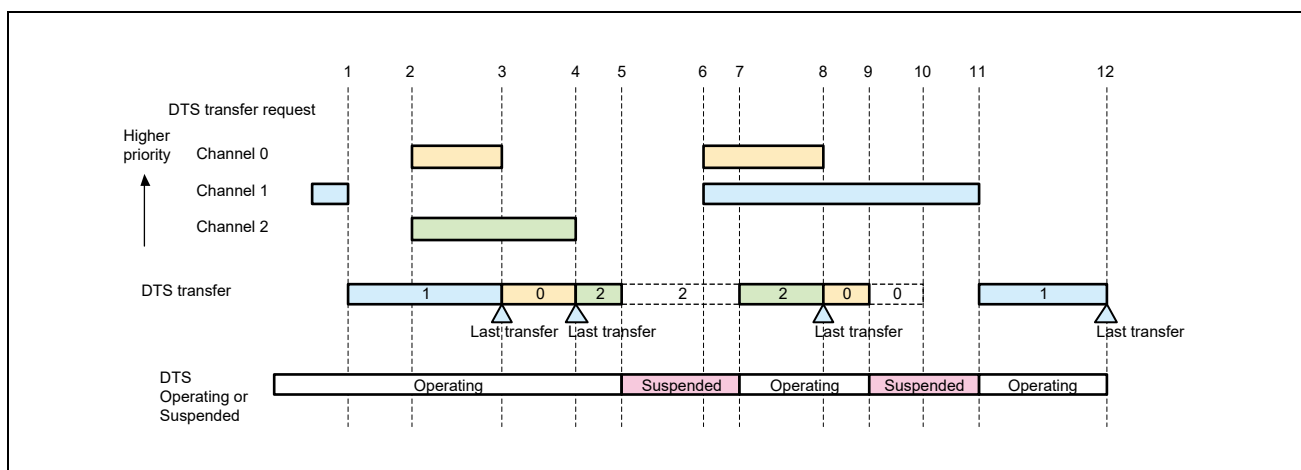


Figure 8.11 Example of Suspension, Resume, and Transfer Abort of a DTS Transfer

8.5.2 Masking and Clearing a Hardware DTS Transfer Request by DTSFSL

A DTS transfer request from a channel to DTS can be temporarily disabled (masked) by clearing the transfer request enable bit (DTFSL_{nnn}.REQEN) in the DTSFSL operation setting register. In this case, the masked channel is excluded from DTSFSL arbitration.

A DTS transfer request held in DTSFSL can be cleared by setting the transfer request clear bit (DTFSC_{nnn}.DRQC) in the DTSFSL transfer request clear register.

DTSFSL monitors hardware transfer source requests regardless of the state of DTS and the transfer request enable bit (DTFSL_{nnn}.REQEN). DTSFSL sets a DTS transfer request for a channel when it detects a hardware transfer source request. When resuming or starting DTS transfer, a hardware DTS transfer request held in DTSFSL should be cleared as required.

8.5.3 List of Suspend, Resume, and Transfer Abort Functions

Table 8.39 List of Suspend, Resume, and Transfer Abort Functions

Function	How to Execute the Function	Operation	Possibility of DTS Transfer Abort
Suspension and resume of DTS	Setting and clearing DTCTL1.DTSUST.	All channels of DTS transfer is suspended.	Possible (by setting DTCTL2.DTSTIT during suspension)

8.6 Error Control

8.6.1 Types of Errors

DTS can generate the following types of errors:

- DTS Transfer Error

This error is generated when an error is detected in the read cycle or write cycle in a DTS cycle or when an error is detected by checking the ECC in the read cycle.

This error can be generated in all DTS channels during execution of DTS transfer.

- DTSRAM ECC Error

This error is generated when an ECC error is detected in the DTSRAM read access by DTS.

This error can be generated in the TI fetch during execution of DTS transfer for DTS or while software is accessing the DTS channel registers.

- DTSRAM Address Feedback Error

This error is generated when an Address Feedback error is detected in the DTSRAM read or write access by DTS.

This error can be generated in the TI fetch or TI writeback during execution of DTS transfer for DTS or while software is accessing the DTS channel registers. See **Section 38, Functional Safety** for details.

8.6.2 DTS Transfer Error

8.6.2.1 Operation of DTS When a DTS Transfer Error Occurs

When a DTS transfer error occurs in DTS, the transfer error flag (DTFSTnnn.ER) in the DTS transfer status register of the channel on which the DTS transfer error occurred is set. The transfer error cycle flag (DTSFSTnnn.ERWR) in the DTS transfer status register of the channel on which the DTS transfer error occurred shows whether the error occurred in the read cycle or the write cycle. The DTS Error register (DTSER) shows the smallest number of DTS channels on which the DTS transfer error occurred.

If a DTS transfer error occurs in a single transfer, a TI writeback is executed to finish the DTS cycle.

If a DTS transfer error occurs in the middle of a block transfer and the DTS error transfer disable bit (DTTCTnnn.ESE) is set, the remaining DTS cycles in the block transfer are not executed, but a TI writeback is executed to finish the DTS cycle. At the same time, the DTS transfer status (DTSSTS.DTSACT) bit in the DTS status register is cleared. If a DTS transfer error occurs in the middle of a block transfer and the DTS error transfer disable bit (DTTCTnnn.ESE) is cleared, the block transfer continues regardless of the DTS transfer error.

If a DTS transfer error occurs during the read cycle of a DTS cycle, the write cycle is not executed. If a DTS transfer error occurs during the write cycle of a DTS cycle, the validity of the write is not guaranteed.

Regardless of whether a DTS transfer error occurs in the read cycle or write cycle of a DTS cycle, the source address, destination address, transfer count, and address reload count registers are updated, and the TI is updated by a TI writeback.

Even if the transfer error flag (DTFSTnnn.ER) in the DTS transfer status register of the channel is set, a TI fetch is executed when DTS accepts a DTS transfer request for the channel. If, as a result of the TI fetch, the DTS error transfer disable bit (DTTCTnnn.ESE) is found to be set, a DTS cycle and a TI writeback are not executed. If the DTS error transfer disable bit (DTTCTnnn.ESE) is cleared, DTS transfer is executed.

8.6.3 DTSRAM Error

There are three types of DTSRAM errors detected in DTSRAM read access: an ECC correctable error, an ECC uncorrectable error and an address feedback error. There is one type of DTSRAM error detected in DTSRAM write access: an address feedback error.

If an ECC correctable 1-bit error is detected during a TI fetch, error corrected data is used, and DTS transfer continues. If an ECC correctable 1-bit error is detected during DTS channel register access from software, error corrected data is returned as read data.

If an ECC uncorrectable error or an address feedback error is detected during a TI fetch, handling of the DTS transfer request is terminated without executing a DTS cycle and TI writeback. If an address feedback error is detected during a TI writeback, the validity of the TI writeback is not guaranteed. If an ECC uncorrectable error or address feedback error is detected during a DTS channel register read from software, a peripheral bus error is reported. If an address feedback error is detected during a DTS channel register write from software, a peripheral bus error is not reported, but the validity of the write access is not guaranteed. In addition to an ECC 2-bit error, an ECC 1-bit error is treated as uncorrectable if 1-bit error correction is disabled in ECC.

8.7 Reliability Functions

8.7.1 Overview

In this product, DTS provides the following reliability functions to support multi-core configurations (multiple master PEs).

- Register access protection function
See **Section 38, Functional Safety**.
- Master information inheritance function
- SPID setting restriction
See **Section 38, Functional Safety**
- Restriction on chain function

8.7.2 Master Information Inheritance Function

The master information inheritance function uses the contents of the channel master setting registers as the sideband information of the data transfer interface.

DTS outputs the sideband information of data transfer for the access guard.

Table 8.40 shows the sideband information that DTS outputs.

Table 8.40 Sideband Information That Is Output from DTS

Meaning	Value Output from DTS
UM	Same as the UM bit value in the channel master setting register
SEC	Non secure
SPID	Same as the SPID bit value in the channel master setting register,

8.7.3 Restriction on Chain Function

The channel configuration of the chain function can be restricted. When chain function restriction is enabled, only a pre-allowed channel in master information can be set as the next channel.

The chain function is not restricted if the CHAIN_RESTRICT bit of the DTS Channel Master Setting Register (DTSnmmCM) is cleared. In this case, DTS channel can specify any other channel to be included in the chain function.

The chain function is restricted if the CHAIN_RESTRICT bit of the DTS Channel Master Setting Register (DTSnmmCM) is set. In this case, only the channels in which CHAIN_SPID[4:0] and CHAIN_UM match can be included into the chain function by DTS channel.

When a chain is executed, DTS compares the channel master settings of the previous and next channels to decide if the chain is permissible.

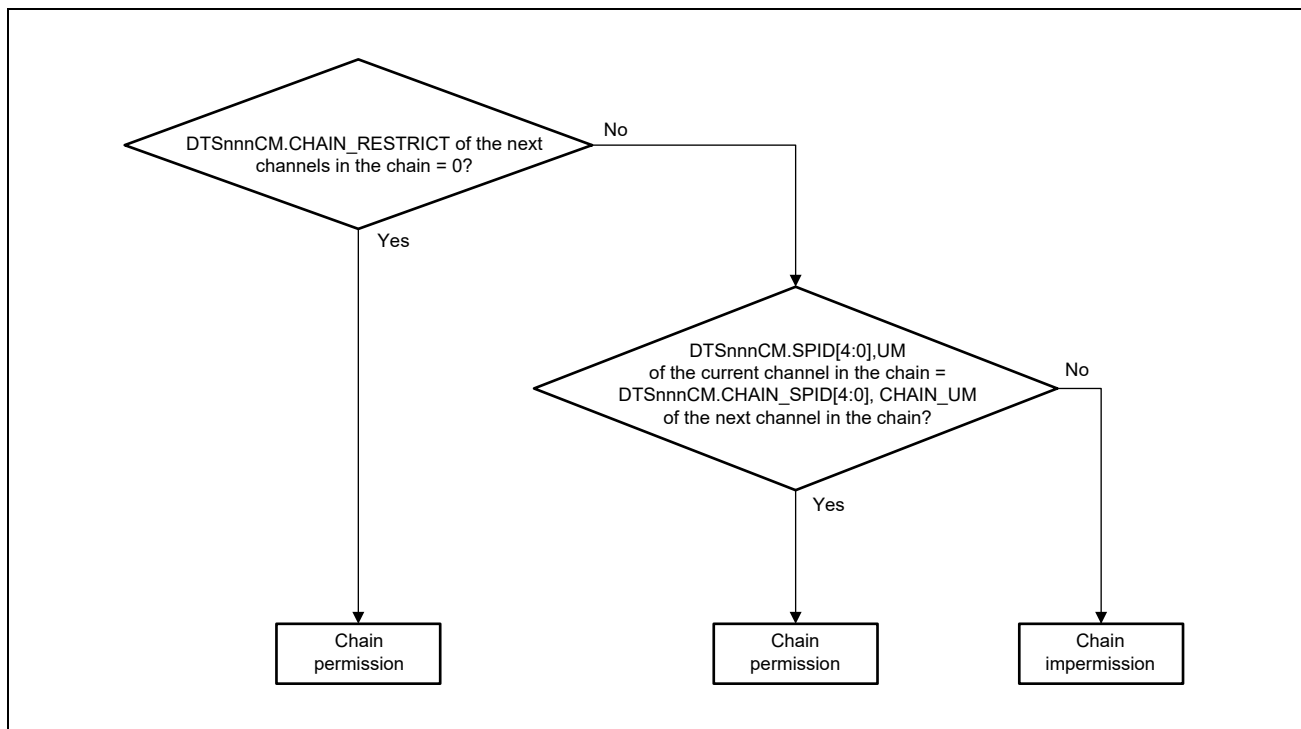


Figure 8.12 Chain Judgement Flow

8.8 Setting up DTS Transfer

8.8.1 Overview of Setting up DTS Transfer

Table 8.41 Overview of Setting up DTS Transfer

Description	Register		Necessity of the Setting
Transfer request group selection setting	DTSSSELm	DTS transfer request group selection	Mandatory (if using a hardware transfer request)
Overall DTS operation setting	DTSPRn	DTS channel priority setting	Mandatory (if DTS is used)
	DTSnnnCM	DTS channel master setting	Mandatory (if DTS is used)
Channel setting	DTSAAnnn	DTS source address	Mandatory
	DTDAnnn	DTS destination address	Mandatory
	DTTCnnn	DTS transfer count	Mandatory
	DTTCTnnn	DTS transfer control	Mandatory
	DTRSAAnnn	DTS reload source address	Mandatory if the reload function is used
	DTRDAAnnn	DTS reload destination address	
	DTRTCnnn	DTS reload transfer count	
	DTTCCnnn	DTS transfer count compare	Mandatory if the transfer count match interrupt is used
Status clear	DTFSCnnn	DTSFSL transfer status clear	Recommended
Transfer request enable	DTFSLnnn	DTSFSL operation setting	Mandatory

8.8.2 Setting up the Transfer Request Group Selection

Set up the transfer request group selection before using DTS. The following registers must be set up to configure the transfer request group.

- DTS Transfer Request group selection Register (DTSSSELm) (m = 0 to 15)
This register configures the transfer request group selection.

8.8.3 Setting up the Overall DTS Operation

Set up the overall DTS operation before using DTS.

The following registers must be set up to configure the overall DTS operation.

- DTS channel priority setting registers (DTSPRn, n = 0 to 7)
These registers configure the priority level of each DTS channel used for DTS channel arbitration.
- DTS channel master setting registers (DTSnnnCM)
These registers configure channel assignment. (For details, see **Section 8.7, Reliability Functions.**)

NOTE

It is important to properly configure the DTS channel master setting registers. Otherwise DTS transfer will not operate as expected.

It is recommended to clear following errors detected while setting up DTS:

- The DTSFSL Transfer Status Register (DTFSTnnn)

NOTE

Channel number information of the channel with error can be obtained by reading the DTS Error Register (DTSER).

8.8.4 Configuring the DTS Channel Settings

The DTS channel settings define the transfer information and transfer source for a DTS channel.

The following procedure shows how to configure the DTS channel settings.

(1) Disable transfer requests by DTSFSL.

Clear the transfer request enable bit (DTFSL_{nnn}.REQEN) in the DTSFSL operation setting register of the DTS channel being configured. This procedure is not mandatory but recommended in order to prevent DTS transfer requests during DTS channel configuration. It is also recommended to check the DTS status register (DTSSTS) and confirm that DTS transfer is not ongoing on the DTS channel being configured.

(2) Configure the transfer information (TI).

The following registers need to be set up to configure DTS channel transfer information.

- DTS Source Address Register (DTSAnnn)
- DTS Destination Address Register (DTDA_{nnn})
- DTS Transfer Count Register (DTTC_{nnn})
- DTS Transfer Control Register (DTTCT_{nnn})
- DTS Reload Source Address Register (DTRSA_{nnn})
- DTS Reload Destination Address Register (DTRDA_{nnn})
- DTS Reload Transfer Count Register (DTRTC_{nnn})
- DTS Transfer Count Compare Register (DTTCC_{nnn})

(3) Set the DTS transfer request.

DTS does not distinguish between hardware and software DTS transfer requests (no specific setting). Both hardware and software DTS transfer requests are retained in the same transfer request pending bit (DTFST_{nnn}.DRQ).

DTSFSL may retain a DTS transfer request that came before the transfer information is set up. If necessary, DTS transfer requests (DTFST_{nnn}.DRQ) retained in DTSFSL can be cleared by using the DTFSL transfer status clear register (DTFSC_{nnn}).

(4) Enable transfer requests by DTSFSL.

Set the transfer request enable bit (DTFSL_{nnn}.REQEN) in the DTSFSL operation setting register to enable DTS transfer requests for the DTS channel.

After the transfer request enable bit for DTSFSL is set, DTS can accept a DTS transfer request and start DTS transfer.

Section 9 RAM

This section describes the RAM mounted on RH850/E2x-FCC1 and E2M.

9.1 List of On-Chip RAMs

Table 9.1 shows the list of On-Chip RAMs.

Table 9.1 List of On-Chip RAMs

		RH850/E2x-FCC1	RH850/E2M
Local RAM per CPU		32 KB	32 KB
Cluster0 RAM	Total Size (Retention RAM included)	512 KB	512 KB
	Retention RAM (Included in Cluster0 RAM)	No	No
Cluster1 RAM	Total Size (Retention RAM included)	192 KB	192 KB
	Retention RAM (Included in Cluster1 RAM)	64 KB	64 KB
Cluster ERAM*1		1 MB	No
Global ERAM*1		1.5 MB	No
Instrumentation RAM*1		64 KB	No

Note 1. All data are retained in Emulation/Instrumentation RAM and Aurora Retention Mode (E2x-FCC1 Only).

9.2 Features

Access:

The CPU0, CPU1, DMA and H-Bus Master can access the Local RAM (CPU0/CPU1), the Cluster 0/1 RAM, the Cluster ERAM, the Global ERAM and the Instrumentation RAM. The access latency from CPU0 to the Local RAM for CPU0 is the same as the access to the Local RAM (self). The access latency from CPU1 to the Local RAM for CPU1 is the same as the access to the Local RAM (self). The access latency from CPU0 or CPU1 to Cluster0 RAM is faster than the access latency to Cluster1 RAM, because CPU0 and CPU1 are in Cluster0.

RAM Retention:

In RH850/E2x-FCC1, all data in the Cluster ERAM, Global ERAM and Instrumentation RAM are retained in Emulation/Instrumentation RAM and Aurora Retention Mode (E2x-FCC1 Only).

In RH850/E2x-FCC1 and E2M, 64 Kbytes of data in the Cluster1 RAM are retained in power-off standby mode.

For details, see **Section 16, Standby Controller**.

ECC:

The Local RAM(CPU0/CPU1) and Cluster RAM include the ECC.

The Local RAM(CPU0/CPU1) and Cluster RAM also include the address feedback function. For details, see **Section 38, Functional Safety**.

9.3 RAM Data Retention

Simultaneous occurrence of write and reset in the RAM space that retains data is not written as undefined data in power-off standby mode.

9.4 Usage Notes

When ECC error detection/correction is enabled for the Local RAM and Cluster RAM, initialize the RAM using a write instruction with the maximum bit length of its access size before using the RAM.

For the maximum bit length of the RAM access size, see the applicable data width in **Section 38, Functional Safety**.

If the RAM is accessed before its initialization, an ECC error may be detected. Also, if initialization with the maximum bit length is not performed (e.g. if a 32-bit RAM is initialized by an 8-bit or 16-bit access), an ECC error may be detected.

Section 10 Reset Controller

10.1 Features

The Reset Controller controls all factors that affect the reset behavior of the device.

The device has several kinds of reset categories depending on the areas that are reset. Each of the reset categories is triggered by one or multiple reset sources.

The relationship between reset categories and their sources is shown in **Table 10.1**.

Table 10.1 Reset Categories and Reset Sources

Reset Category	Source
Power Up Reset	<ul style="list-style-type: none"> Power Up Reset*¹ (by SYSVCC Voltage level) Debugger Reset (forcible reset by a debugger) For details, see Section 45, Debugging and Calibration .
System Reset 1	<ul style="list-style-type: none"> Standby Reset (exiting power off standby mode using $\overline{\text{RES_IN}}$) External Reset (by $\overline{\text{RES_IN}}$) VMON Reset (by Voltage monitor and Delay monitor)
System Reset 2	<ul style="list-style-type: none"> Software Reset (by SWSRESA, SWSRESA_ICUM) ECM Reset (if RESC0 = 0) SWDT Reset (if RESC1 = 0)
Application Reset	<ul style="list-style-type: none"> Software Reset (by SWARESAS, SWARESAS_ICUM) ECM Reset (if RESC0 = 1) SWDT Reset (if RESC1 = 1) <p>NOTE: This reset is basically identical to System Reset 2 with the following exceptions: Clock configuration is preserved, except that of external clock output. Initial setting of peripheral functions using the option byte is not changed. HW BIST is not executed.</p>
Module Reset	<ul style="list-style-type: none"> Software Module Reset Assertion Register for ICU-M (SWMRESAS_ICUM) ICU-M WDTA Reset For details, see Section 43, Intelligent Cryptographic Unit/Master (ICUMD) .
JTAG Reset	JTAG Reset (by $\overline{\text{TRST}}$)

Note 1. External Reset ($\overline{\text{RES_IN}}$) must be asserted low at power up and power down. For the timing details, see *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

10.1.1 Register Base Addresses

Reset base addresses are listed in the following table.

Reset register addresses are given as offsets from the base addresses in general.

Table 10.2 Register Base Addresses

Base Address Name	Base Address	Bus Group
<RESET_base>	FF70 0000 _H	Peripheral Group 6
<BOOT_base>	FFFB 0000 _H	Peripheral Group 0

10.1.2 Input/Output Pins

I/O pins related to reset are shown in **Table 10.3**.

Table 10.3 I/O Pins

Pin Function Name	Direction	Description
$\overline{\text{RES_IN}}$	Input	Reset Input
$\overline{\text{TRST}}$	Input	JTAG Reset Input
$\overline{\text{RES_OUT}}$	Output	Reset Output

10.2 Register Description

10.2.1 Register Protection

Write-protected registers are protected from inadvertent write access due to erroneous program execution, etc. Write-protected registers can be written by releasing the protection of RESKCPROT.

10.2.2 List of Registers

The registers related to reset are listed in **Table 10.4**.

Table 10.4 List of Registers

Unit Name	Register Name	Symbol	Address	Access size	Access Protection
PFSYSTEM	Boot Control Register	BOOTCTRL	<BOOT_base> + 2000 _H	32	
SYSCTRL	Reset Factor Register	RESF	<RESET_base> + 0480 _H	32	
SYSCTRL	Reset Factor Clear Register	RESFC	<RESET_base> + 0484 _H	32	
SYSCTRL	Software System Reset Assertion Register	SWSRESA	<RESET_base> + 0500 _H	32	RESKCPROT
SYSCTRL	Software Application Reset Assertion Register	SWARESAS	<RESET_base> + 0504 _H	32	RESKCPROT
SYSCTRL	Reset Configuration Register	RESC	<RESET_base> + 0580 _H	32	RESKCPROT
SYSCTRL	RAM Initialization Mode Control Register for DTS RAM	STAC_DTS RAM	<RESET_base> + 0880 _H	32	
SYSCTRL	RAM Initialization Mode Control Register for sDMAC Descriptor RAM	STAC_DPR AM	<RESET_base> + 0888 _H	32	
SYSCTRL	RAM Initialization Mode Control Register for DFE	STAC_DFE	<RESET_base> + 088C _H	32	
SYSCTRL	RAM Initialization Mode Control Register for FlexRay	STAC_FLX A	<RESET_base> + 0894 _H	32	
SYSCTRL	RAM Initialization Mode Control Register for Ethernet	STAC_ETH ER	<RESET_base> + 0898 _H	32	
SYSCTRL	RAM Initialization Mode Control Register for GTM	STAC_GTM	<RESET_base> + 089C _H	32	
SYSCTRL	Reset Controller Register Key Code Protection Register	RESKCPROT	<RESET_base> + 0FF0 _H	32	

10.2.3 Reset of Registers

The register reset conditions are shown in **Table 10.5**.

Table 10.5 Register Reset Conditions

Register Name	Reset Condition					
	Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
BOOTCTRL	✓	✓	✓	✓	—	—
RESF	✓*1	✓*1	—	—	—	—
RESFC	✓	✓	✓	✓	—	—
SWSRESA	✓	✓	✓	—	—	—
SWARESAS	✓	✓	✓	—	—	—
RESC	✓	✓	—	—	—	—
STAC_DTSTRAM	✓	✓	✓	—	—	—
STAC_DPRAM	✓	✓	✓	—	—	—
STAC_DFE	✓	✓	✓	—	—	—
STAC_FLXA	✓	✓	✓	—	—	—
STAC_ETHER	✓	✓	✓	—	—	—
STAC_GTM	✓	✓	✓	—	—	—
RESKCPROT	✓	✓	✓	—	—	—

Note 1. For details, see **Section 10.2.5, RESF — Reset Factor Register**.

10.2.4 BOOTCTRL — Boot Control Register

This register controls the startup of each PE. At the time of reset (System Reset 1/2, and Application Reset) release, the PEs are started according to the initial setting of this register, which is defined by a FLASH option byte. After reset release, PEs can be started selectively by asserting the corresponding bit of this register. Only 1 can be written to these bits. Writing 0 is ignored. For details on value of the FLASH option byte, see **Section 41, Flash Memory**.

Value after reset: 0000 000X_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BC1	BC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.6 BOOTCTRL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	BC1	Boot Control 1 This bit triggers the startup of the PE1 (When OPBT3 PE1_DISABLE = 1, PE1 is inactive regardless of the value of this bit) 0: Inactive 1: Active
0	BC0	Boot Control 0 This bit triggers the startup of the PE0 (When OPBT3 STARTUPPE=1 or BC0=1, PE0 is active.) 0: Inactive 1: Active

10.2.5 RESF — Reset Factor Register

This register indicates whether a Reset has occurred since each flag was last cleared.

Value after reset: 0000 0005_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICUMA RESF0	ICUMS RESF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	HWBIS TF	—	ARESF 2	ARESF 1	ARESF 0	—	SRES2 F2	SRES2 F1	SRES2 F0	SRES1 F2	SRES1 F1	SRES1 F0	PRESF 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10.7 RESF Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 18	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
17	ICUMARESFO	ICU-M Software Application Reset Flag This flag is a status flag controlled by ICU-M. For details, see Section 43, Intelligent Cryptographic Unit/Master (ICUMD) . This flag is cleared to 0 by Power Up Reset* ¹ , Standby Reset and VMON Reset. 0: No reset has occurred 1: Reset has occurred
16	ICUMSRESFO	ICU-M Software System Reset Flag This flag is a status flag controlled by ICU-M. For details, see Section 43, Intelligent Cryptographic Unit/Master (ICUMD) . This flag is cleared to 0 by Power Up Reset* ¹ , Standby Reset and VMON Reset. 0: No reset has occurred 1: Reset has occurred
15 to 13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12	HWBISTF	HW BIST Execution Flag This flag will always be asserted after a preceding Power-Up or System Reset if the HW BIST is enabled. In this case, the corresponding Power-Up or System Reset Flag is asserted as well. 0: HW BIST was not executed 1: HW BIST was executed
11	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
10	ARESF2	ECM Application Reset Flag This flag is cleared to 0 by Power Up Reset* ¹ , Standby Reset and VMON Reset. 0: No reset has occurred 1: Reset has occurred
9	ARESF1	SWDT Application Reset Flag This flag is cleared to 0 by Power Up Reset* ¹ , Standby Reset and VMON Reset. 0: No reset has occurred 1: Reset has occurred

Table 10.7 RESF Register Contents (2/2)

Bit Position	Bit Name	Function
8	ARESF0	Software Application Reset Flag This flag is cleared to 0 by Power Up Reset* ¹ , Standby Reset and VMON Reset. 0: No reset has occurred 1: Reset has occurred
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	SRES2F2	ECM System Reset Flag This flag is cleared to 0 by Power Up Reset* ¹ , Standby Reset and VMON Reset. 0: No reset has occurred 1: Reset has occurred
5	SRES2F1	SWDT System Reset Flag This flag is cleared to 0 by Power Up Reset* ¹ , Standby Reset and VMON Reset. 0: No reset has occurred 1: Reset has occurred
4	SRES2F0	Software System Reset Flag This flag is cleared to 0 by Power Up Reset* ¹ , Standby Reset and VMON Reset. 0: No reset has occurred 1: Reset has occurred
3	SRES1F2	VMON Reset Flag This flag is cleared to 0 by Power Up Reset* ¹ and Standby Reset. 0: No reset has occurred 1: Reset has occurred
2	SRES1F1	External Reset Flag* ² 0: No reset has occurred 1: Reset has occurred
1	SRES1F0	Standby Reset Flag This flag is cleared to 0 by Power Up Reset.* ¹ 0: No reset has occurred 1: Reset has occurred
0	PRESF0	Power Up Reset Flag 0: No reset has occurred 1: Reset has occurred This bit is used to judge Power Up Retention RAM data can not be guaranteed with this bit

Note 1. A Power up reset caused by a debugger also clears this flag.

Note 2. SRES1F1 is a high priority flag. Do not use ARESF2-0 and SRES2F2-0 for judgment when SRES1F1=1 at the time of reset factor judgment.

Note 3. HWBISTF is also set at the condition of "BIST not executed" by setting TRST pin to High. Please refer to **Table 38.437** and **Table 38.438** about the condition.

Table 10.8 Reset Factor Register Value in Usecase

Reset Factor	PRES F0	SRES 1F0	SRES 1F1	SRES 1F2	SRES 2F0	SRES 2F1	SRES 2F2	ARES F0	ARES F1	ARES F2	ICUM SRES F0	ICUM ARES F0
Power Up Reset with External Reset	1	0	1	0	0	0	0	0	0	0	0	0
Standby Reset with External Reset	—	1	1	0	0	0	0	0	0	0	0	0
External Reset	—	—	1	—	X	X	X	X	X	X	X	X
VMON Reset	—	—	—	1	0	0	0	0	0	0	0	0
Software Reset (System Reset2)	—	—	—	—	1	—	—	—	—	—	—	—
ICUM Software Reset (System Reset2)	—	—	—	—	—	—	—	—	—	—	1	—
ECM Reset (RESC0=0)	—	—	—	—	—	—	1	—	—	—	—	—
SWDT reset (RESC1=0)	—	—	—	—	—	1	—	—	—	—	—	—
Software Reset (Application Reset)	—	—	—	—	—	—	—	1	—	—	—	—
ICUM Software Reset (Application Reset)	—	—	—	—	—	—	—	—	—	—	—	1
ECM Reset (RESC0=1)	—	—	—	—	—	—	—	—	—	1	—	—
SWDT Reset (RESC1=1)	—	—	—	—	—	—	—	—	1	—	—	—

Note: —: keep, X: Unknown

10.2.6 RESFC — Reset Factor Clear Register

Writing this register clears the reset flags of the RESF register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	HWBIS TFC	—	ARESFC2	ARESFC1	ARESFC0	—	SRES2FC2	SRES2FC1	SRES2FC0	SRES1FC2	SRES1FC1	SRES1FC0	PRESFC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	W	R	W	W	W	R	W	W	W	W	W	W	W

Table 10.9 RESFC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12	HWBISTFC	HW BIST Execution Flag Clear This bit is read as 0. 0: no function 1: Clear flag
11	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
10	ARESFC2	ECM Application Reset Flag Clear This bit is read as 0. 0: no function 1: Clear flag
9	ARESFC1	SWDT Application Reset Flag Clear This bit is read as 0. 0: no function 1: Clear flag
8	ARESFC0	Software Application Reset Flag Clear 0: no function 1: Clear flag
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	SRES2FC2	ECM System Reset Flag Clear This bit is read as 0. 0: no function 1: Clear flag
5	SRES2FC1	SWDT System Reset Flag Clear This bit is read as 0. 0: no function 1: Clear flag
4	SRES2FC0	Software System Reset Flag Clear This bit is read as 0. 0: no function 1: Clear flag

Table 10.9 RESFC Register Contents (2/2)

Bit Position	Bit Name	Function
3	SRES1FC2	VMON Reset Flag Clear This bit is read as 0. 0: no function 1: Clear flag
2	SRES1FC1	External Reset Flag Clear This bit is read as 0. 0: no function 1: Clear flag
1	SRES1FC0	Standby Reset Flag Clear This bit is read as 0. 0: no function 1: Clear flag
0	PRESFC0	Power Up Reset Flag Clear This bit is read as 0. 0: no function 1: Clear flag

10.2.7 SWSRESA — Software System Reset Assertion Register

This register is used to generate a System Reset 2.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWSRESA0_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 10.10 SWSRESA Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SWSRESA0_0	System Reset 2 Trigger This bit is read as 0. 0: No function 1: Generate System Reset 2

10.2.8 SWARESA — Software Application Reset Assertion Register

This register is used to generate an Application Reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWARE SA0_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 10.11 SWARESA Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SWARESA0_0	Application Reset Trigger This bit is read as 0. 0: No function 1: Generate Application Reset

10.2.9 RESC — Reset Configuration Register

This register contains configuration settings for the behavior of the device during reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RESC1	RESC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.12 RESC Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	RESC1	SWDT Reset Configuration 0: SWDT Reset Generates a System Reset 2 1: SWDT Reset Generates an Application Reset
0	RESC0	ECM Reset Configuration 0: ECM Reset Generates a System Reset 2 1: ECM Reset Generates an Application Reset

10.2.10 STAC_DTSTRAM — RAM Initialization Mode Control Register for DTSTRAM

This register is used to control initialization of the DTS RAM. For an Application Reset, initialization of DTSTRAM is executed according to this register.

Value after reset: 0000 000X_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RZERO MD1	RZERO MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.13 STAC_DTSTRAM Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	RZEROMD[1:0]	RAM Initialization Mode for DTS RAM The value after reset can be set by using the flash option byte. X0 _B : Disabled 01 _B : Setting is prohibited 11 _B : Enabled

10.2.11 STAC_DPRAM — RAM Initialization Mode Control Register for sDMAC Descriptor RAM

This register is used to control initialization of the sDMAC Descriptor RAM. For an Application Reset, initialization of sDMAC Descriptor RAM is executed according to this register.

Value after reset: 0000 000X_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RZERO MD1	RZERO MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.14 STAC_DPRAM Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	RZEROMD[1:0]	RAM Initialization Mode for sDMAC Descriptor RAM The value after reset can be set by using the flash option byte. X0 _B : Disabled 01 _B : Setting is prohibited 11 _B : Enabled

10.2.12 STAC_DFE — RAM Initialization Mode Control Register for DFE

This register is used to control initialization of the RAM for DFE. For an Application Reset, initialization of RAM for DFE is executed according to this register.

Value after reset: 0000 000X_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RZERO MD1	RZERO MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.15 STAC_DFE Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	RZEROMD[1:0]	RAM Initialization Mode for DFE The value after reset can be set by using the flash option byte. X0 _B : Disabled 01 _B : Setting is prohibited 11 _B : Enabled

10.2.13 STAC_FLXA — RAM Initialization Mode Control Register for FlexRay

This register is used to control initialization of the RAM for FlexRay. For an Application Reset, initialization of RAM for FlexRay is executed according to this register.

Value after reset: 0000 000X_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RZERO MD1	RZERO MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.16 STAC_FLXA Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	RZEROMD[1:0]	RAM Initialization Mode for FlexRay The value after reset can be set by using the flash option byte. X0 _B : Disabled 01 _B : Setting is prohibited 11 _B : Enabled

10.2.14 STAC_ETHER — RAM Initialization Mode Control Register for Ethernet

This register is used to control initialization of the RAM for Ethernet. For an Application Reset, initialization of RAM for Ethernet is executed according to this register.

Value after reset: 0000 000X_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RZERO MD1	RZERO MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.17 STAC_ETHER Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	RZEROMD[1:0]	RAM Initialization Mode for Ethernet The value after reset can be set by using the flash option byte. X0 _B : Disabled 01 _B : Setting is prohibited 11 _B : Enabled

10.2.15 STAC_GTM — RAM Initialization Mode Control Register for GTM

This register is used to control initialization of the RAM for GTM. For an Application Reset, initialization of RAM for GTM is executed according to this register.

Value after reset: 0000 000X_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RZERO MD1	RZERO MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.18 STAC_GTM Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	RZEROMD[1:0]	RAM Initialization Mode for GTM The value after reset can be set by using the flash option byte. X0 _B : Disabled 01 _B : Setting is prohibited 11 _B : Enabled

10.2.16 RESKCPROT — Reset Controller Register Key Code Protection Register

The RESKCPROT register is used to protect against writing to the registers that may have a material effect on the system so that the application system is not incorrectly stopped due to program malfunction and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31: 16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15: 1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 10.19 RESKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing to protected registers.
Write A5A5A501_H to this register to enable writing to protected registers.

10.3 Operation

10.3.1 Reset Categories

The following are the reset categories of this product.

Each reset category has multiple reset sources (described in the next section). The reset source for each category has a different impact on the system. See the description of resets in the product specifications to determine the exact response of each block to each reset.

Power Up Reset

The Power Up Reset indicates a partial (beginning) power up of the device with SYSVCC. The device begins to function only when a valid SYSVCC is applied while no other power is being supplied. An internal circuit monitors SYSVCC and generates this Power Up Reset signal when SYSVCC transitions to being valid or transitions to being invalid. The external reset pin (RES_IN) must be asserted low during this power up process (when Power Up Reset is negated internally), and during the power down process (when Power Up Reset is asserted internally). Power up Reset can also be triggered by the debugger. All blocks and resources on the device are affected by this reset.

System Reset 1

The System Reset 1 indicates a full power up of the device with all power supplies being applied (excluding EMUVDD and EMUVCC for E2x-FCC1). It also indicates when the external reset (RES_IN) is asserted, even though all of the power supplies remain valid. Internal voltage monitors of all of the supplies can also cause this reset.

For timing details, see *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*. Almost all blocks and resources of this device are affected by this reset. The sources of this reset are as follows:

- Standby Reset
- External Reset
- Voltage Monitor & Delay monitor Reset (VMON Reset)

System Reset 2

System Reset 2 indicates a possible catastrophic software problem that requires a fresh boot of the device.

The sources of this reset are as follows:

- Software Reset by writing the associated register
- Error Control Module Reset (The Reset Category is configured by the Reset Configuration Register)
- Secure Watchdog Timer Reset (The Reset Category is configured by the Reset Configuration Register)

Application Reset

Application Reset is used for fast initialization of the application. This reset is basically identical to System Reset 2 with the following exceptions: Clock configuration is preserved except that of external clock output; the initial setting of peripheral functions using the option byte is not changed; and HW BIST is not executed. This enables fast initialization. Application Reset sources are as follows:

- Software Reset by writing the associated register
- Error Control Module Reset (The Reset Category is configured by the Reset Configuration Register)
- Secure Watchdog Timer Reset (The Reset Category is configured by the Reset Configuration Register)

Module Reset

This is a Software Module Reset for ICUM. For details, see **Section 43, Intelligent Cryptographic Unit/Master (ICUMD)**.

JTAG Reset

JTAG Reset is used for initialization of debug module. For details, see **Section 46, Boundary Scan**.

10.3.2 Reset Sources

Power Up Reset

A power up reset occurs when the SYSVCC power supply becomes valid (within specified limits). An internal power on clear (POC) circuit continuously monitors the voltage level of SYSVCC and asserts an internal power on reset to logic that is powered by SYSVCC. The RESF.PRESF0 bit will be set when this reset occurs.

For details on the POC specifications see *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

Debugger Reset

This device (microcontroller) can be forcibly reset by a debugger.

For details, see **Section 45, Debugging and Calibration**.

Standby Reset

A standby reset occurs when returning from power off standby mode to normal operation mode (or user boot mode or serial programming mode). The RESF.SRES1F0 bit will be set when this condition occurs.

For details on standby functionality, see **Section 16, Standby Controller**.

External Reset (by RES_IN)

A dedicated Reset Input pin ($\overline{\text{RES_IN}}$) is available for initialization.

In case of an open reset input, the device will be initialized via the internal pull-down of the Reset Input Buffer.

In the case of Power-up and Power-down, $\overline{\text{RES_IN}}$ should be low. For details on the required $\overline{\text{RES_IN}}$ low period, see *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

To activate a reset, $\overline{\text{RES_IN}}$ must be asserted low for a time longer than the noise filter. For details on the required $\overline{\text{RES_IN}}$ low pulse width, see *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

VMON Reset

The VMON reset can be asserted as soon as one of the supply voltages is outside the operating range. The execution of the VMON reset can be masked. For details, see **Section 12, Power Supply Voltage Monitor**. The VMON reset operation is independent from External Reset.

ECM Reset

All failures that can be detected by the Error Control Module (ECM) generate dedicated internal resets or interrupts. The failure signals and system behavior can be configured in the ECM (see **Section 39, Error Control Module (ECM)**). Optionally, the device failures can be configured to be used as a Reset Source.

The ECM Reset can generate a System Reset 2 or an Application Reset. The behavior can be configured by the Reset Configuration Register.

SWDT Reset

The Secure Watchdog Timer (SWDT) module can generate a SWDT reset. For details, see **Section 44, Secure Watchdog Timer (SWDT)**.

Software Reset

The device supports multiple software resets that can trigger the reset of dedicated Reset Domains:

- Software System Reset Assertion Register (SWSRESA) triggers a System Reset 2
- Software Application Reset Assertion Register (SWARESA) triggers an Application Reset

Each software reset can be triggered when software writes to the associated Reset Assertion Register.

Each Reset Assertion Register is protected against unintended access by configuring the Key Code protection register. For details, see **Section 38, Functional Safety**.

JTAG Reset

A dedicated Reset Input pin ($\overline{\text{TRST}}$) is available for initialization of the Boundary Scan and debug function.

The Boundary Scan and Debug function are described in **Section 45, Debugging and Calibration** and **Section 46, Boundary Scan**.

The JTAG Reset is active low.

10.3.3 Reset Flags

Any reset root cause can be identified by SW in the Reset Factor Register (RESF). Each reset source will be indicated by one bit when it occurs. The status can be read out after the reset has been executed.

The method for determining the reset factors is shown in **Figure 10.1**.

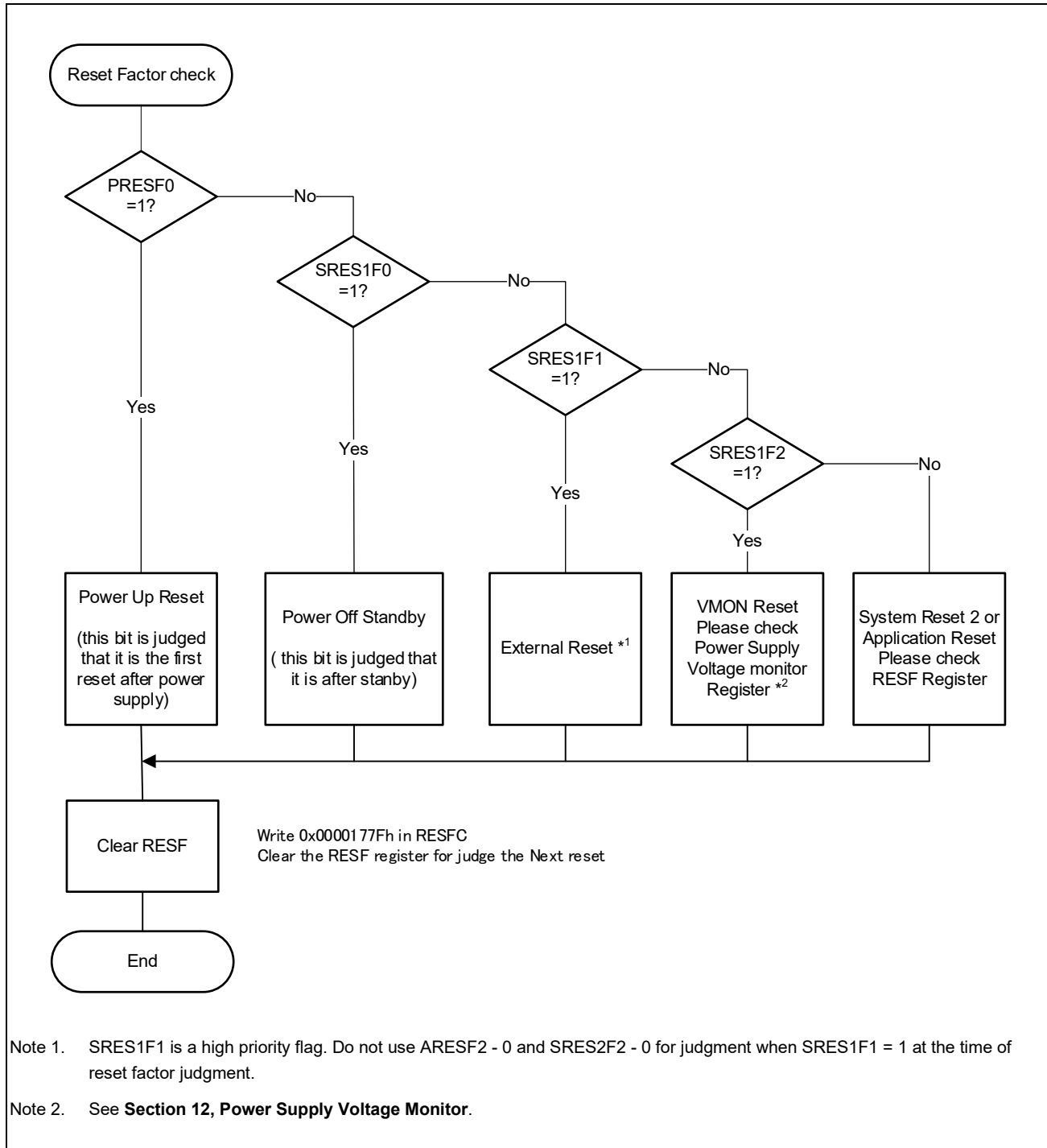


Figure 10.1 Flowchart of Reset Factor Result

10.3.4 FACI Reset Transfer

Transfer of Configuration settings, Block Protection settings, Security settings (Option Bytes) from FLASH memory are executed by System Reset 1 and System Reset 2.

These data become effective with completion of FACI transfer.

If FACI reset transfer Error is detected in Normal operation mode or in User boot mode, the device will be kept in reset state and will not start up.

FACI reset transfer Error is an error factor of the ECM.

10.3.5 HW BIST

HW BIST is executed by System Reset 1 and System Reset 2.

For System Reset 2, HW BIST execution can be disabled. For details, see **Section 38.6, BIST**.

10.3.6 RAM Initialization

RAM data can be initialized (to all 0) according to the setting of a Flash option byte. This initialization, if enabled by the flash option byte, will occur for a system reset 1, a system reset 2, and an application reset (if the value of the STAC register is not changed by using a flash option byte). For an Application Reset and Module Reset, RAM initialization can also be controlled by STAC register. For the Module Reset of ICUM, only RAM data of ICUM can be initialized (to all 0) depending on a STAC register of ICU-M.

A separate flash option byte exists for each major RAM block, so that each block is independently controlled.

Target RAM

- DTS RAM
- sDMAC Descriptor RAM
- DFE RAM
- FlexRay RAM
- Ethernet RAM
- GTM RAM

Local RAM and Cluster RAM are initialized by using a software write instruction.

10.3.7 Reset Output ($\overline{\text{RES_OUT}}$)

A dedicated Reset Output pin $\overline{\text{RES_OUT}}$ is available for indicating when the microcontroller is in reset and for controlling the reset of external devices. It will be asserted by all reset categories (except Module Reset and JTAG Reset).

For the $\overline{\text{RES_OUT}}$ function, refer to **Section 2.6.8, $\overline{\text{RES_OUT}}$ Function**.

10.4 Usage Notes

10.4.1 RAM and Register Data Retention when External Reset Is Asserted

The data of RAM and registers*¹ will not be retained if the operating modes differ when an external reset is asserted or released by a change of the value set to the corresponding external terminal (e.g. MD0, TRST).

Note 1. Registers that are not initialized by External Reset.

Table 10.20 RAM and Register Data Retention when External Reset Is Asserted

Release Timing	Assert Timing		
	Normal Operation Mode	User Boot Mode	Serial Programming Mode
Normal Operation Mode	Retained	Retained	Not retained
User Boot Mode	Retained	Retained	Not retained
Serial Programming Mode	Not retained	Not retained	Retained

Note: The data of MBIST target RAM is not retained in BIST. For details about MBIST target, see **Section 38.6.2.10, MBIST1FTAG0**, **Section 38.6.2.11, MBIST1FTAG1**, and **Section 38.6.2.12, MBIST1FTAG2**.

10.4.2 VMON Reset Assertion

The data of RAM and registers*¹ will not be retained when the VMON reset is asserted.

Note 1. Registers are not initialized by VMON Reset.

Section 11 Power Supply

11.1 Features

This section describes the external voltage connection and internal voltage distribution.

11.2 External Pin List

Table 11.1 shows the list of external pins.

Table 11.1 List of External Pins

Power Supply Name	Power Related Pin Name	Power Combination Condition		Usage
		Operational Voltage Range (3 supply)	Operational Voltage Range (VDD & 5 V)	
E0VCC		4.5 V to 5.5 V	4.5 V to 5.5 V	Port IOs The same power source can be used with each power supply that uses the same voltage range. Refer to Figure 11.1 See the <i>RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware</i> for the supported functions with each voltage range.
E1VCC		3.0 V to 3.6 V or 4.5 V to 5.5 V		
E2VCC		3.0 V to 3.6 V or 4.5 V to 5.5 V		
LVDVCC		3.0 V to 3.6 V or 4.5 V to 5.5 V	4.5 V to 5.5 V	RHSIF LVDS outputs. The power source of LVDVCC should be the same with E0VCC while the GPIO shared with RHSIF is used. Refer to Figure 11.2 .
SYSVCC		3.0 V to 3.6 V	4.5 V to 5.5 V	System logics, retention RAM
VCC		3.0 V to 3.6 V	4.5 V to 5.5 V	Oscillator, PLL, Flash programming
	RAMSVCL	—	—	Stabilizing capacitance for retained RAM
EMUVCC*1		3.0 V to 3.6 V	3.0 V to 3.6 V	ERAM / Instrumentation RAM, Aurora
EMUVDD*1		1.2 V to 1.3 V	1.2 V to 1.3 V*2	ERAM / Instrumentation RAM, Aurora
VDD		1.2 V to 1.35 V	1.2 V to 1.35 V	Core logics
A0VCC		4.5 V to 5.5 V	4.5 V to 5.5 V	SAR-ADC*3
A1VCC		4.5 V to 5.5 V	4.5 V to 5.5 V	*3
A2VCC		4.5 V to 5.5 V	4.5 V to 5.5 V	*3
A3VCC		4.5 V to 5.5 V	4.5 V to 5.5 V	*3
	A0VREFH	4.5 V to 5.5 V	4.5 V to 5.5 V	Reference voltage for SAR-ADC
	A1VREFH	4.5 V to 5.5 V	4.5 V to 5.5 V	
	A2VREFH	4.5 V to 5.5 V	4.5 V to 5.5 V	
	A3VREFH	4.5 V to 5.5 V	4.5 V to 5.5 V	
ADSVCC		4.5 V to 5.5 V	4.5 V to 5.5 V	DS-ADC, and Cyclic-ADC*3
	ADSVREFH	4.5 V to 5.5 V	4.5 V to 5.5 V	Reference voltage for power for DS-ADC, and Cyclic-ADC
	ADSVCL	—	—	Stabilizing capacitance for DS-ADC operation

- Note 1. Only E2x-FCC1 supports EMUVCC, EMUVDD.
- Note 2. Only when using Aurora. EMUVDD max = 1.35 V when not using Aurora.
- Note 3. The same power source shall be used with each A0VCC, A1VCC, A2VCC, A3VCC, and ADSVCC power supply.
See the application note separately issued.

11.3 Example of Power Supply Connection

11.3.1 Example of Power Supply Connection

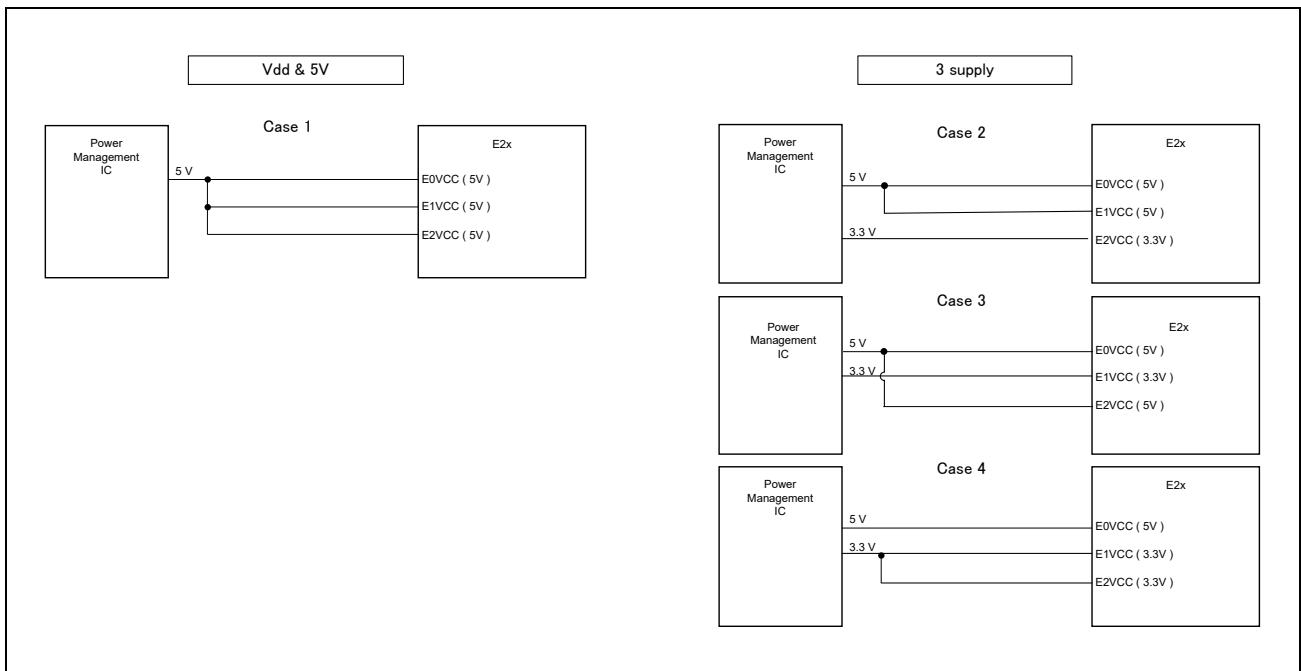


Figure 11.1 Combinations of power connection about EnVCC (n = 0 to 2)

The power source connection policy for EnVCC (n = 0 to 2):

In the case of VDD and 5V, the power source shall be connected as in Case 1.

In the case of 3 power supplies, the power source shall be connected as in Cases 2 to 4.

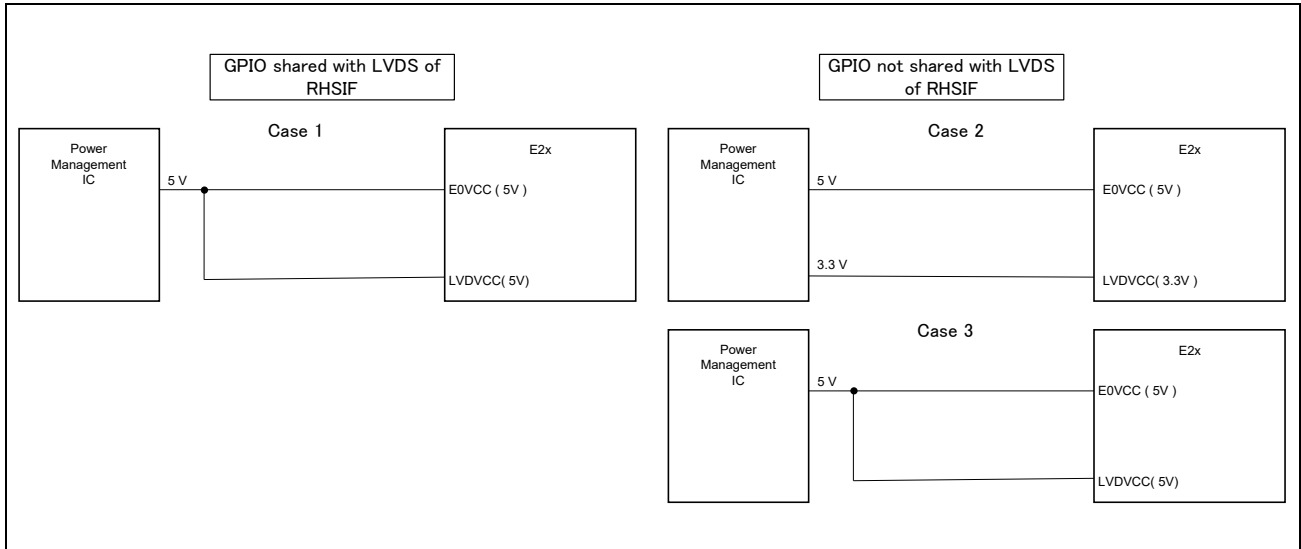


Figure 11.2 Combinations of the power connection about E0VCC and LVDVCC

The power source connection policy about LVDVCC that concerns E0VCC.

i) Case to use GPIO shared with LVDS of RHSIF

The power source of LVDVCC should be the same as the source of E0VCC if the user uses the GPIO shared with the LVDS of RHSIF. The combinations of the power connection are restricted as Case 1 in **Figure 11.2**.

ii) Case NOT to use GPIO shared with the LVDS of RHSIF

The power source of LVDVCC can be set independently of the source of E0VCC (Case 2 or Case 3).

11.4 Power Up/Down Timing

See details in the *RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

11.5 Usage Note

The handling of the power supply pins is described in this section.

For the handling of unused pins, see **Section 2, Pin Function**.

Table 11.2 List of External Pins

Power Supply Name	Power Related Pin Name	Required Setting	Recommended Handling
EnVCC (n = 0 to 2)			Connect an external capacitor to suit the conditions of usage.
SYSVCC		Decoupling capacitor: 0.1 μF $\pm 30\%$	
VCC		Stabilizing capacitor for power: 10 μF $\pm 30\%$ or higher	Decoupling capacitor for each pin
	RAMSVCL	Stabilizing capacitor for power: 0.22 μF $\pm 30\%$	See the application note separately issued.
EMUVCC*1		When Aurora IF is used. Low pass filter: Cut off frequency is lower than 100 kHz including the inductor and the 2.2 μF $\pm 20\%$ or higher capacitor. When Aurora IF is not used. Decoupling capacitor: 0.1 μF $\pm 30\%$	Refer to Figure 11.3 as an example.
EMUVDD*1		When Aurora IF is used. Decoupling capacitors: 0.1 μF $\pm 30\%$ and 0.01 μF $\pm 30\%$ When Aurora IF is not used. Decoupling capacitors: 0.1 μF $\pm 30\%$	Refer to Figure 11.3 as an example.
VDD		Decoupling capacitors: 0.1 μF $\pm 20\%$ and 10 μF $\pm 20\%$	See the application note separately issued.
LVDVCC			See the application note separately issued.
AnVCC (n = 0 to 3)			See the application note separately issued.
	AnVREFH (n = 0 to 3)		See the application note separately issued.
ADSVCC			See the application note separately issued.
	ADSVREFH		See the application note separately issued.
	ADSVCL		See the application note separately issued.

Note 1. Only E2xFCC1 supports EMUVCC, EMUVDD.

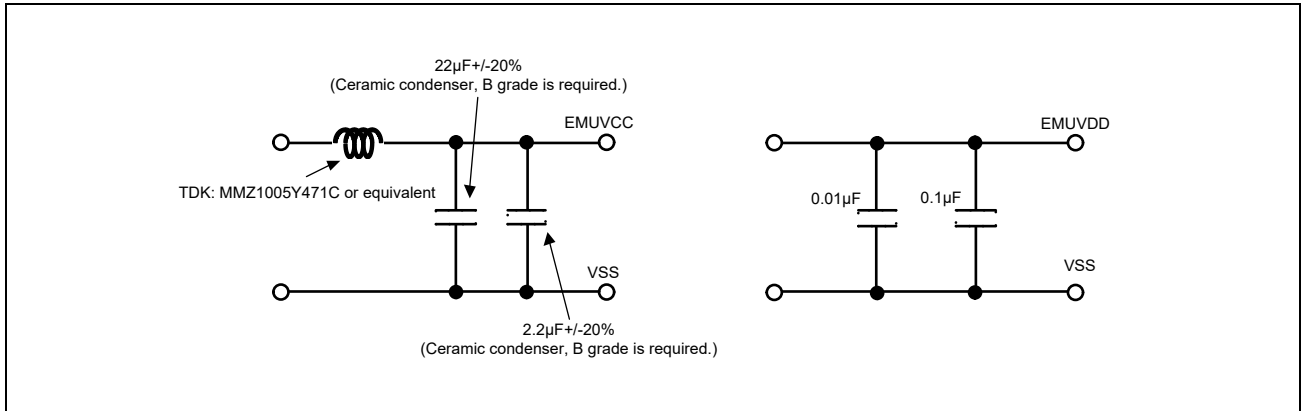


Figure 11.3 Filtering and Bypass Condenser Sample Design for EMUVCC and EMUVDD in case when Aurora IF is used

Section 12 Power Supply Voltage Monitor

12.1 Overall Configuration

12.1.1 Features

- The power supply voltage monitor is used for monitoring the domains E0VCC (hereafter called EVCC in this section), VCC and VDD.
The power supply voltage monitor has H-side (HDET) and L-side (LDET) voltage detectors, which detect if the monitored voltage is higher or lower than the specified voltage. See **Figure 12.1** and **Figure 12.2** for the detection level.
- The power supply voltage monitor has two types of detection functions: Primary detection and Secondary detection
- Primary detection is conducted by Voltage Monitor. For details, refer to **Section 12.2, Primary Detection of Voltage Monitor**.
- Secondary detection is conducted by SAR-ADC. For details, refer to **Section 34, Analog to Digital Converter (ADCH)**.
- Delay Monitor (DMON) assists VMON which detects the L-side voltage of VDD. For details, refer to **Section 12.2.9.2, VCC Primary Low Detection**.
- Delay Monitor (DMON) assists the power supply voltage monitor (VMON) to detect the low level voltage of VDD. For details, refer to **Section 12.3, Delay Monitor (DMON)**.
- Primary power supply voltage monitor can be set ON/OFF for the VMON reset high level and low level detection of VDD, with a high level of VCC and EVCC. Low level detection of VCC and EVCC always raises VMON reset.
- Primary detection voltage value is fixed, secondary detection voltage value can be set by the SAR-ADC.
- If the Secondary power supply voltage monitor detects voltage failures, it notifies the error to the ECM. In this case, it is possible to generate an interrupt or reset by setting the ECM. For details, refer to **Section 39, Error Control Module (ECM)**.
Figure 12.3 shows the construction of the notification with primary and secondary voltage monitor.

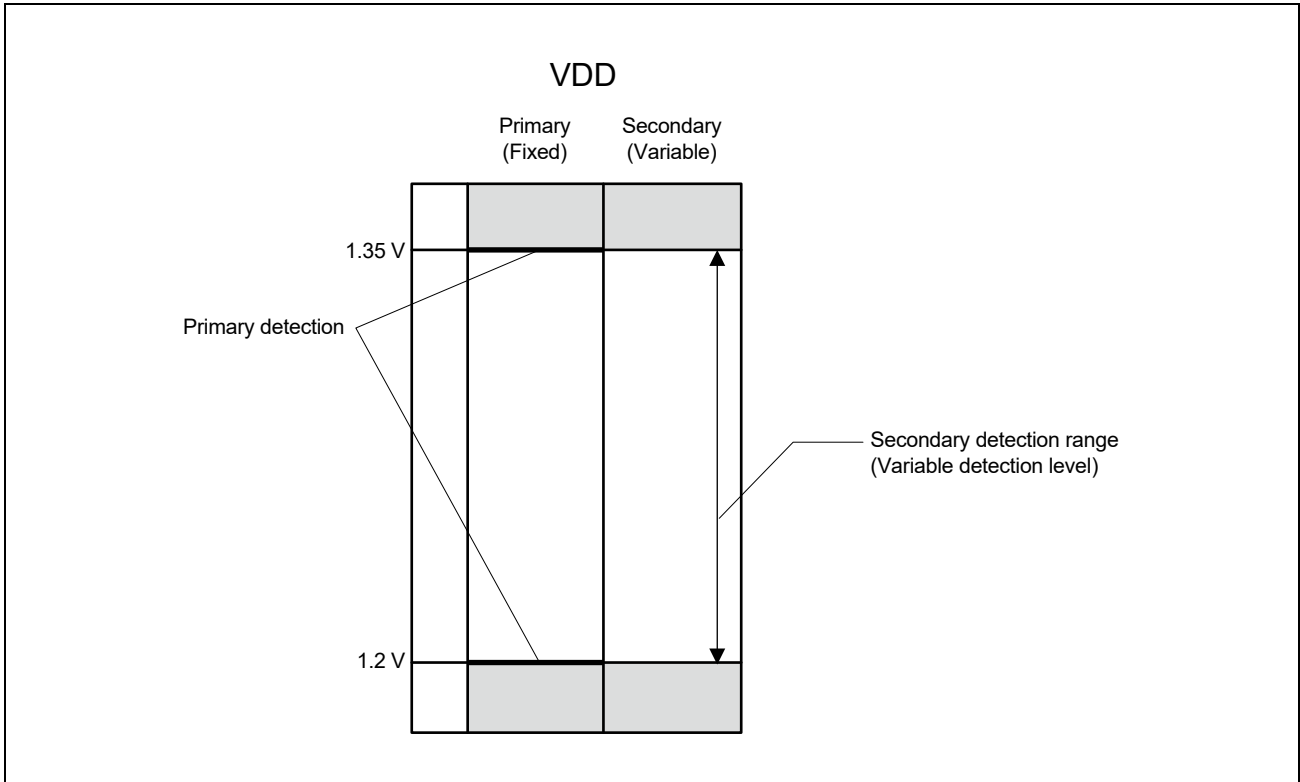


Figure 12.1 VMON Detection Level (VDD)

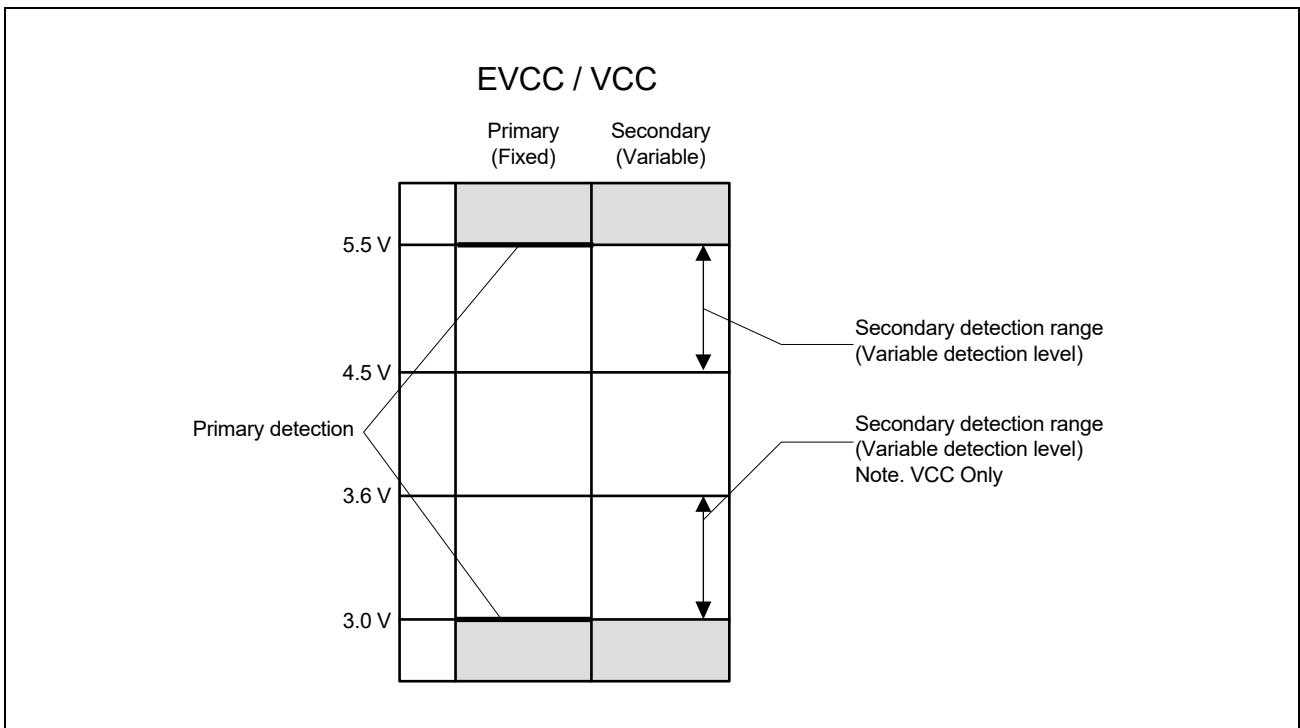


Figure 12.2 VMON Detection Level (EVCC / VCC)

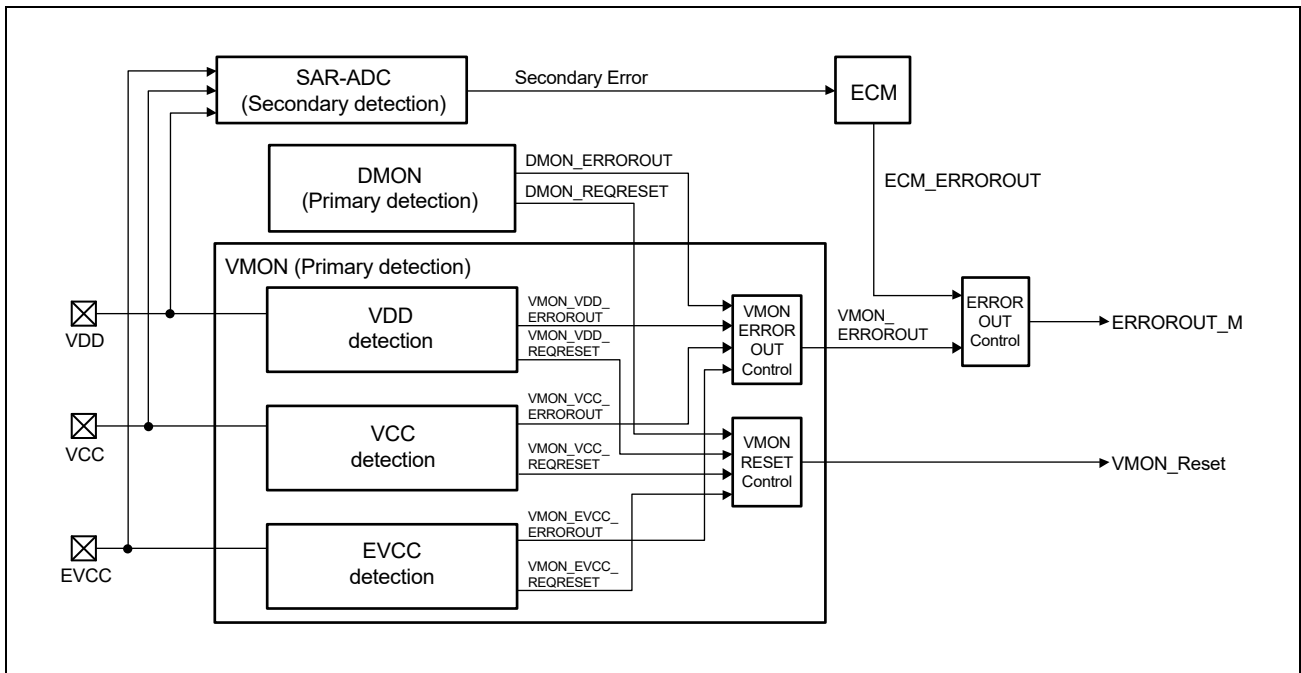


Figure 12.3 Block Diagram of Power Supply Voltage Monitor (Overall)

12.2 Primary Detection of Voltage Monitor (VMON)

12.2.1 Features

- Violation of the operating range of each voltage is indicated by:
 - Error output pin ($\overline{\text{ERROROUT_M}}$)
 - Each power supply voltage high and low-voltage flags
 - VMON Reset
- Diagnostic function:
 - VMON function is testable.
 - High and low-voltage errors can be generated without influencing the voltage itself.
 - VMON error test is done by changing the reference voltage.
 - The signal path to the $\overline{\text{ERROROUT_M}}$ pin can optionally be masked.

12.2.2 Register Base Address

The voltage Monitor base address is given in the following table.

Voltage Monitor register addresses are given as offsets from the base addresses in general.

Table 12.1 Register Base Address

Base Address Name	Base Address	Bus Group
<VMON_base>	FF70 1800 _H	Peripheral Group 6

12.2.3 Clock Supply

The clock supply to VMON is shown in **Table 12.2**.

Table 12.2 Clock Supply

Unit Name	Unit Clock Name	Clock Name
VMON	Digital noise filter clock	CLK_HVIOSC
	Register access clock	CLK_LSB

12.2.4 Interrupt Requests

This module has no interrupt requests.

12.2.5 External Input and Output Pins

The I/O pin related to reset is given in **Table 12.3**.

Table 12.3 I/O Pin

Pin Function Name	Direction	Description
$\overline{\text{ERROROUT_M}}$	Output	VMON error detection output, ECM error output

12.2.6 Overview

12.2.6.1 Block Diagram

Figure 12.4, Figure 12.5 and Figure 12.6 show the block diagram of the power supply voltage monitor.

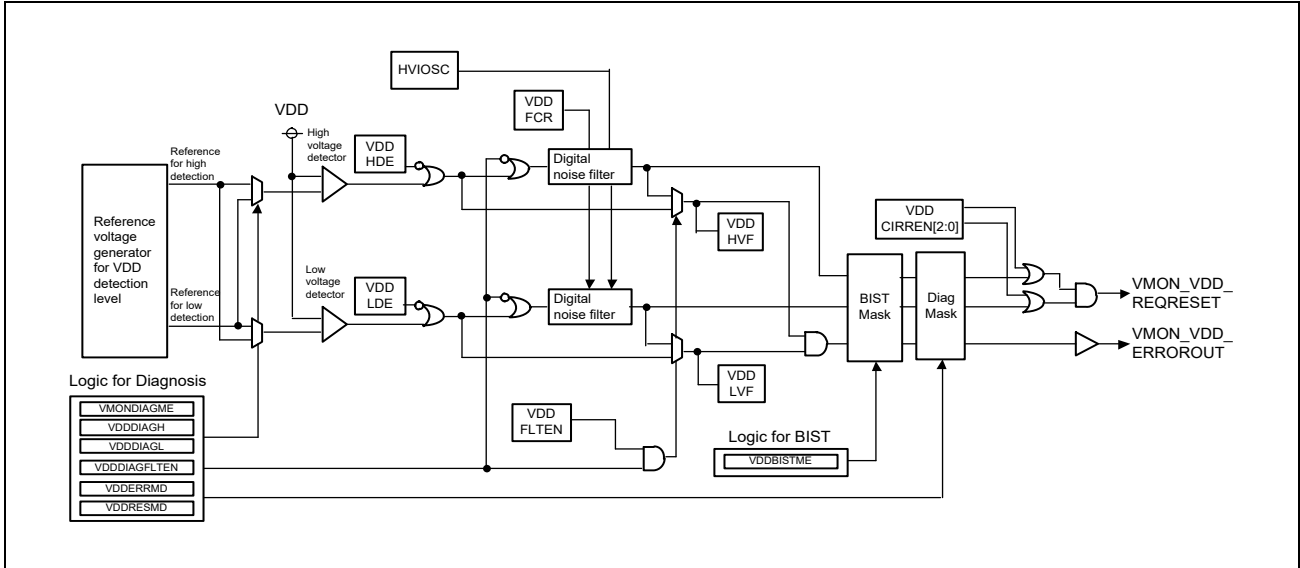


Figure 12.4 Block Diagram of the Power Supply Voltage Monitor (VDD)

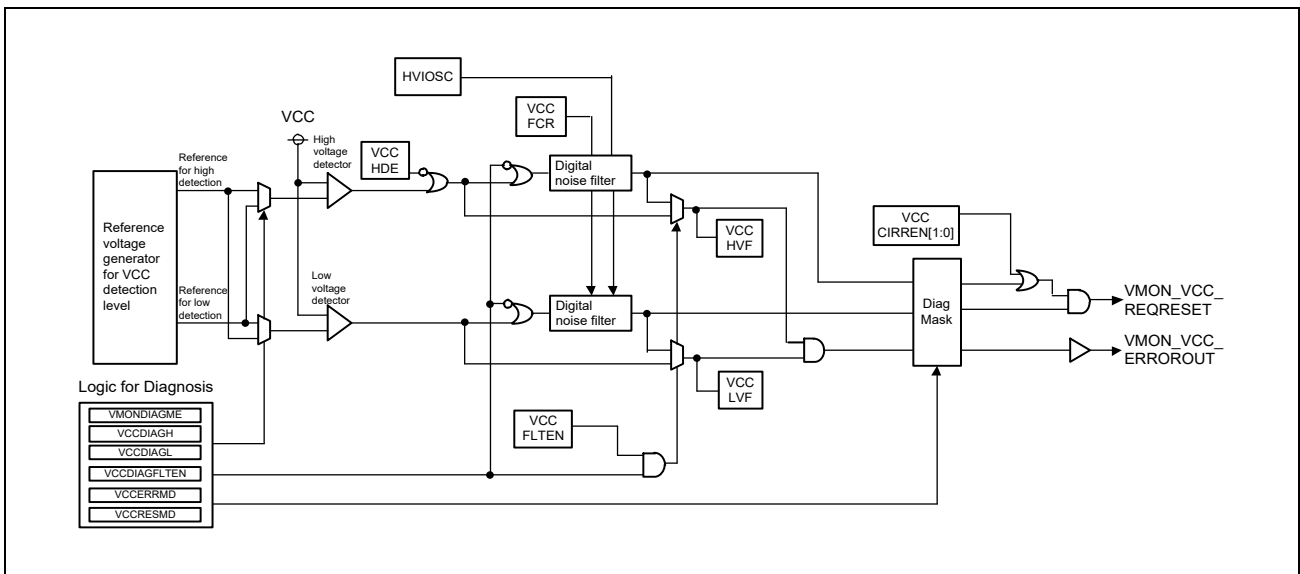


Figure 12.5 Block Diagram of the Power Supply Voltage Monitor (VCC)

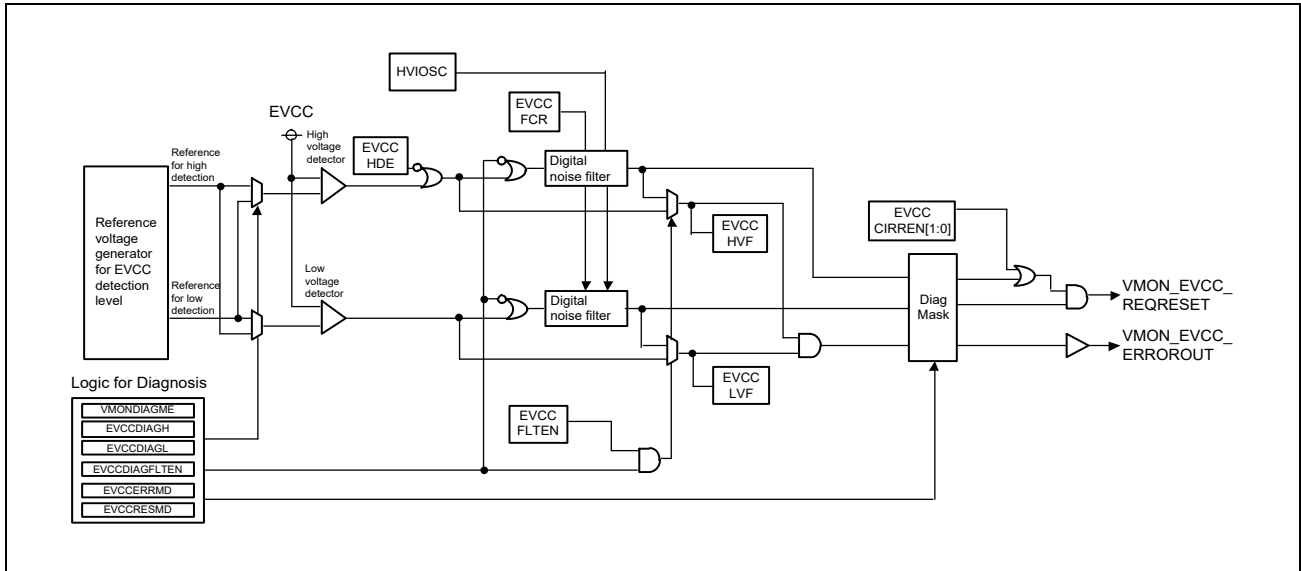


Figure 12.6 Block Diagram of the Power Supply Voltage Monitor (EVCC)

12.2.7 Registers

12.2.7.1 List of Registers

The list of registers related to VMON is shown in **Table 12.4**.

Table 12.4 List of Registers

Unit Name	Register Name	Symbol	Address	Access Size	Access Protection
VMON	VMON Factor Register	VMONF	<VMON_base>+ 000 _H	8	—
VMON	VMONF Clear Register	VMONFC	<VMON_base>+ 004 _H	8	VMONKCPROT
VMON	VMON DIAG Monitor Enable Register	VMONDIAGME	<VMON_base>+ 008 _H	8	—
VMON	VMONDIAGME Write Register* ¹	VMONDIAGMEW	<VMON_base>+ 00C _H	8	VMONKCPROT
VMON	VMON Detection Output Diagnosis MASK Register	VMONDMASK	<VMON_base>+ 010 _H	8	—
VMON	VMON DIAG Mode Setting Register	VMONDIAG	<VMON_base>+ 014 _H	8	—
VMON	VMON DIAG Filter Enable Register	VMONDIAGFE	<VMON_base>+ 018 _H	8	—
VMON	VDD Detection Enable Register	VDDDE	<VMON_base>+ 080 _H	8	VMONKCPROT
VMON	VDD Monitor Filter Control Register	VDDFCR	<VMON_base>+ 084 _H	8	—
VMON	VCC Detection Enable Register	VCCDE	<VMON_base>+ 100 _H	8	VMONKCPROT
VMON	VCC Monitor Filter Control Register	VCCFCR	<VMON_base>+ 104 _H	8	—
VMON	EVCC Detection Enable Register	EVCCDE	<VMON_base>+ 180 _H	8	VMONKCPROT
VMON	EVCC Monitor Filter Control Register	EVCCFCR	<VMON_base>+ 184 _H	8	—
VMON	VMON Register Key Code Protection Register	VMONKCPROT	<VMON_base>+ 300 _H	32	—

Note 1. Writing is permitted only once after Standby Reset or External Reset.

Register reset conditions are shown in **Table 12.5**.

Table 12.5 Register Reset Conditions

Register Name	Reset Source					
	Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
VMONF	✓	✓*2	—	—	—	—
VMONFC	✓	✓	—	—	—	—
VMONDIAGME	✓	✓*1	—	—	—	—
VMONDIAGMEW	✓	✓	—	—	—	—
VMONDMASK	✓	✓	—	—	—	—
VMONDIAG	✓	✓	—	—	—	—
VMONDIAGFE	✓	✓	—	—	—	—
VDDDE	✓	✓*1	—	—	—	—
VDDFCR	✓	✓*1	—	—	—	—
VCCDE	✓	✓*1	—	—	—	—
VCCFCR	✓	✓*1	—	—	—	—
EVCCDE	✓	✓*1	—	—	—	—
EVCCFCR	✓	✓*1	—	—	—	—
VMONKCPROT	✓	✓	✓	—	—	—

Note 1. VMON Reset is excluded.

Note 2. Standby reset only

12.2.7.2 VMONF — VMON Factor Register

VMONF register indicates that an error is detected by VMON. Each flag of VMONF can be cleared individually by writing “1” to the corresponding bit in VMONFC.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	EVCCCHVF	EVCCCLVF	VCCHVF	VCCLVF	VDDHVF	VDDLVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 12.6 VMONF Register Contents

Bit Position	Bit Name	Function
7 to 6	—	Reserved When read, the value after reset is returned.
5	EVCCCHVF	EVCC high voltage detection flag 0: No high EVCC voltage violation detected 1: High EVCC voltage violation occurred
4	EVCCCLVF	EVCC low voltage detection flag 0: No low EVCC voltage violation detected 1: low EVCC voltage violation occurred
3	VCCHVF	VCC high voltage detection flag 0: No high VCC voltage violation detected 1: High VCC voltage violation occurred
2	VCCLVF	VCC low voltage detection flag 0: No low VCC voltage violation detected 1: low VCC voltage violation occurred
1	VDDHVF	VDD high voltage detection flag 0: No high VDD voltage violation detected 1: High VDD voltage violation occurred
0	VDDLVF	VDD low voltage detection flag 0: No low VDD voltage violation detected 1: low VDD voltage violation occurred

12.2.7.3 VMONFC — VMONF Clear Register

VMONFC is a register to clear the VMONF register. The read value of this register is always 00_H.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	EVCCHVFC	EVCLVFC	VCCHVFC	VCCLVFC	VDDHVFC	VDDLFC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	W	W	W	W	W	W

Table 12.7 VMONFC Register Contents

Bit Position	Bit Name	Function
7 to 6	—	Reserved When writing, write the value after reset.
5	EVCCHVFC	Clear EVCC high voltage detection flag 0: Writing 0 has no effect 1: Writing 1 clears EVCCHVF
4	EVCLVFC	Clear EVCC low voltage detection flag 0: Writing 0 has no effect 1: Writing 1 clears EVCLVF
3	VCCHVFC	Clear VCC high voltage detection flag 0: Writing 0 has no effect 1: Writing 1 clears VCCHVF
2	VCCLVFC	Clear VCC low voltage detection flag 0: Writing 0 has no effect 1: Writing 1 clears VCCLVF
1	VDDHVFC	Clear VDD high voltage detection flag 0: Writing 0 has no effect 1: Writing 1 clears VDDHVF
0	VDDLFC	Clear VDD low voltage detection flag 0: Writing 0 has no effect 1: Writing 1 clears VDDLVF

12.2.7.4 VMONDIAGME — VMON DIAG Monitor Enable Register

VMONDIAGME is a read only register to control the VMON DIAG function.

This register reflects the settings written to VMONDIAGMEW.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	VMONDIAGME
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 12.8 VMONDIAGME Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned.
0	VMONDIAGME	Permit DIAG function 0: DIAG function of the Voltage Monitor can be enabled 1: DIAG function of the Voltage Monitor cannot be enabled

12.2.7.5 VMONDIAGMEW — VMONDIAGME Write Register

VMONDIAGMEW is a register to set values of VMONDIAGME. Writing is permitted only once after Standby Reset or External Reset are released. Subsequent write operations are ignored. The read value of this register is always 00_H.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	VMON DIAGMEW
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 12.9 VMONDIAGMEW Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When writing, write the value after reset.
0	VMONDIAGMEW	The data written in this bit is set to VMONDIAGME.

12.2.7.6 VMONDMASK — VMON Detection Output Diagnosis Mask Register

VMONDMASK is a register to mask VMON_ERROROUT and VMON Reset when VMONDIAGME = 0.

The VMONDMASK setting is ignored if VMONDIAGME.VMONDIAGME = 1.

Writing to VMONDMASK is permitted during the period from External Reset release until writing to VMONDIAGMEW.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	EVCCRESMD	VCCRESMD	VDDRESMD	—	EVCCERRMD	VCCERRMD	VDDERRMD
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 12.10 VMONDMASK Register Contents

Bit Position	Bit Name	Function
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	EVCCRESMD	This bit masks VMON_EVCC_REQ_RESET when VMONDIAGME.VMONDIAGME = 0. 0: VMON_EVCC_REQ_RESET is not masked. 1: VMON_EVCC_REQ_RESET is masked.
5	VCCRESMD	This bit masks VMON_VCC_REQ_RESET when VMONDIAGME.VMONDIAGME = 0. 0: VMON_VCC_REQ_RESET is not masked. 1: VMON_VCC_REQ_RESET is masked.
4	VDDRESMD	This bit masks VMON_VDD_REQ_RESET when VMONDIAGME.VMONDIAGME = 0. 0: VMON_VDD_REQ_RESET is not masked. 1: VMON_VDD_REQ_RESET is masked.
3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	EVCCERRMD	This bit masks VMON_EVCC_ERROROUT when VMONDIAGME.VMONDIAGME = 0. 0: VMON_EVCC_ERROROUT is not masked. 1: VMON_EVCC_ERROROUT is masked.
1	VCCERRMD	This bit masks VMON_VCC_ERROROUT when VMONDIAGME.VMONDIAGME = 0. 0: VMON_VCC_ERROROUT is not masked. 1: VMON_VCC_ERROROUT is masked.
0	VDDERRMD	This bit masks VMON_VDD_ERROROUT when VMONDIAGME.VMONDIAGME = 0. 0: VMON_VDD_ERROROUT is not masked. 1: VMON_VDD_ERROROUT is masked.

12.2.7.7 VMONDIAG — VMON DIAG Mode Setting Register

VMONDIAG forces VMON comparators to output an error. This register is valid only when VMONDIAGME = 0. VMONDIAG's setting is ignored if VMONDIAGME.VMONDIAGME = 1.

Writing to VMONDIAG is permitted during the period from External Reset release until writing to VMONDIAGMEW.

Value after reset: 00H

Bit	7	6	5	4	3	2	1	0
	—	—	EVCCDIAGH	EVCCDIAGL	VCCDIAGH	VCCDIAGL	VDDDIAGH	VDDDIAGL
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.11 VMONDIAG Register Contents

Bit Position	Bit Name	Function
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	EVCCDIAGH	This bit can change the detection level of high voltage detector to low level intentionally when VMONDIAGME.VMONDIAGME = 0. 0: EVCC for high voltage monitors normally. 1: EVCC for high voltage will detect violation condition because detection level changes.
4	EVCCDIAGL	This bit can change the detection level of low voltage detector to high level intentionally when VMONDIAGME.VMONDIAGME = 0. 0: EVCC for low voltage monitors normally. 1: EVCC for low voltage will detect violation condition because detection level changes.
3	VCCDIAGH	This bit can change the detection level of high voltage detector to low level intentionally when VMONDIAGME.VMONDIAGME = 0. 0: VCC for high voltage monitors normally. 1: VCC for high voltage will detect violation condition because detection level changes.
2	VCCDIAGL	This bit can change the detection level of low voltage detector to high level intentionally when VMONDIAGME.VMONDIAGME = 0. 0: VCC for low voltage monitors normally. 1: VCC for low voltage will detect violation condition because detection level changes.
1	VDDDIAGH	This bit can change the detection level of high voltage detector to low level intentionally when VMONDIAGME.VMONDIAGME = 0. 0: VDD for high voltage monitors normally. 1: VDD for high voltage will detect violation condition because detection level changes.
0	VDDDIAGL	This bit can change the detection level of low voltage detector to high level intentionally when VMONDIAGME.VMONDIAGME = 0. 0: VDD for low voltage monitors normally. 1: VDD for low voltage will detect violation condition because detection level changes.

12.2.7.8 VMONDIAGFE — VMON DIAG Filter Enable Register

VMONDIAGFE specifies whether the output filter for ERROROUT is enabled.

This register is valid only when VMONDIAGME = 0. The setting of this register is ignored if VMONDIAGME.VMONDIAGME = 1.

Writing to VMONDIAGFE is permitted during the period from External Reset release until writing to VMONDIAGMEW.

VMON reset does not occur when each bit is set to 0.

Value after reset: 07_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	EVCC DIAGFLTEN	VCC DIAGFLTEN	VDD DIAGFLTEN
Value after reset	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 12.12 VMONDIAGFE Register Contents

Bit Position	Bit Name	Function
7 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	EVCCDIAGFLTEN	Enable output filter for VMON_EVCC_ERROROUT when VMONDIAGME.VMONDIAGME = 0. 0: Disable output filter for VMON_EVCC_ERROROUT 1: Enable output filter for VMON_EVCC_ERROROUT
1	VCCDIAGFLTEN	Enable output filter for VMON_VCC_ERROROUT when VMONDIAGME.VMONDIAGME = 0. 0: Disable output filter for VMON_VCC_ERROROUT 1: Enable output filter for VMON_VCC_ERROROUT
0	VDDDIAGFLTEN	Enable output filter for VMON_VDD_ERROROUT when VMONDIAGME.VMONDIAGME = 0. 0: Disable output filter for VMON_VDD_ERROROUT 1: Enable output filter for VMON_VDD_ERROROUT

12.2.7.9 VDDDE — VDD Detection Enable Register

VDDDE is a register to control VMON reset, VMON signal mask and VMON signal filter.

Bits 3, 1, 0 are set by the flash option byte when Standby Reset or External Reset is released.

Writing to VDDDE is permitted during the period from External Reset release until writing to VMONDIAGMEW.

Value after reset: 0X_H

Bit	7	6	5	4	3	2	1	0
	VDDCIRREN[2:0]			VDDFBISTME	VDDFLTEN	—	VDDHDE	VDDLDE
Value after reset	0	0	0	0	X	0	X	X
R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 12.13 VDDDE Register Contents

Bit Position	Bit Name	Function
7 to 5	VDDCIRREN[2:0]	Permit VMON reset by VDD detection. VDDCIRREN 2 to 0 101 _B : VMON reset by VDD high voltage detection is permitted 110 _B : VMON reset by VDD low voltage detection is permitted 111 _B : VMON reset by VDD high/low voltage detection is permitted Other value: VMON reset by VDD detection is not permitted
4	VDDFBISTME	When this bit is set to 0, VMON_VDD_ERROROUT and VMON_VDD_REQRESET are masked when BIST is carried out after VMON Reset. When this bit is set to 1, VMON_VDD_ERROROUT and VMON_VDD_REQRESET are not masked when BIST is carried out after VMON Reset. 0 : During BIST the VMON_VDD_ERROROUT and VMON_VDD_REQRESET can be masked 1 : During BIST the VMON_VDD_ERROROUT and VMON_VDD_REQRESET cannot be masked Note: VMON_VDD_ERROROUT output and VMON_VDD_REQRESET are always masked when BIST is carried out after Standby Reset or External Reset. VMON_VDD_ERROROUT output and VMON_VDD_REQRESET are never masked when BIST is carried out after System Reset 2.
3	VDDFLTEN	Enable output filter for $\overline{\text{ERROROUT_M}}$ and VMONF. 0: Disable output filter for $\overline{\text{ERROROUT_M}}$ and VMONF. 1: Enable output filter for $\overline{\text{ERROROUT_M}}$ and VMONF.
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	VDDHDE	VDD High voltage detection enable 0: Disable VDD high voltage detection 1: Enable VDD high voltage detection
0	VDDLDE	VDD Low voltage detection enable 0: Disable VDD low voltage detection 1: Enable VDD low voltage detection

12.2.7.10 VDDFCR — VDD Monitor Filter Control Register

VDDFCR is a register to adjust the filtering width of VDD Monitor digital noise filter.

This register is set by the flash option byte when Standby Reset or External Reset is released.

Value after reset: 0X_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	VDDFLTW[1:0]		VDDCLKSEL[1:0]	
Value after reset	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R

Table 12.14 VDDFCR Register Contents

Bit Position	Bit Name	Function
7 to 4	—	Reserved When read, the value after reset is returned.
3, 2	VDDFLTW[1:0]	These bits select the minimum filtering width of digital noise filter. (The minimum filtering width is given by cycle-count of the clock selected by VDDCLKSEL1, 0.)
	VDDFLTW[1:0]	Minimum Filtering Width (Cycle)
	00	20 cycles
	01	13 cycles
	10	8 cycles
	11	3 cycle
1, 0	VDDCLKSEL[1:0]	These bits select the clock of the digital noise filter from the following clocks.
	VDDCLKSEL[1:0]	Clock of the Digital Filter
	00	CLK_HVIOSC frequency / 256
	01	CLK_HVIOSC frequency / 128
	10	CLK_HVIOSC frequency / 64
	11	CLK_HVIOSC frequency / 32
CAUTION: The frequency of CLK_HVIOSC is typ. 16 MHz, min. 8 MHz.		

Wait specified time = 10us + Digital noise filter width.

For example, VDDFLTW[1:0] = 00b VDDCLKSEL[1:0] = 00b

Wait specified time = 10us + 20cycle * 256 * 62.5ns = 330us

12.2.7.11 VCCDE — VCC Detection Enable Register

VCCDE is a register to control VMON reset, VMON signal mask and VMON signal filter.

Bits 3, 1 are set by the flash option byte when Standby Reset or External Reset is released.

Writing to VCCDE is permitted during the period from External Reset release until writing to VMONDIAGMEW.

Value after reset: 0X_H

Bit	7	6	5	4	3	2	1	0
	VCCIRREN1	—	VCCIRREN0	—	VCCFLTEN	—	VCCHDE	—
Value after reset	0	0	0	0	X	0	X	0
R/W	R/W	R	R/W	R	R	R	R	R

Table 12.15 VCCDE Register Contents

Bit Position	Bit Name	Function
7	VCCIRREN1	Permit VMON reset by VCC high voltage detection. VCCIRREN1,0 11 _B : VMON reset by VCC high voltage detection is permitted Other value: VMON reset by VCC high voltage detection is not permitted
6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	VCCIRREN0	Refer to bit 7 (VCCIRREN1).
4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	VCCFLTEN	Enable output filter for $\overline{\text{ERROROUT_M}}$ and VMONF. 0: Disable output filter for $\overline{\text{ERROROUT_M}}$ and VMONF. 1: Enable output filter for $\overline{\text{ERROROUT_M}}$ and VMONF.
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	VCCHDE	VCC high voltage detection enable 0: Disable VCC high voltage detection 1: Enable VCC high voltage detection
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

12.2.7.12 VCCFCR — VCC Monitor Filter Control Register

VCCFCR is a register to adjust the filtering width of the VCC Monitor digital noise filter.

This register is set by the flash option byte when Standby Reset or External Reset is released.

Value after reset: 0X_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	VCCFLTW[1:0]		VCCCLKSEL[1:0]	
Value after reset	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R

Table 12.16 VCCFCR Register Contents

Bit Position	Bit Name	Function
7 to 4	—	Reserved When read, the value after reset is returned.
3, 2	VCCFLTW[1:0]	These bits select the minimum filtering width of digital noise filter. (The minimum filtering width is given by the cycle-count of the clock selected by VCCCLKSEL1, 0.)
	VCCFLTW[1:0]	Minimum Filtering Width (Cycle)
	00	20 cycles
	01	13 cycles
	10	8 cycles
	11	3 cycles
1, 0	VCCCLKSEL[1:0]	These bits select the clock of the digital noise filter from the following clocks.
	VCCCLKSEL[1:0]	Clock of the Digital Filter
	00	CLK_HVIOSC frequency / 256
	01	CLK_HVIOSC frequency / 128
	10	CLK_HVIOSC frequency / 64
	11	CLK_HVIOSC frequency / 32
CAUTION: The frequency of CLK_HVIOSC is typ. 16 MHz, min. 8 MHz.		

Wait specified time = 10us + Digital noise filter width.

For example, VCCFLTW[1:0] = 00b VCCCLKSEL[1:0] = 00b

Wait specified time = 10us + 20cycle * 256 * 62.5ns = 330us

12.2.7.13 EVCCDE — EVCC Detection Enable Register

EVCCDE is register to control VMON reset, VMON signal mask and VMON signal filter.

Bits 3 and 1 are set by the flash option byte when Standby Reset or External Reset is released.

Writing to EVCCDE is permitted during the period from External Reset release until writing to VMONDIAGMEW.

Value after reset: 0X_H

Bit	7	6	5	4	3	2	1	0
	EVCCIRREN1	—	EVCCIRREN0	—	EVCCFLTEN	—	EVCCHDE	—
Value after reset	0	0	0	0	X	0	X	0
R/W	R/W	R	R/W	R	R	R	R	R

Table 12.17 EVCCDE Register Contents

Bit Position	Bit Name	Function
7	EVCCIRREN1	Permit VMON reset by EVCC high voltage detection. EVCCIRREN1,0 11 _B : VMON reset by EVCC high voltage detection is permitted Other value: VMON reset by EVCC high voltage detection is not permitted
6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	EVCCIRREN0	Refer to bit 7 (EVCCIRREN1).
4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	EVCCFLTEN	Enable output filter for $\overline{\text{ERROROUT_M}}$ and VMONF. 0: Disable output filter for $\overline{\text{ERROROUT_M}}$ and VMONF. 1: Enable output filter for $\overline{\text{ERROROUT_M}}$ and VMONF.
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	EVCCHDE	EVCC high voltage detection enable 0: Disable EVCC high voltage detection 1: Enable EVCC high voltage detection
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

12.2.7.14 EVCCFCR — EVCC Monitor Filter Control Register

EVCCFCR is a register to adjust the filtering width of the EVCC Monitor digital noise filter.

This register is set by the flash option byte when Standby Reset or External Reset is released.

Value after reset: 0X_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EVCCFLTW[1:0]		EVCCCLKSEL[1:0]	
Value after reset	0	0	0	0	X	X	X	X
R/W	R	R	R	R	R	R	R	R

Table 12.18 EVCCFCR Register Contents

Bit Position	Bit Name	Function
7 to 4	—	Reserved When read, the value after reset is returned.
3, 2	EVCCFLTW[1:0]	These bits select the minimum filtering width of digital noise filter. (The minimum filtering width is given by the cycle-count of the clock selected by EVCCCLKSEL1, 0)
	EVCCFLTW[1:0]	Minimum Filtering Width (Cycle)
	00	20 cycles
	01	13 cycles
	10	8 cycles
	11	3 cycles
1, 0	EVCCCLKSEL[1:0]	These bits select the clock of the digital noise filter from the following clocks.
	EVCCCLKSEL[1:0]	Clock of the Digital Filter
	00	CLK_HVIOSC frequency / 256
	01	CLK_HVIOSC frequency / 128
	10	CLK_HVIOSC frequency / 64
	11	CLK_HVIOSC frequency / 32
CAUTION: The frequency of CLK_HVIOSC is typ. 16 MHz, min 8 MHz.		

Wait specified time = 10us + Digital noise filter width.

For example, EVCCFLTW[1:0] = 00b EVCCCLKSEL[1:0] = 00b

Wait specified time = 10us + 20cycle * 256 * 62.5ns = 330us

12.2.7.15 VMONKCPROT — VMON Register Key Code Protection Register

The VMONKCPROT register is used for protection against writing operation to the registers that may have a material effect on the system so that the application system is not incorrectly stopped due to program malfunctions and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 12.19 VMONKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing to protected registers.
Write A5A5A501_H to this register to enable writing to protected registers.

12.2.8 Operation

12.2.8.1 VMON Basic Function

VMON watches VDD, VCC and EVCC voltage.

VMON has a high and low voltage detection circuit. If it detects a voltage error, it records the error to the VMONF register and notifies the error to the outside via the $\overline{\text{ERROROUT_M}}$ pin. It also generates internal reset (VMON reset) triggered by detection of voltage error.

(1) VMON detection flag

VMONF.VDDHF is set to “1” when VDD becomes higher than high voltage detection level of VMON.

VMONF.VCCHF is set to “1” when VCC becomes higher than high voltage detection level of VMON.

VMONF.EVCCHF is set to “1” when EVCC becomes higher than high voltage detection level of VMON.

Similarly, VMONF.VDDLf is set to “1” when VDD becomes lower than low voltage detection level of VMON.

VMONF.VCCLf is set to “1” when VCC becomes lower than low voltage detection level of VMON.

VMONF.EVCCLf is set to “1” when EVCC becomes lower than low voltage detection level of VMON.

These flags can be cleared to “0” by writing in the corresponding bit of VMONFC.

These flags are also cleared by Standby Reset (not cleared by other resets). **Figure 12.7** shows an operation example of VMONF.

(2) $\overline{\text{ERROROUT_M}}$ pin

While monitor voltage is higher than the high voltage detection level or lower than the low voltage detection level of VMON, the $\overline{\text{ERROROUT_M}}$ pin outputs a low level.

If VMON does not detect voltage error and ECM does not detect error, $\overline{\text{ERROROUT_M}}$ pin outputs a high level.

To avoid unwanted operation of the VMON circuit caused by instable power supply at power up, $\overline{\text{ERROROUT_M}}$ is fixed to low during the External Reset period and Standby Reset period.

$\overline{\text{ERROROUT_M}}$ outputs the error detection result during other reset periods.

The status of $\overline{\text{ERROROUT_M}}$ is readable from ECM. For details, refer to **Section 39, Error Control Module (ECM)**.

Figure 12.7 shows an operation example of $\overline{\text{ERROROUT_M}}$.

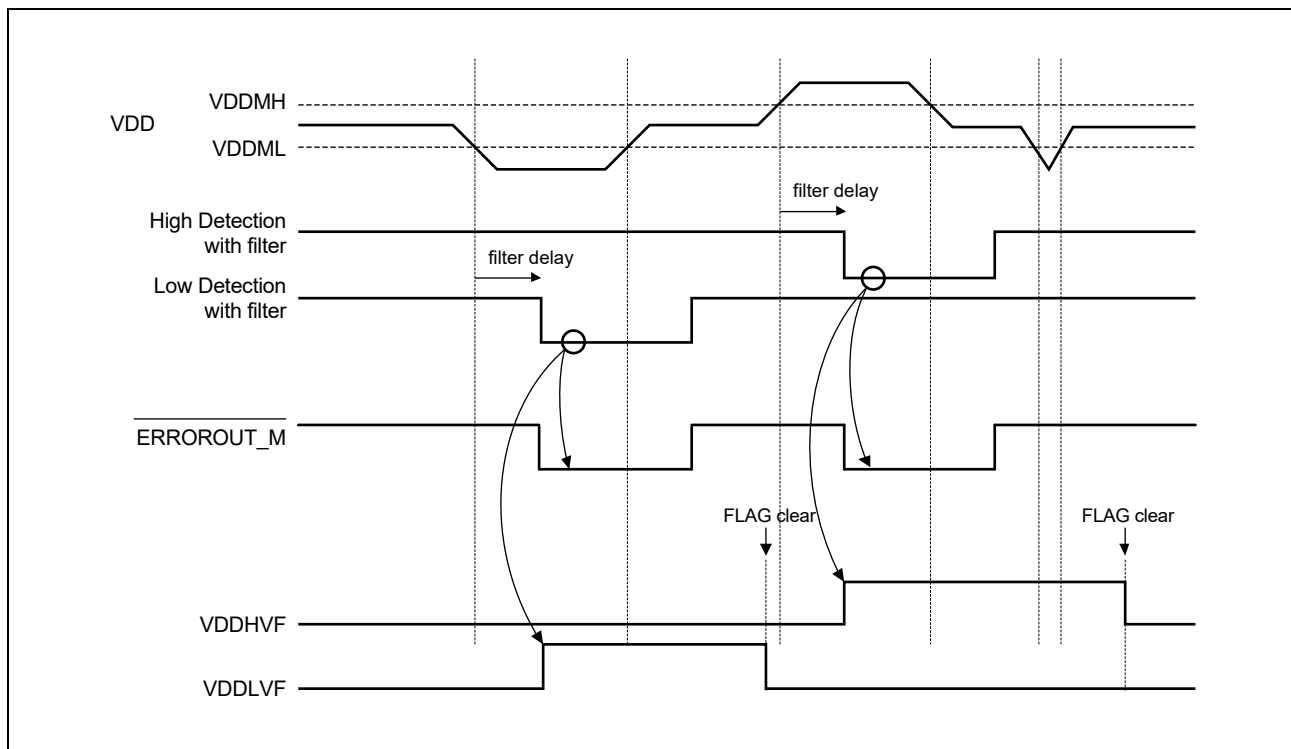


Figure 12.7 Example of VMONF and $\overline{\text{ERROROUT_M}}$ Operation (VDD)

(3) VMON reset

If VDD voltage becomes higher than high voltage detection level of VMON when “101_H” or “111_H” were set to VDDDE.VDDCIRREN[2:0] in advance, VMON reset occurs.

If VDD voltage becomes lower than low voltage detection level of VMON when “110_H” or “111_H” were set to VDDDE.VDDCIRREN[2:0] in advance, VMON reset occurs.

If VCC voltage or EVCC voltage becomes higher than high voltage detection level of VMON when “11_H” was set to VCCDE.VCCCIRREN1, 0*1 or EVCCDE.EVCCIRREN1, 0*1 in advance respectively, VMON reset occurs.

VMON reset is released after the specified time has been passed since VMON detects the normal voltage. **Figure 12.8** shows an operation example of VMON reset.

Note 1. LDET of VCC / EVCC will always generate the VMON RESET at the time of low voltage detection, except during diagnosis.

NOTE

Notes on before enabling VMON after reset release. Before the self-diagnosis is completed, VMON operation is out of guarantee, but the actual operation at normal case is as follows. According to the initial value of the register, when LDET of VCC and EVCC is detected, VMON reset occurs, and when HDET of VCC and EVCC is detected, VMON reset doesn't occur.

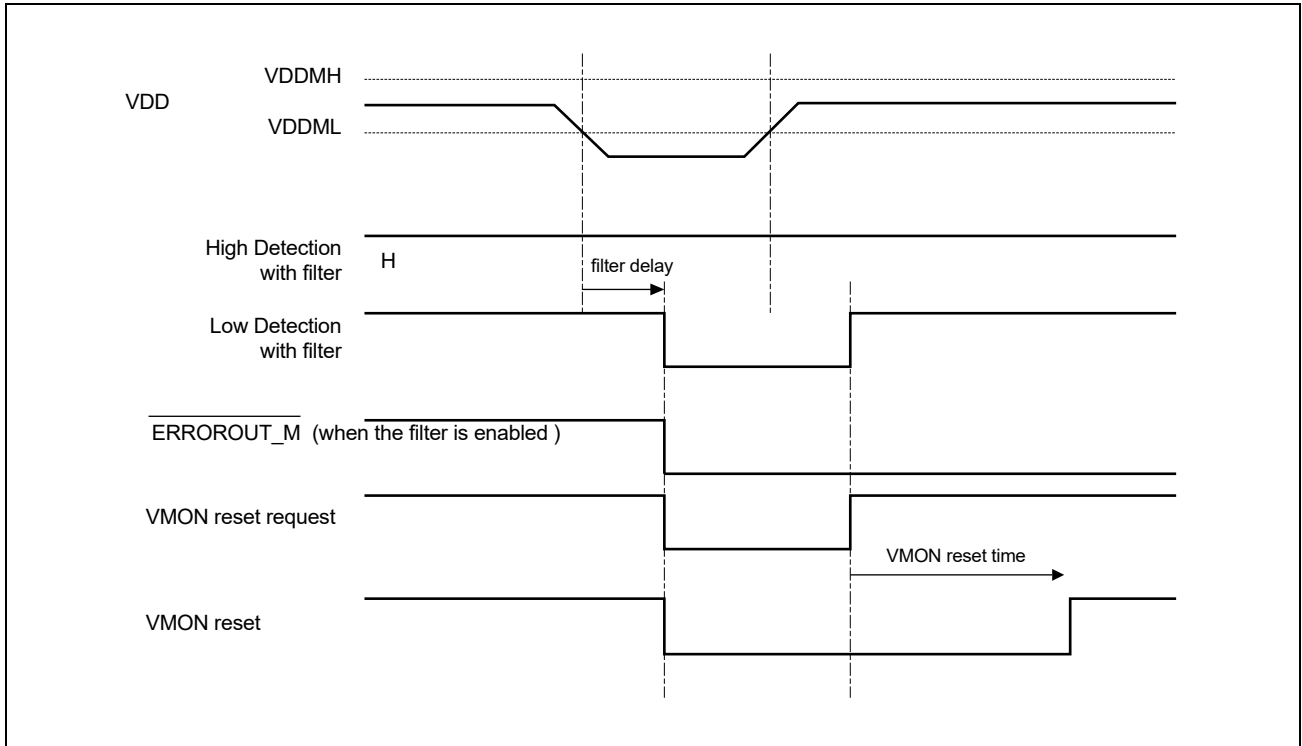


Figure 12.8 Example of VMON Reset Operation (VDD)

(4) Digital Noise Filter

The output signal of the voltage detector passes through a digital noise filter to remove unintended glitches. See *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware* for further information on the filter characteristics.

The digital noise filter clock and the cycle are controlled by setting of the following registers. (These register are read only and set by flash option byte.)

- VDDFCR
- VCCFCR
- EVCCFCR

The relation of filter enable is shown in **Figure 12.4**, **Figure 12.5** and **Figure 12.6**.

12.2.8.2 VMON Function in BIST (Mask of VMON_VDD_ERROROUT and VMON_VDDREQRESET)

It is possible to mask VMON_ERROROUT output and VMON reset by VDDDE.VDDFBISTME when BIST is carried out after VMON Reset.

In this case, VMON_VDD_ERROROUT is fixed at a high level, and VMON_VDD_REQRESET does not occur.

When VDDDE.VDDFBISTME is set to "1", VMON_VDD_ERROROUT and VMON_VDD_REQRESET are not masked when BIST is carried out after VMON Reset.

NOTE

VMON_VDD_ERROROUT output and VMON_VDD_REQRESET are always masked when BIST is carried out after Standby Reset or External Reset.

VMON_VDD_ERROROUT output and VMON_VDD_REQRESET are never masked when BIST is carried out after System Reset 2.

12.2.8.3 VMON Diagnosis Function

(1) Change of VMON detection level for the error injection

When VMONDIAGME.VMONDIAGME is “0”, VDD detection level can be changed by VMONDIAG.VDDDIAGH or VMONDIAG.VDDDIAGL.

When VMONDIAGME.VMONDIAGME is “0”, VCC detection level can be changed by VMONDIAG.VCCDIAGH or VMONDIAG.VCCDIAGL.

When VMONDIAGME.VMONDIAGME is “0”, EVCC detection level can be changed by VMONDIAG.EVCCDIAGH or VMONDIAG.EVCCDIAGL.

VMON error detection signal can be intentionally generated by setting these registers even when each power supply voltage is in the operating range.

NOTE

During self-diagnosis, VMON detection by abnormal voltage can not be operated. As ERROROUT is injected with error in VMON Diagnosis Function, it can not detect VMON by abnormal voltage and ERROROUT output high level.

(2) Mask of VMON_ERROROUT and VMON reset

When VMONDIAGME.VMONDIAGME is “0” and VMONDMASK.VDDERRMD is set to “1”, VMON_VDD_ERROROUT can be masked.

When VMONDIAGME.VMONDIAGME is “0” and VMONDMASK.VCCERRMD is set to “1”, VMON_VCC_ERROROUT can be masked.

When VMONDIAGME.VMONDIAGME is “0” and VMONDMASK.EVCCERRMD is set to “1”, VMON_EVCC_ERROROUT can be masked.

If above three settings are done, $\overline{\text{ERROROUT_M}}$ is dependent on the setting of ECM and DMON during diagnosis.

When VMONDIAGME.VMONDIAGME is “0” and VMONDMASK.VDDRESMD is set to “1”, VMON_VDD_REQ_RESET can be masked.

When VMONDIAGME.VMONDIAGME is “0” and VMONDMASK.VCCRESMD is set to “1”, VMON_VCC_REQ_RESET can be masked.

When VMONDIAGME.VMONDIAGME is “0” and VMONDMASK.EVCCRESMD is set to “1”, VMON_EVCC_REQ_RESET can be masked.

(3) Flow of VMON diagnosis application

VMONDIAGME.VMONDIAGME is initialized to “0” by Standby Reset or External Reset. Therefore, VMON diagnosis function is enabled at start-up after these resets.

The user’s application controls the diagnostic function by VMONDMASK, VMONDIAG and VMONDIAGFE.

Figure 12.9 shows an example of the diagnosis application.

After diagnosis application is finished, write “1” to VMONDIAGME.VMONDIAGME and disable VMON diagnosis function immediately.

This function protects each flag against erroneous operation when each power supply is out of the operating range.

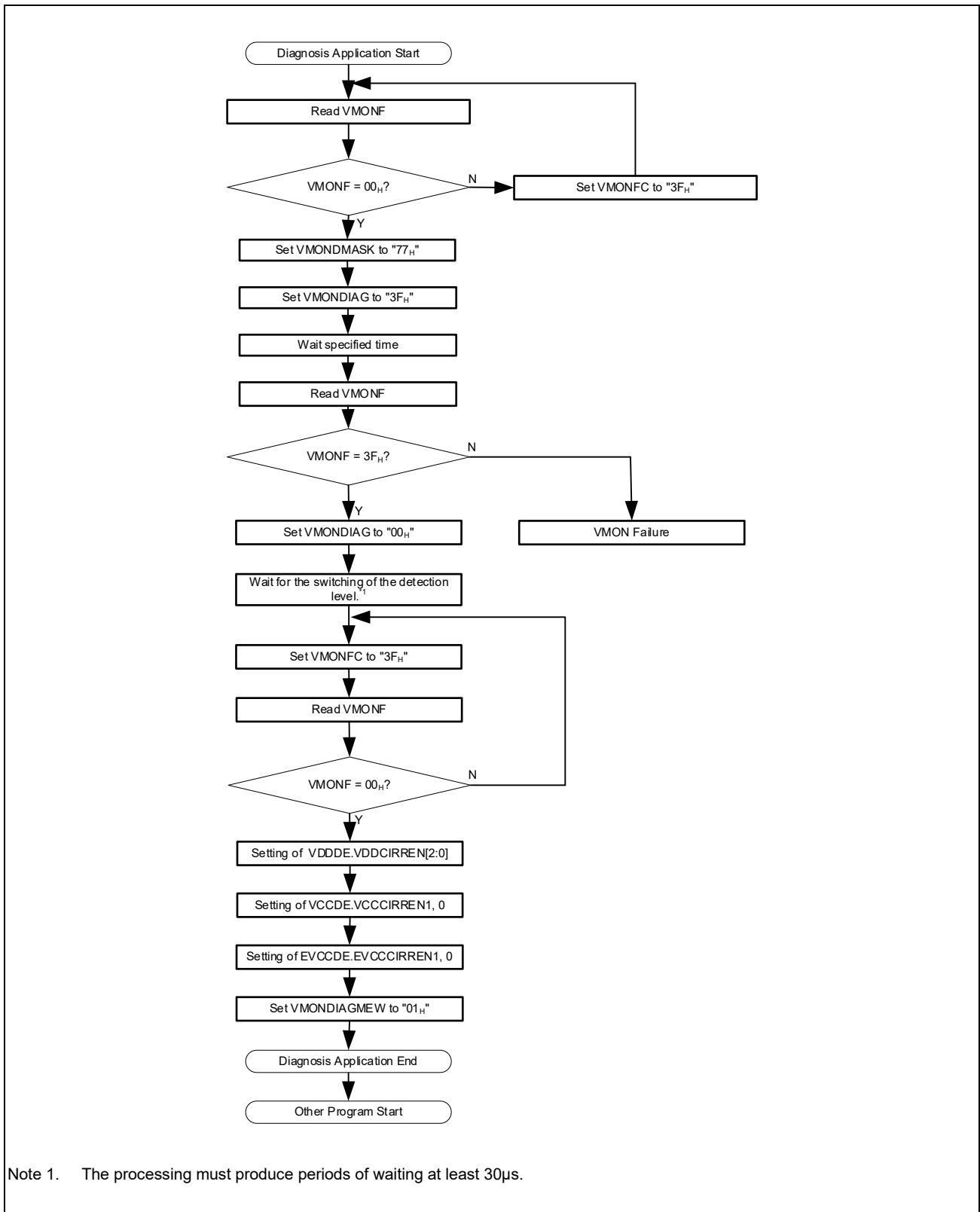


Figure 12.9 Example of the Flowchart of VMON Diagnosis Application

Wait specified time = 10us + Digital noise filter width.

Please refer VDDFCR and VCCFCR and EVCCFCR for “Wait specified time”

12.2.9 Usage Notes

12.2.9.1 Flag Bit (VMONF)

Setting 1 to VMONF by error detection is prior to clearing VMONF by VMONFC.

These functions protect each flag against erroneous operation when each power supply is out of the operating range.

When Monitor voltages are recovered to the operating range, the MCU needs to be initialized by External Reset or VMON reset, or the MCU may still continue erroneous operation and then unintended flag clearing may occur.

12.2.9.2 VCC Primary Low Detection

Because of internal circuit guard of this LSI, VMON Reset might occur in a shorter time than the filter width when VCC voltage is lower than the lowest level of VCC primary low detection range*1.

In that case, VCCLVF will not be set by the VCC primary low detection.

Note 1. VCC primary low detection level has some width. For details, refer to *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

12.2.9.3 External Reset occurs during self-diagnosis

If the External Reset conflicts with the VCC and EVCC low voltage detection of self-diagnosis, Standby Reset will occur.

12.3 Delay Monitor (DMON)

12.3.1 Features

DMON can detect abnormalities in the delay time of transistor devices by using an on-chip delay monitor while the microcontroller is operating. DMON can only monitor the delay time on the VDD domain. It detects an error if the monitored delay time is lower than the threshold according to the fabrication process conditions.

DMON assists the power supply voltage monitor (VMON), which detects the low level voltage of VDD.

When DMON detects a delay error within the ranges of VMON detection, it notifies VMON about the error detection to operate VMON reset normally.

12.3.2 Register Base Addresses

The Delay Monitor base address is given in the following table.

Delay Monitor register addresses are given as offsets from the base addresses in general.

Table 12.20 Register Base Addresses

Base Address Name	Base Address	Bus Group
<DMON_base>	FF70 1C00 _H	Peripheral Group 6

12.3.3 Clock Supply

The clock supply to DMON is shown in the following table.

Table 12.21 Clock Supply

Unit Name	Clock Name	Symbol
DMON	Main OSC	CLK_MOSC
	Peripheral low speed clock	CLK_LSB

12.3.4 Interrupt and DMA/DTS Requests

This module has no interrupt and DMA/DTS requests.

12.3.5 External Input and Output Pins

This module has no external input/output pins.

12.3.6 Overview

12.3.6.1 Functional Overview

The On-Chip delay monitor measures the delay time related to the power supply voltage. It converts analog delay time into digital codes (DCODE) and outputs the DCODE into a comparator. The DCODE is compared with the threshold, which is a fixed value. When the DCODE becomes lower than the threshold, an error output signal (DMON_ERROROUT) and a reset request signal (DMON_REQRESET) are generated to notify the VMON.

12.3.6.2 Block Diagram

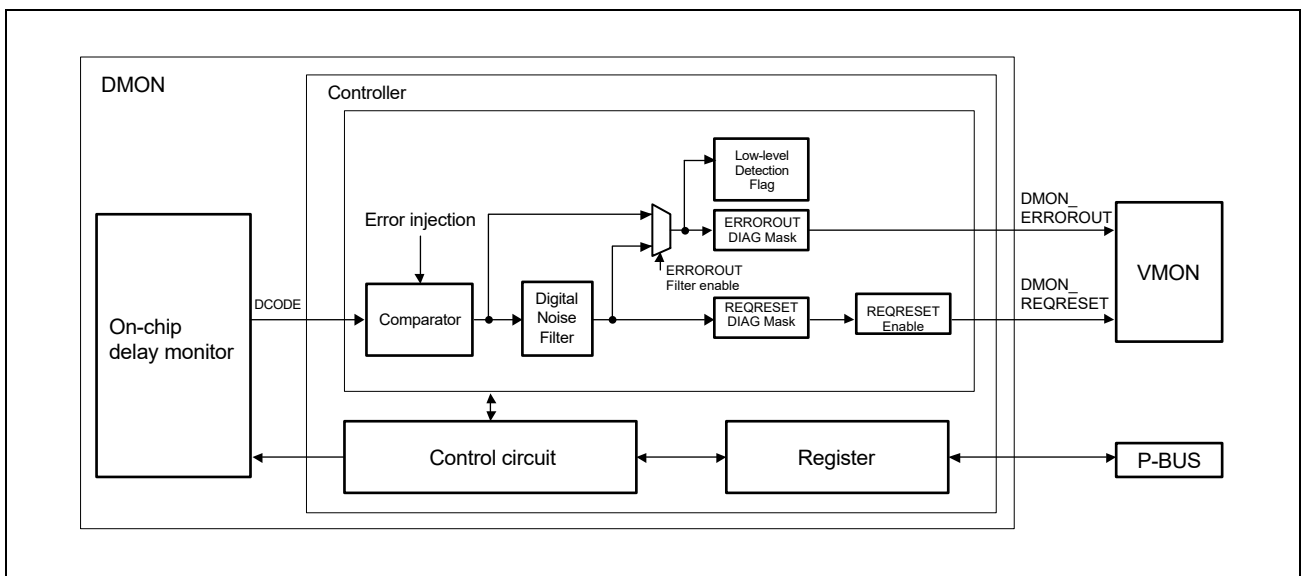


Figure 12.10 Block Diagram of Delay Monitor

12.3.7 Registers

12.3.7.1 List of Registers

The register list related to DMON is shown in **Table 12.22**.

Table 12.22 List of Registers (DMON)

Unit Name	Register Name	Symbol	Address	Access Size	Access Protection
DMON	DMON Factor Register	DMONF	<DMON_base> + 000 _H	32	
DMON	DMONF Clear Register	DMONFC	<DMON_base> + 004 _H	32	DMONKCPR0
DMON	DMON DIAG Monitor Enable Register	DMONDIAGME	<DMON_base> + 008 _H	32	
DMON	DMONDIAGME Write Register* ¹	DMONDIAGMEW	<DMON_base> + 00C _H	32	DMONKCPR0
DMON	DMON Detection Output Diagnosis Mask Register	DMONDMASK	<DMON_base> + 010 _H	32	
DMON	DMON DIAG Mode Setting Register	DMONDIAG	<DMON_base> + 014 _H	32	
DMON	DMON Detection Enable Register	DMONDE	<DMON_base> + 018 _H	32	DMONKCPR0
DMON	DMON Filter Control Register	DMONFCR	<DMON_base> + 01C _H	32	
DMON	DMON Register Key Code Protection Register 0	DMONKCPR0	<DMON_base> + 300 _H	32	

Note 1. Writing is permitted only once after System Reset 1/2.

Register reset conditions are shown in **Table 12.23**.

Table 12.23 Register Reset Conditions for DMON

Register Name	Reset Source					
	Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
DMONF	✓	✓* ¹	—	—	—	—
DMONFC	✓	✓	—	—	—	—
DMONDIAGME	✓	✓	✓	—	—	—
DMONDIAGMEW	✓	✓	✓	—	—	—
DMONDIAG	✓	✓	✓	—	—	—
DMONDMASK	✓	✓	✓	—	—	—
DMONDE	✓	✓	✓	—	—	—
DMONFCR	✓	✓	✓	—	—	—
DMONKCPR0	✓	✓	✓	✓	—	—

Note 1. Standby Reset only.

12.3.7.2 DMONF — DMON Factor Register

The DMONF register indicates that an error is detected by DMON. Each flag of DMONF can be cleared individually by writing “1” to the corresponding bit in DMONFC.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMON LVF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.24 DMONF Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned.
0	DMONLVF	Lower limit delay detection flag 0: No lower limit delay error detected 1: Lower limit delay error occurred

12.3.7.3 DMONFC — DMONF Clear Register

DMONFC is a register to clear the DMONF register. The read value of this register is always 0000 0000_H.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMONLVFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 12.25 DMONFC Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned.
0	DMONLVFC	Clear lower limit delay detection flag 0: Writing 0 has no effect 1: Writing 1 clears DMONLVF

12.3.7.4 DMONDIAGME — DMON DIAG Monitor Enable Register

DMONDIAGME is a read only register to control the DIAG function.

This register reflects the settings written to DMONDIAGMEW.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMON DIAGM E
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.26 DMONDIAGME Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned.
0	DMONDIAGME	Permit DIAG function 0: DIAG function of the DMON can be enabled 1: DIAG function of the DMON cannot be enabled

12.3.7.5 DMONDIAGMEW — DMONDIAGME Write Register

DMONDIAGMEW is a register to set values of DMONDIAGME. Writing is permitted only once after System Reset 1/2 is released. Subsequent write operations are ignored. The read value of this register is always 0000 0000_H.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMON DIAGM EW
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 12.27 DMONDIAGMEW Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When writing, write the value after reset.
0	DMONDIAGMEW	The data written in this bit is set to DMONDIAGME.DMONDIAGME.

12.3.7.6 DMONDIAG — DMON DIAG Mode Setting Register

The DMONDIAG register is used for controlling the diagnostic function of DMON.

This register is valid only when DMONDIAGME.DMONDIAGME = 0. The setting of DMONDIAG is ignored if DMONDIAGME.DMONDIAGME = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMON DIAGL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.28 DMONDIAG Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	DMONDIAGL	This bit is used for error injection of diagnosis function when DMONDIAGME = 0. 0: Normal operation. 1: Execute error injection.

12.3.7.7 DMONDMASK — DMON Detection Output Diagnosis Mask Register

DMONDMASK is a register to mask DMON_REQRESET and DMON_ERROROUT when

DMONDIAGME.DMONDIAGME = 0. DMONDMASK setting is ignored if DMONDIAGME.DMONDIAGME = 1.

Value after reset: 0000 0011_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	DMON RESMD	—	—	—	DMON ERRMD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Table 12.29 DMONDMASK Register Contents

Bit Position	Bit Name	Function
31 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	DMONRESMD	This bit masks DMON_REQRESET when DMONDIAGME.DMONDIAGME = 0. 0: DMON_REQRESET is not masked. 1: DMON_REQRESET is masked.
3 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	DMONERRMD	This bit masks DMON_ERROROUT when DMONDIAGME.DMONDIAGME = 0. 0: DMON_ERROROUT is not masked. 1: DMON_ERROROUT is masked.

12.3.7.8 DMONDE — DMON Detection Enable Register

DMONDE is a register to control DMON detection, the DMON signal mask and DMON signal filter.

Writing in DMONDE is permitted during the period before writing in DMONDIAGMEW after System Reset 1/2 is released. Writing in this register is ignored after writing in DMONDIAGMEW.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DMON CIRRE N1	—	DMON CIRRE N0	—	DMONF LTEN	—	—	DMON LDE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W

Table 12.30 DMONDE Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	DMONCIRREN1	Permit DMON_REQRESET notification. DMONCIRREN1, 0 00 _B , 01 _B , 10 _B : DMON_REQRESET is not permitted 11 _B : DMON_REQRESET is permitted
6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	DMONCIRREN0	Refer to bit 7 (DMONCIRREN1).
4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	DMONFLTEN	Enable output filter for DMON_ERROROUT and DMON Error Detection Flag 0: Disable output filter for DMON_ERROROUT and DMON Error Detection Flag 1: Enable output filter for DMON_ERROROUT and DMON Error Detection Flag
2 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	DMONLDE	DMON Low delay detection enable 0: Disable DMON lower limit delay detection 1: Enable DMON lower limit delay detection

12.3.7.9 DMONFCR — DMON Filter Control Register

DMONFCR is a register to adjust the filtering width of the DMON digital noise filter.

Writing in DMONFCR is permitted during the period before writing in DMONDIAGMEW after System Reset 1/2 is released. Writing in this register is ignored after writing in DMONDIAGMEW.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DMONFLTW		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12.31 DMONFCR Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	DMONFLTW	These bits select the minimum filtering width of the digital noise filter. The minimum filtering width is given as a number of sampling times. 000: 1 sampling times 001: 2 sampling times 010: 4 sampling times 011: 8 sampling times 100: 16 sampling times 101: 32 sampling times 110: 64 sampling times 111: 128 sampling times

CAUTION: The sampling period is 4 μ s.

12.3.7.10 DMONKCPROT0 — DMON Register Key Code Protection Register 0

The DMONKCPROT0 register is used for protection against writing operation to the registers that may have a material effect on the system so that the application system is not incorrectly stopped due to program malfunction and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 12.32 DMONKCPROT0 Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing to protected registers.
Write A5A5A501_H to this register to enable writing to protected registers.

12.3.8 Operation

12.3.8.1 DMON Basic Function

DMON monitors the delay time depending on the VDD supply voltage.

If it detects a delay error, it records the error to DMONF register. It also generates DMON_ERROROUT and DMON_REQRESET triggered by detection of the lower limit delay, and notifies the error to VMON.

Figure 12.11 shows the flowchart of starting up DMON. DMON begins to operate after VMON starts up and the Clock Monitor confirms that Main OSC clock is normally operating. The user must invoke DMON after starting up VMON and checking Clock Monitor to see whether or not Main OSC clock is operating normally. After DMON starts up, the system clock needs to be changed to PLL and the clock gear up is carried out.

The timing chart of DMON operation is shown in **Figure 12.12**.

When Power Up Reset or System Reset 1/2 is asserted, DMON is also initialized by these resets. After restart from the reset, the start-up sequence needs to be executed again.

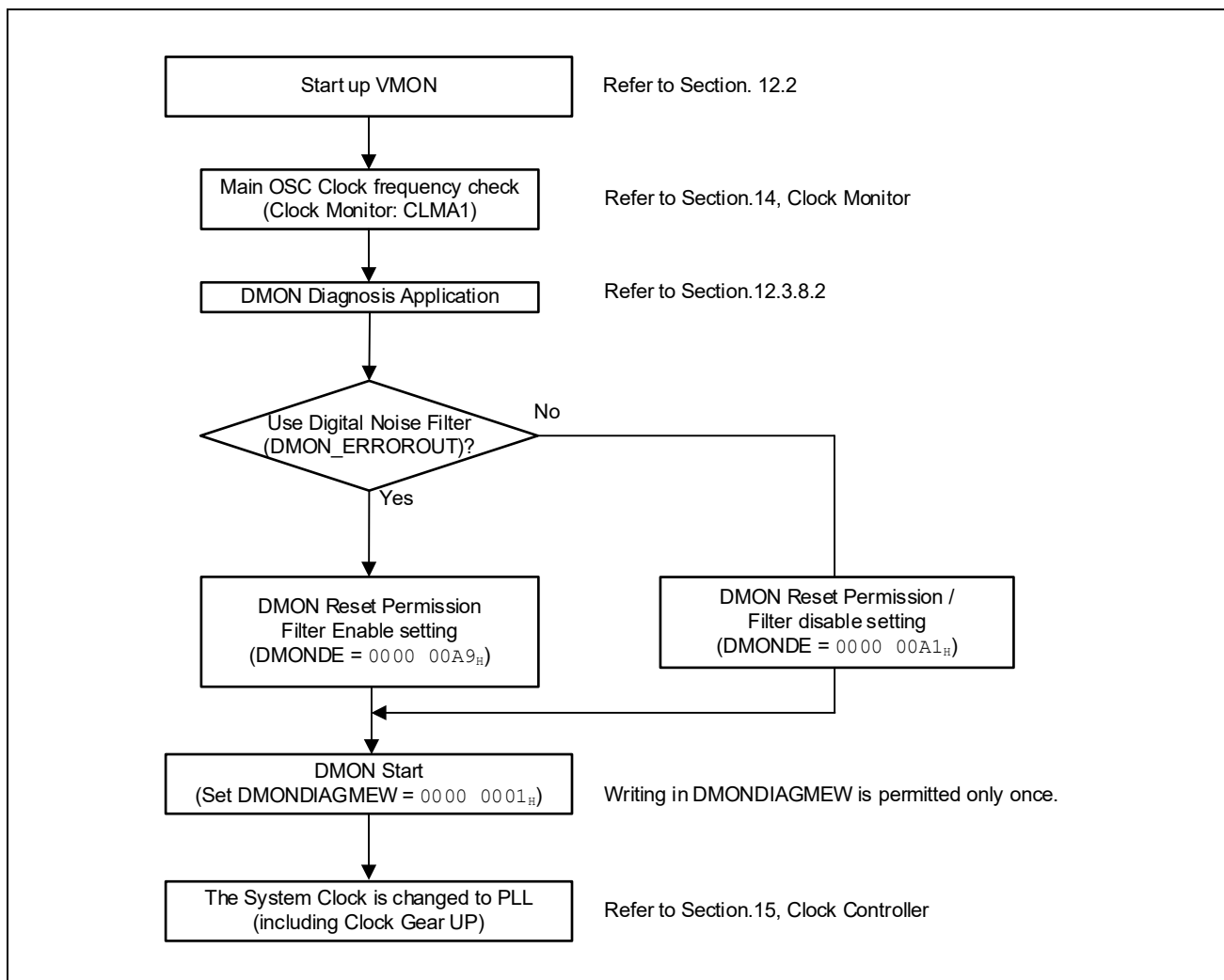


Figure 12.11 Flowchart of DMON Start-Up

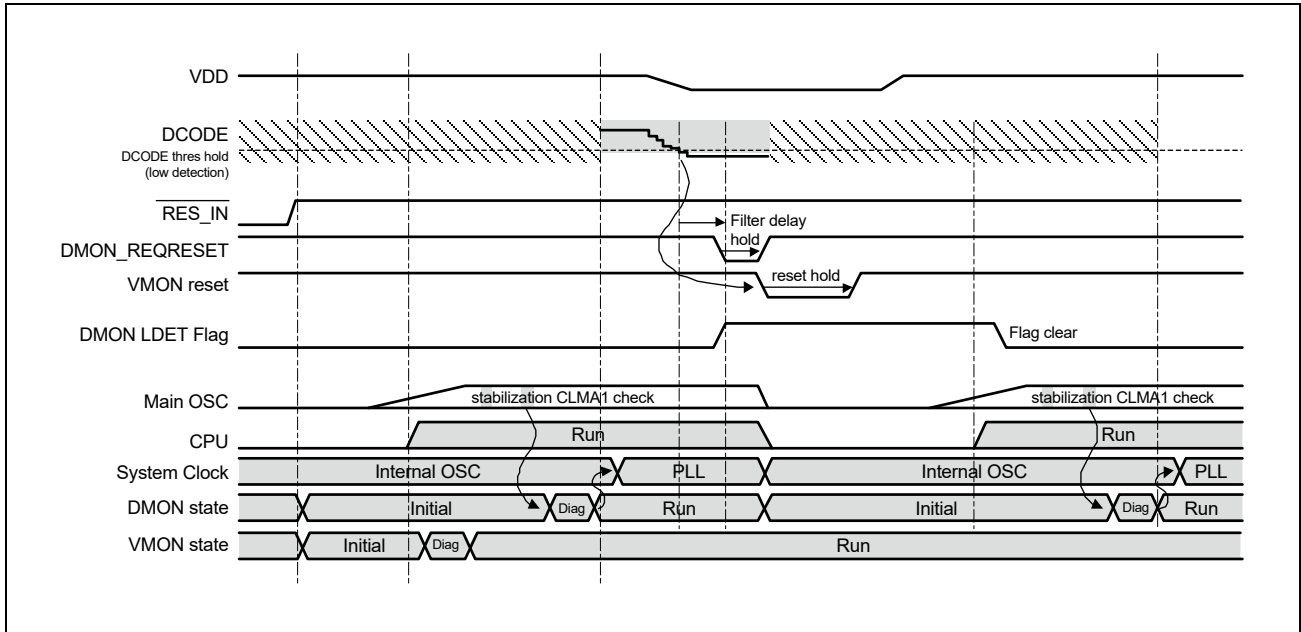


Figure 12.12 Timing Chart of the DMON Function

DMON generates the sampling period from Main OSC clock (CLK_MOSC). If the frequency of Main OSC is faster than the spec, the DCODE will become smaller than the expected value due to shortening of the sampling period. Therefore, there is a possibility that DMON detects the delay error before the error detection of clock monitor for Main OSC and the MCU is initialized by VMON reset. If the user distinguishes between the failure of Main OSC and the reset factor caused by DMON error after the MCU restarts from VMON reset, the user needs to confirm the DMON error flag and check the frequency of Main OSC by clock monitor (CLMA1). **Figure 12.13** shows an example of a flowchart to identify the reset factor.

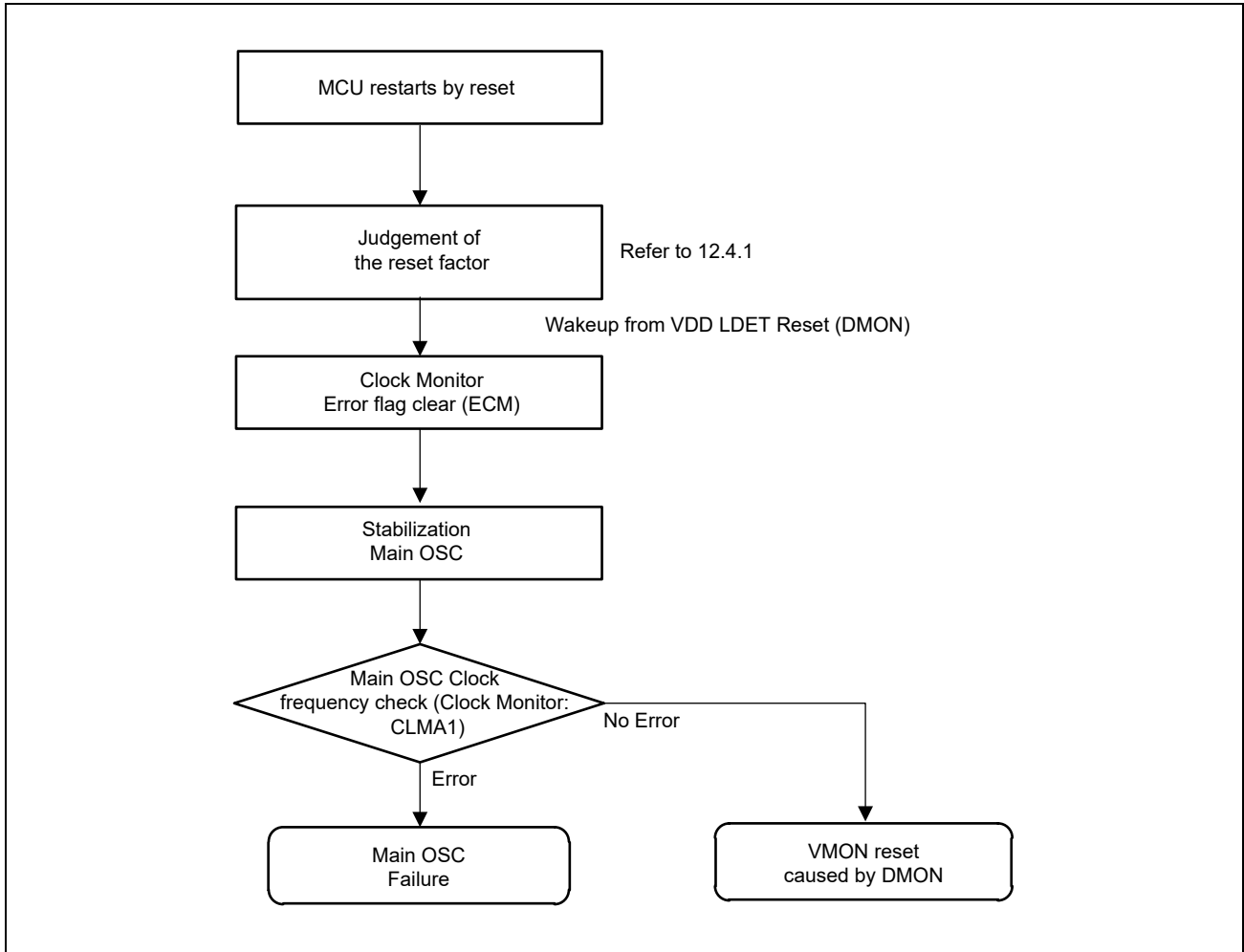


Figure 12.13 Example of Flowchart to Identify the Failure of Main OSC

(1) DMON Error Detection Flag

DMON sets “1” to DMONF.DMONLVF when the DCODE becomes lower than the low detection threshold level.

The flag can be cleared to “0” by writing to the corresponding bit of DMONFC.

The flag is also cleared by power up reset and standby reset (not cleared by other resets). **Figure 12.14** shows an operation example of DMONF.

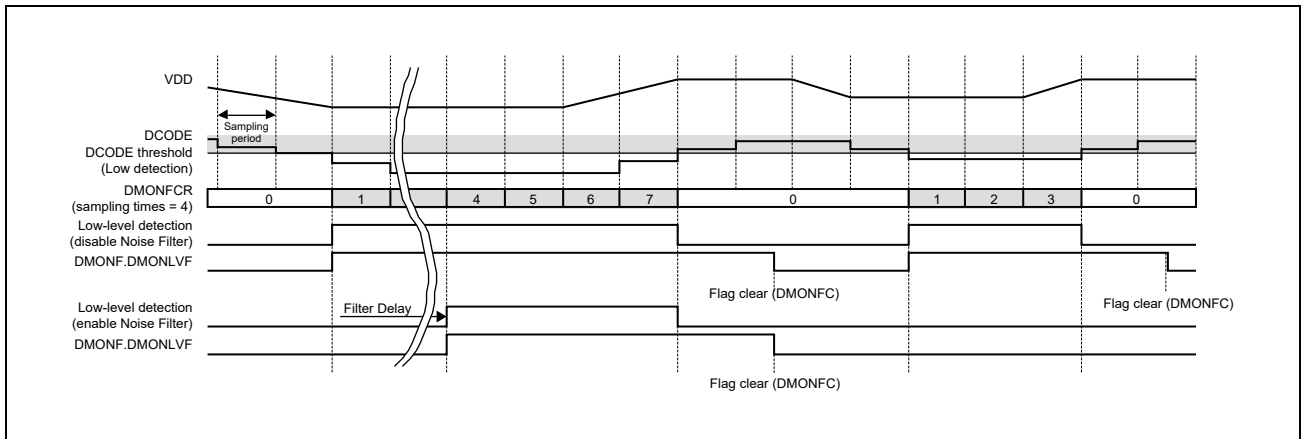


Figure 12.14 DMON Error Detection Flag

(2) DMON_REQRESET Output

If the DCODE value becomes lower than the DCODE threshold level when DMONDE.DMONCIRREN is set to “11_B”, DMON notifies VMON that the DMON_REQRESET has occurred. After that, VMON reset occurs and the MCU including DMON is initialized. DMON continues to hold the DMON_REQRESET state until DMON is initialized by VMON reset. **Figure 12.15** shows an operation example of DMON_REQRESET.

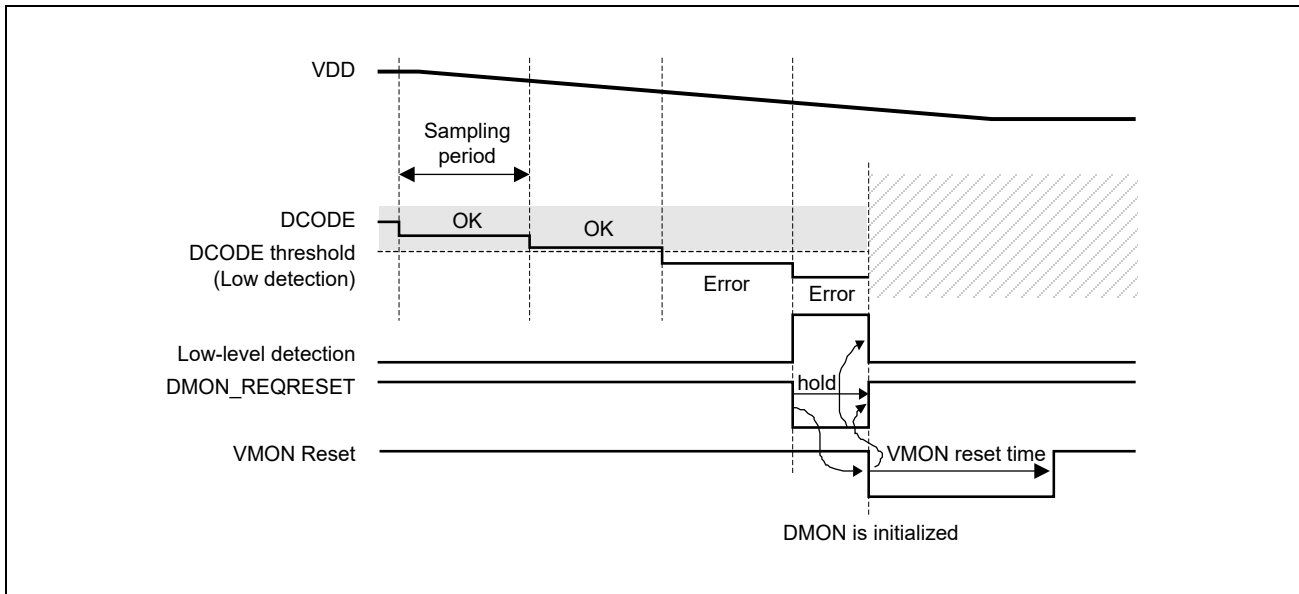


Figure 12.15 DMON_REQRESET Operation

(3) Digital Noise Filter

The output signal of the DMON comparators passes through a digital noise filter to remove unintended glitches.

Whether to use filtered output or to bypass the filter is selectable by DMONDE.DMONFLTEN.

The digital noise filter period can be changed by the DMONFCR register.

12.3.8.2 DMON Diagnosis Function

(1) Change of DMON detection level for error injection

When DMONDIAGME.DMONDIAGME is set to “0”, the Lower limit delay threshold level can be changed by DMONDIAG.DMONDIAGL.

The DMON error detection signal can be intentionally generated by setting this register when VMON did not detect a voltage error and the clock monitor CLMA1 did not detect frequency abnormalities.

(2) Mask of DMON_ERROROUT and DMON_REQRESET

When DMONDIAGME.DMONDIAGME is set to “0” and DMONDMASK.DMONERRMD is set to “1”, DMON_ERROROUT can be masked.

When DMONDIAGME.DMONDIAGME is set to “0” and DMONDMASK.DMONRESMD is set to “1”, DMON_REQRESET can be masked.

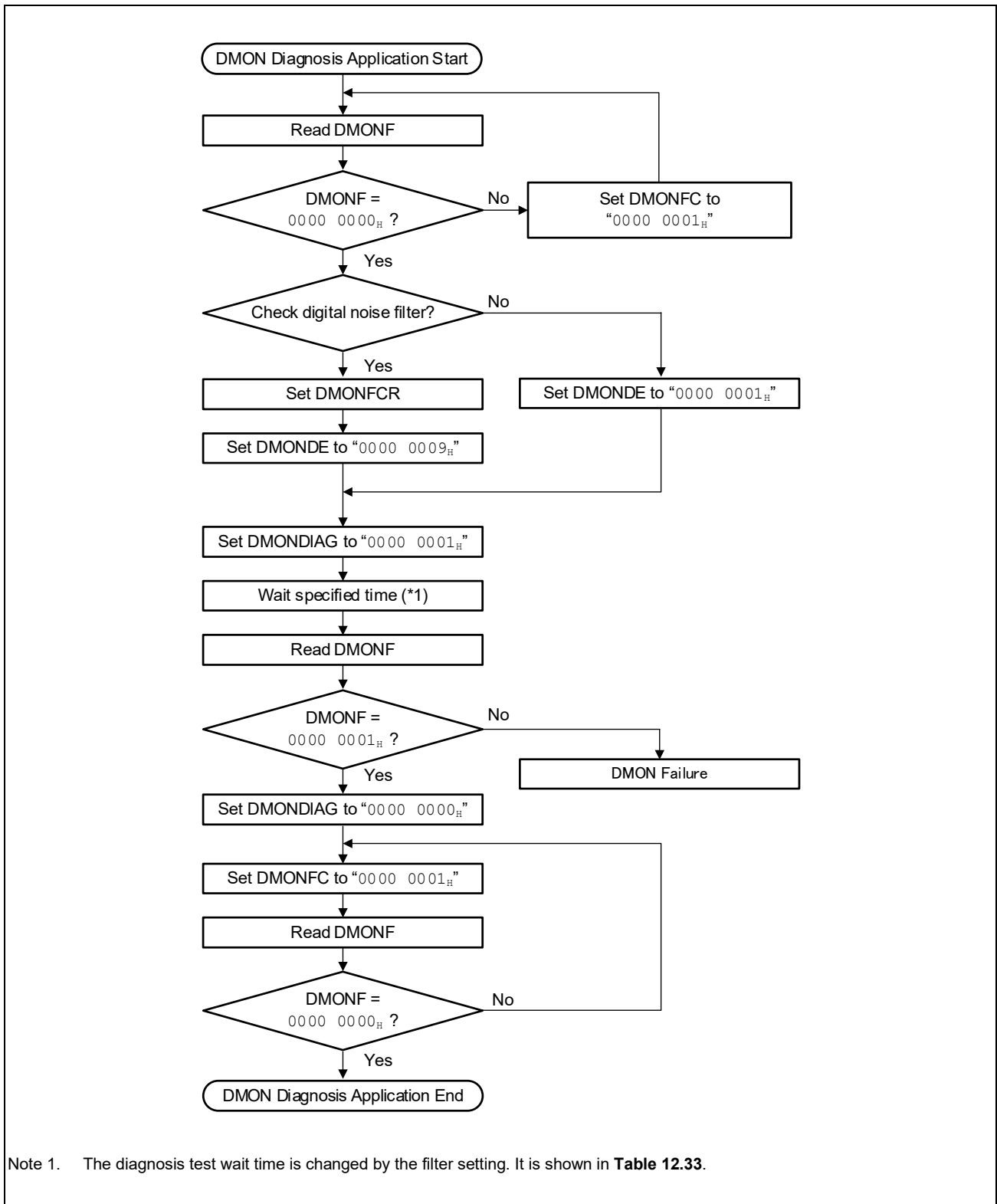
(3) Flow of DMON diagnosis application

DMONDIAGME.DMONDIAGME is initialized to “0” by Power Up Reset or System Reset 1/2. Therefore, the DMON diagnosis function is enabled at start-up after these resets.

The user’s application controls the diagnostic function by DMONDMASK and DMONDIAG. **Figure 12.16** shows the example of the diagnosis application.

After diagnosis application is finished, write “1” to DMONDIAGMEW.DMONDIAGMEW and disable the DMON diagnosis function immediately.

This function protects each flag against erroneous operation when each power supply is out of the operating range.



Note 1. The diagnosis test wait time is changed by the filter setting. It is shown in **Table 12.33**.

Figure 12.16 Example of Flowchart of DMON Diagnosis Application

Table 12.33 Wait Times of Diagnosis Test

DMONFCR.DMONFLTW	Wait time
000	12 us
001	16 us
010	24 us
011	40 us
100	72 us
101	136 us
110	264 us
111	520 us

12.4 Usage Notes

12.4.1 Judgment Method of Reset Factors for Voltage Monitor

If RESF.SRES1F2 bit is “1”, this product is woken up

Check the VMONF and DMONF for the detection power.

The judgment method of the reset factors is shown in **Figure 12.17**.

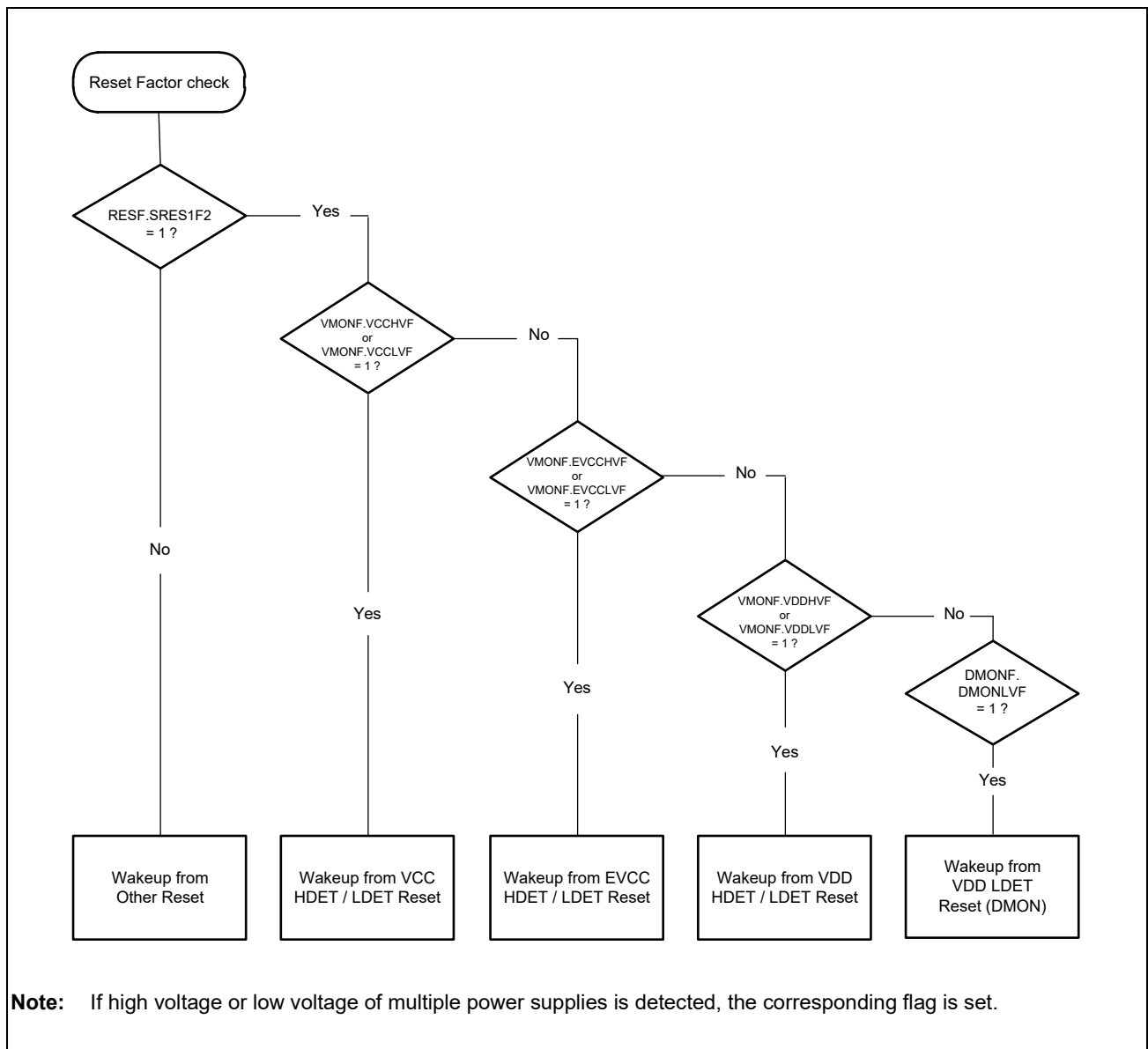


Figure 12.17 Flowchart of Reset Factor Judgment

Section 13 Temperature Sensor

13.1 Features

13.1.1 Number of Channels

Table 13.1 Number of Channels

Temperature Sensor	
Number of Channels	1
Name	OTS0

13.1.2 Register Base Address

Table 13.2 Register Base Addresses

Base Address Name	Base Address	Bus Group
<OTS0_base>	FF70 A000 _H	Peripheral Group 6

13.1.3 Clock Supply

The clocks supplied to the temperature sensor are listed in the following table.

Table 13.3 Clock Supply

Unit Name	Unit Clock Name	Clock Name
OTS0	Operation clock	CLK_LSB
	Register access clock	CLK_LSB

13.1.4 Interrupt Requests

The temperature sensor interrupt requests are listed in the following table.

Table 13.4 Interrupt Requests

Unit Interrupt Name	Unit Interrupt Signal Outline	Interrupt Number
INTOTSOTE	Temperature sensor error interrupt	501
INTOTSOTI	Temperature measurement end interrupt	502
INTOTSOTULI	Triggered if the state machine changes the state due to a temperature rise or drop in the guaranteed temperature range	503*1
INTOTSOTABE	Abnormal temperature error interrupt	503*1

Note 1. INTOTSOTULI and INTOTSOTABE are merged.

13.1.5 External Input and Output Pins

This module has no external input/output pin.

13.2 Overview

13.2.1 Functional Overview

The following describes the features of the temperature sensor.

- Temperature data register

A temperature data register stores a temperature measurement value.

- Temperature measurement mode

Single measurement mode: Used to measure temperature only once.

Continuous measurement mode: Used to measure temperature continuously.

- Supporting temperature measurement end interrupt

Each time temperature measurement ends, the temperature sensor can generate an interrupt request (INTOTSOTI) to be sent to the INTC.

- Supporting temperature alarm error interrupt and temperature rise/drop interrupt

Six temperature threshold values (high-temperature border AU > high-temperature border AL > high-temperature border BU > high-temperature border BL > low-temperature border AU > low-temperature border AL) should be set in advance. Furthermore, the temperature sensor has four temperature states (high temperature A, high temperature B, normal temperature, and low temperature A). The temperature state is updated at each temperature measurement according to six temperature threshold values. The following describes the conditions for generating a temperature alarm error interrupt and a temperature rise/drop interrupt.

- A temperature alarm error interrupt (INTOTSOTABE) is output at a transition from a state other than high temperature A to high temperature A or at a transition from a state other than low temperature A to low temperature A.
- A temperature rise/drop interrupt (INTOTSOTULI) is output at a transition from high temperature A to high temperature B or normal temperature, or at a transition from high temperature B to normal temperature, that is, when the temperature drops.
- A temperature rise/drop interrupt (INTOTSOTULI) is output at a transition from low temperature A to high temperature B or normal temperature, or at a transition from normal temperature to high temperature B, that is, when the temperature rises.

Temperature status can be monitored by reading the temperature status register.

- Diagnosis function

When temperature is measured in continuous measurement mode, if the difference between the measured value and the previously measured value is larger than the specified temperature difference limit value, a temperature sensor error interrupt (INTOTSOTE) is output.

- Reducing errors by temperature correction

Three temperature correction coefficients are stored during device manufacturing in coefficient registers A to C.

13.2.2 Block Diagram

Figure 13.1 shows a block diagram of the temperature sensor.

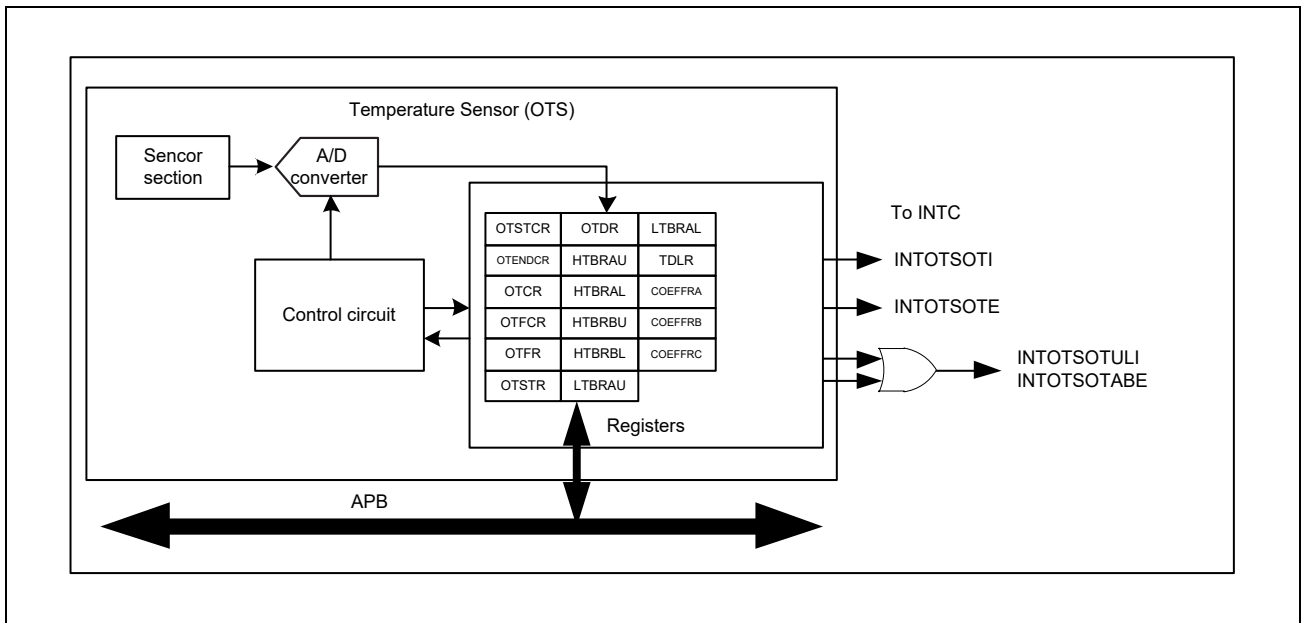


Figure 13.1 Temperature Sensor Block Diagram

13.3 Register

13.3.1 List of registers

The following table lists the registers of the temperature sensor.

<OTS_n_base> is defined in **Section 13.1.2, Register Base Address**.

Table 13.5 List of Registers

Unit Name	Register Name	Symbol	Address	Access Size
OTS _n	Temperature Measurement Start Control Register	OTS _n OTSTCR	<OTS _n _base> + 00 _H	8
OTS _n	Temperature Measurement End Control register	OTS _n OTENDCR	<OTS _n _base> + 04 _H	8
OTS _n	Temperature Sensor Control Register	OTS _n OTOCR	<OTS _n _base> + 08 _H	8
OTS _n	Temperature Sensor Flag Clear Register	OTS _n OTFCR	<OTS _n _base> + 0C _H	8
OTS _n	Temperature Sensor Flag Register	OTS _n OTFR	<OTS _n _base> + 10 _H	8
OTS _n	Temperature Status Register	OTS _n OTSTR	<OTS _n _base> + 14 _H	8
OTS _n	Temperature Data Register	OTS _n OTDR	<OTS _n _base> + 18 _H	16
OTS _n	High-Temperature Border AU Register	OTS _n HTBRAU	<OTS _n _base> + 1C _H	16
OTS _n	High-Temperature Border AL Register	OTS _n HTBRAL	<OTS _n _base> + 20 _H	16
OTS _n	High-Temperature Border BU Register	OTS _n HTBRBU	<OTS _n _base> + 24 _H	16
OTS _n	High-Temperature Border BL Register	OTS _n HTBRBL	<OTS _n _base> + 28 _H	16
OTS _n	Low-Temperature Border AU Register	OTS _n LTBRAU	<OTS _n _base> + 2C _H	16
OTS _n	Low-Temperature Border AL Register	OTS _n LTBRAL	<OTS _n _base> + 30 _H	16
OTS _n	Temperature Difference Limiting Register	OTS _n TDLR	<OTS _n _base> + 34 _H	16
OTS _n	Coefficient A Register	OTS _n COEFFRA	<OTS _n _base> + 38 _H	16
OTS _n	Coefficient B Register	OTS _n COEFFRB	<OTS _n _base> + 3C _H	16
OTS _n	Coefficient C Register	OTS _n COEFFRC	<OTS _n _base> + 40 _H	16

Note: (n = 0)

13.3.2 Reset of Registers

Table 13.6 Register Reset Conditions

Register Name	Reset Source					
	Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
All registers	✓	✓	✓	✓	—	—

13.3.3 OTS0OTSTCR — Temperature Measurement Start Control Register

OTS0OTSTCR is an 8-bit write-only control register to start the temperature sensor.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OTST
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 13.7 OTS0OTSTCR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	OTST	Temperature Measurement Start Condition for starting temperature measurement by OTST. Write 1 to OTST when OACT = 0

13.3.4 OTS0OTENDCR — Temperature Measurement End Control Register

OTS0OTENDCR is an 8-bit write-only control register to terminate the temperature sensor.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OTEND
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 13.8 OTS0OTENDCR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	OTEND	Temperature Measurement End Condition for terminating temperature measurement by OTEND. Write 1 to OTEND and then wait for 2 cycles of the OTS clock.

13.3.5 OTS0OTCR — Temperature Sensor Control Register

OTS0OTCR is an 8-bit readable/writable register to control the temperature sensor.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	OTEE	OTULIE	OTABEE	SDE	OTMD
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 13.9 OTS0OTCR Register Contents

Bit Position	Bit Name	Function
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	OTEE	Temperature Sensor Error Enable 0: Disabled 1: Enabled When OTMD = 0 (single measurement mode) or SDE = 0 (diagnosis disabled), no temperature sensor error is generated regardless of the OTEE setting.
3	OTULIE	Temperature Rise/Drop Interrupt Enable 0: Disabled 1: Enabled
2	OTABEE	Temperature Alarm Error Enable 0: Disabled 1: Enabled
1	SDE	Diagnosis Enable 0: Disabled 1: Enabled When OTMD = 0 (single measurement mode), diagnosis is not performed regardless of the SDE setting.
0	OTMD	Temperature Measurement Mode 0: Single temperature measurement mode 1: Continuous temperature measurement mode

CAUTION

To prevent malfunction, set OTS0OTCR while the OTACT bit in OTS0OTFR is 0.

13.3.6 OTS0OTFCR — Temperature Sensor Flag Clear Register

OTS0OTFCR is an 8-bit write-only register to clear the OTS0OTFR register.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	SDERC	—	OTFC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	W	R	W

Table 13.10 OTS0OTFCR Register Contents

Bit Position	Bit Name	Function
7 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	SDERC	Diagnosis Error Clear Writing 0: Does not clear the diagnosis error. Writing 1: Clears the diagnosis error.
1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	OTFC	Temperature Measurement End Flag Clear Writing 0: Does not clear the temperature measurement end flag. Writing 1: Clears the temperature measurement end flag.

13.3.7 OTS0OTFR — Temperature Sensor Flag Register

OTS0OTFR is an 8-bit read-only register to indicate temperature sensor flags.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	SDER	OTACT	OTF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 13.11 OTS0OTFR Register Contents

Bit Position	Bit Name	Function
7 to 3	—	Reserved When read, the value after reset is returned.
2	SDER	Diagnosis Error Setting condition A diagnosis error occurred. Clearing condition Write SDERC to 1 in OTS0OTFCR
1	OTACT	Temperature Sensor Status 0: The temperature sensor is in the idle state. 1: The temperature sensor is operating.
0	OTF	Temperature Measurement End Flag Setting condition A temperature measurement value is written to OTS0OTDR Clearing condition Write OTFC to 1 in OTS0OTFCR. Reading of the OTS0OTDR register.

CAUTION

SDER is updated when a temperature measurement value is written to OTS0OTDR.

13.3.8 OTS0OTSTR — Temperature Status Register

OTS0OTSTR is an 8-bit read-only register to indicate temperature sensor status.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TSTAT[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 13.12 OTS0OTSTR Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned.
1 to 0	TSTAT[1:0]	Temperature Status 00: Normal temperature (value after reset) 01: High temperature B 10: High temperature A 11: Low temperature A When temperature is measured, temperature status is determined by the temperature measurement value and the status is reflected in TSTAT[1:0]. Figure 13.2 shows the status transitions of TSTAT, and Table 13.13 shows the temperature alarm error and temperature rise/drop interrupt generating conditions.

CAUTION

The TSTAT[1:0] bits are updated when a temperature measurement value is written to OTS0OTDR.

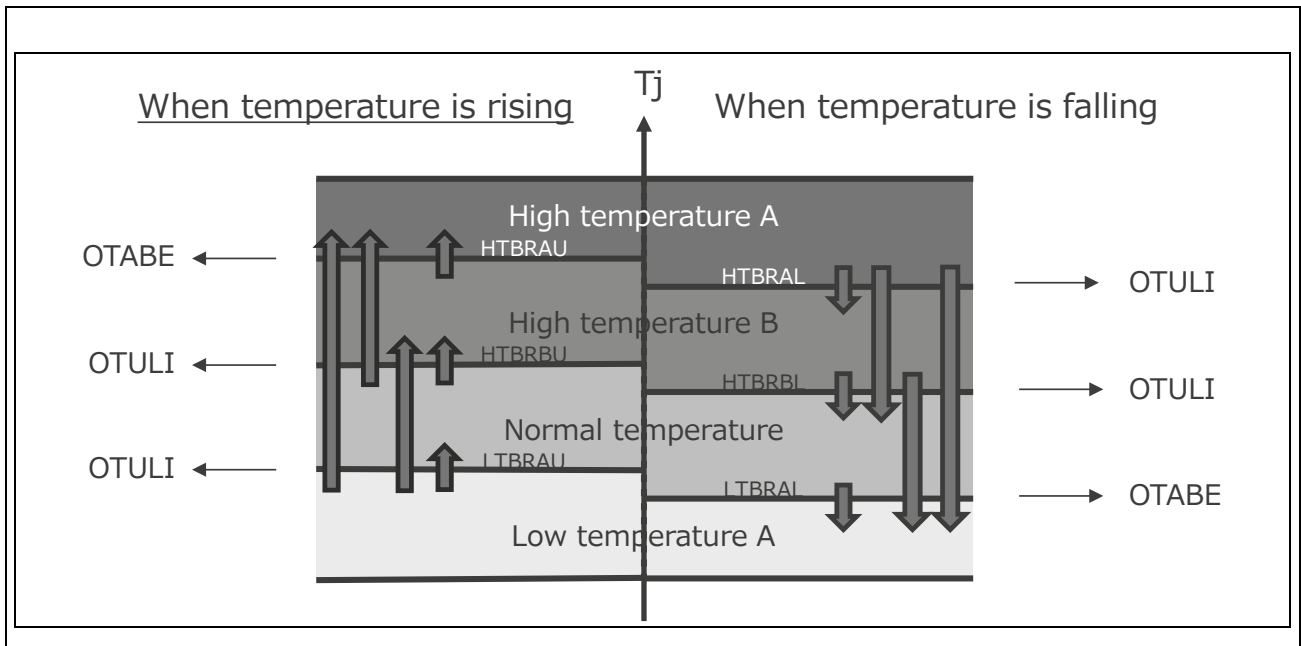


Figure 13.2 Temperature Status and Transition States

Table 13.13 Conditions for Generating Temperature Alarm Error and Temperature Rise/Drop Interrupts

		After Status Transition			
		High temp. A	High temp. B	Normal temp.	Low temp. A
Before status transition	High temp. A	N/A	INTOTSOTULI	INTOTSOTULI	INTOTSOTABE
	High temp. B	INTOTSOTABE	N/A	INTOTSOTULI	INTOTSOTABE
	Normal temp.	INTOTSOTABE	INTOTSOTULI	N/A	INTOTSOTABE
	Low temp. A	INTOTSOTABE	INTOTSOTULI	INTOTSOTULI	N/A

[Legend]

INTOTSOTABE: Occurrence of temperature alarm error

INTOTSOTULI: Occurrence of temperature rise or drop interrupt

N/A: Neither INTOTSOTABE nor INTOTSOTULI occurs.

13.3.9 OTS0OTDR — Temperature Data Register

OTS0OTDR is a 16-bit read-only register that stores a temperature measurement value.

Value after reset: 0000_H

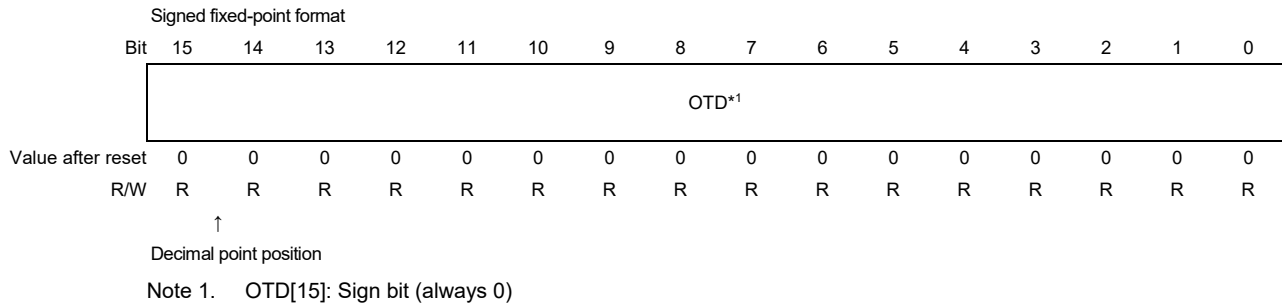


Table 13.14 OTS0OTDR Register Contents

Bit Position	Bit Name	Function
15 to 0	OTD	Temperature Data Value These bits specify a value for temperature data. OTD[15] is always 0.

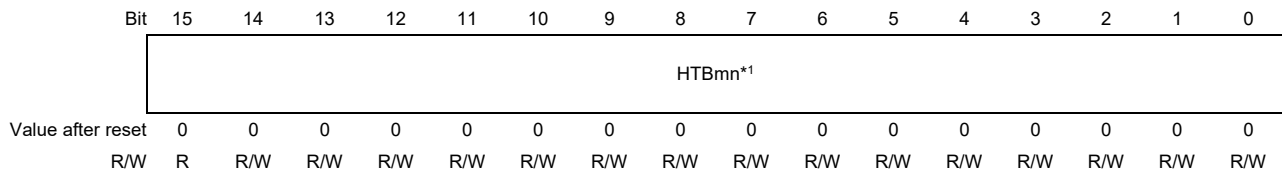
NOTE

When the CPU reads the register while the temperature sensor module is updating the register, the value before updating will be read. **Figure 13.4** show the updating of this register.

13.3.10 OTS0HTBRmn — High-Temperature Border mn Register (m = A or B, n = U or L)

OTS0HTBRmn is a 16-bit readable/writable register to specify the temperature border of high temperature m.

Value after reset: 0000_H



Note 1. HTBmn[15]: Sign bit (always 0)

Table 13.15 OTS0HTBRmn Register Contents

Bit Position	Bit Name	Function
15 to 0	HTBmn	<p>High-Temperature Border mn</p> <p>These bits specify the high-temperature border mn. Specify it so that the following condition is met.</p> <p style="padding-left: 40px;">OTS0HTBRAU > OTS0HTBRAL > OTS0HTBRBU > OTS0HTBRBL > OTS0LTBRAU > OTS0LTBRAL</p> <p>The HTBmn format is the same as the OTS0OTDR format. HTBmn[15] is always 0.</p> <p>For details, see Section 13.3.8, OTS0OTSTR — Temperature Status Register.</p>

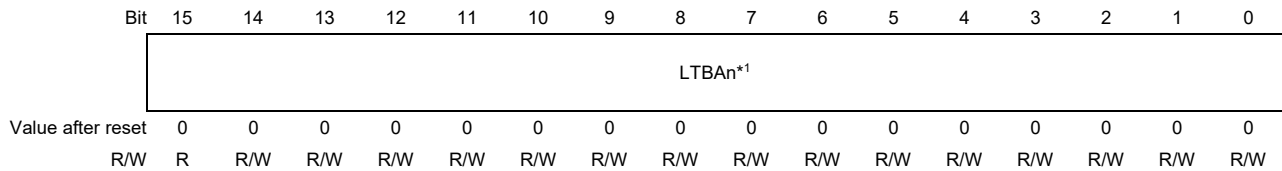
CAUTION

To prevent malfunction, set OTS0HTBRmn while the OTACT bit in OTS0OTFR is 0.

13.3.11 OTS0LTBRAn — Low-Temperature Border An Register (n = U or L)

OTS0LTBRAn is a 16-bit readable/writable register to specify the temperature border of low temperature A.

Value after reset: 0000_H



Note 1. LTBA_n[15]: Sign bit (always 0)

Table 13.16 OTS0LTBRAn Register Contents

Bit Position	Bit Name	Function
15 to 0	LTBA _n	<p>Low-Temperature Border An</p> <p>These bits specify the low-temperature border An. Specify it so that the following condition is met.</p> <p style="padding-left: 20px;">OTS0HTBRAU > OTS0HTBRAL > OTS0HTBRBU > OTS0HTBRBL > OTS0LTBRAU > OTS0LTBRAL</p> <p>The LTBA_n format is the same as the OTS0OTDR format. LTBA_n[15] is always 0.</p> <p>For details, see Section 13.3.8, OTS0OTSTR — Temperature Status Register.</p>

CAUTION

To prevent malfunction, set OTS0LTBRAn while the OTACT bit in OTS0OTFR is 0.

13.3.12 OTS0TDLR — Temperature Difference Limiting Register

OTS0TDLR is a 16-bit readable/writable register to specify the temperature difference limit value for self-diagnosis.

Value after reset: 7FFF_H

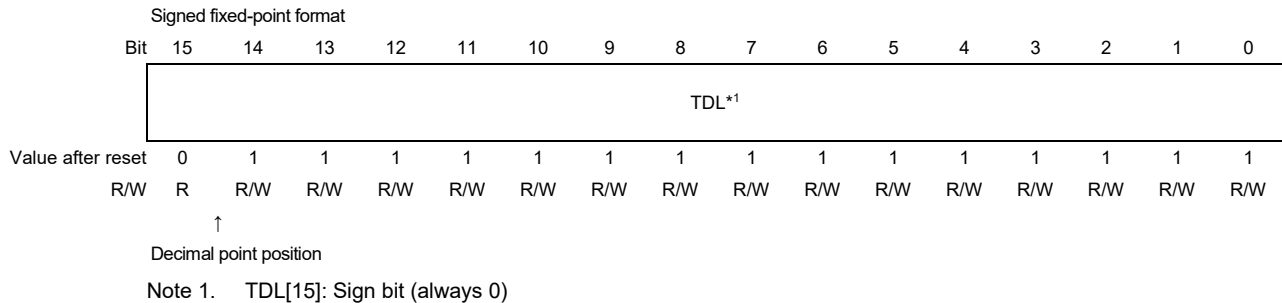


Table 13.17 OTS0TDLR Register Contents

Bit Position	Bit Name	Function
15 to 0	TDL	<p>Temperature Difference Limit Value</p> <p>These bits specify a value for limiting the temperature difference between present temperature measurement value and previous temperature measurement value in continuous measurement mode. When the following condition is met, SDER (diagnosis error) is set to 1.</p> $\text{OTS0TDLR} < \text{present temperature measurement value} - \text{previous temperature measurement value} $ <p>To calculate from the temperature, calculate back from the formula described in the OTS0COEFFRn register ($T[K] = AX^2 + BX + C$). For details, see Section 13.3.13, OTS0COEFFRn — Coefficient n Register (n = A, B, C).</p> <p>The TDL format is the same as the OTS0OTDR format. TDL[15] is always 0. For details, see Section 13.3.8, OTS0OTSTR — Temperature Status Register.</p>

CAUTION

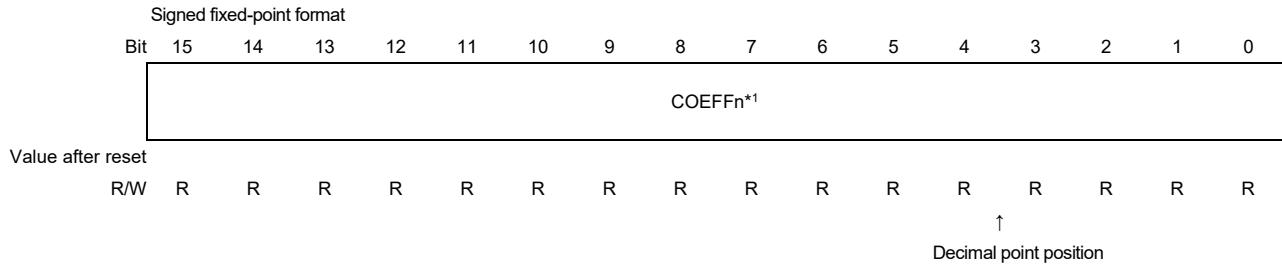
To prevent malfunction, set OTS0TDLR while the OTACT bit in OTS0OTFR is 0.

13.3.13 OTS0COEFFRn — Coefficient n Register (n = A, B, C)

OTS0COEFFRn is a 16-bit read-only register to store coefficients for correction calculation.

Coefficients are stored in OTS0COEFFRn during device manufacturing.

Value after reset: The value after reset will be stored at delivery inspection for each device.



Note 1. COEFFn[15]: Sign bit (always 0)

Table 13.18 OTS0COEFFRn Register Contents

Bit Position	Bit Name	Function
15 to 0	COEFFn	Coefficient values for correction calculation These bits specify the values of the coefficients. COEFFn[15] is always 0.

- Reducing errors by temperature correction

Three temperature correction coefficients are stored during device manufacturing in coefficient registers A to C.

Errors in temperature measurement values can be reduced by making a correction by using the temperature correction calculation formula that uses the temperature correction coefficients. However, temperature corrections must be made by the CPU. By using this process, the Temperature Border setting can be improved.

- Correction calculation formula: $T[K] = AX^2 + BX + C$

- T: Temperature value after correction
- X: Temperature value before correction (OTS0OTDR)
- A to C: Coefficients A to C (OTS0COEFFRn n = A, B, C)

For example

$$\text{OTS0OTDR} = 36\text{DA}_\text{H}, \text{OTS0COEFFRA} = 2106_\text{H}, \text{OTS0COEFFRB} = 2\text{D87}_\text{H}, \text{OTS0COEFFC} = 0040_\text{H}$$

$$X = \text{OTS0OTDR} / (2^{15}) = 0.428528$$

$$A = \text{OTS0COEFFRA} / (2^4) = 528.375$$

$$B = \text{OTS0COEFFRB} / (2^4) = 728.4375$$

$$C = \text{OTS0COEFFRC} / (2^4) = 4$$

$$T[^\circ\text{C}] = T[\text{K}] - 273.15 = AX^2 + BX + C - 273.15 = 140.035^\circ\text{C}$$

13.4 Operation

13.4.1 Temperature Measurement Sequence

The temperature sensor is equipped with a Single measurement mode and a Continuous measurement mode.

The measurement start trigger and measurement end trigger of each mode is shown below.

Table 13.19 Measurement Start/End Trigger

Measurement Mode	Measurement Start Trigger	Measurement End Trigger
Single measurement mode OTS0OTCR.OTMD = 0	Write 1 _B in OTS0OTSTCR.OTST.	1) Automatic stop after one temperature measurement ends 2) Write 1 _B in OTS0OTENDCR.OTEND.*1 *2
Continuous measurement mode OTS0OTCR.OTMD = 1	Write 1 _B in OTS0OTSTCR.OTST.	Write 1 _B in OTS0OTENDCR.OTEND.*1

Note 1. If the temperature measurement has been stopped by the OTS0OTENDCR.OTEND, data during the measurement will be discarded.

Note 2. If the temperature measurement has been stopped by the OTS0OTENDCR.OTEND before the first measurement has been completed, the measurement operation will not output the measurement result at all.

The time until the temperature data is output requires 512 states (9.984 ms.) of the OTS clock to convert the measured temperature to digital data.

There is a measurement stabilization period of 3 OTS clock states before the temperature is sampled after the measurement process is started by writing to the OTS0OTSSTCR.OTST bit.

In Single measurement mode, to update the temperature measurement result requires 515 OTS clock states, including the measurement preparation period, until the temperature measurement operation ends.

In Continuous measurement mode, 515 of the OTS clock states, including the 3-state measurement stabilization period, are required only for the first temperature measurement, the second and subsequent updates of the temperature measurement result require 512 states, and the measurement operation is repeated until 1 is written to OTENDCR.OTEND.

13.4.2 Examples of Temperature Measurement Operation

The following figure shows the temperature measurement operation in single measurement mode.

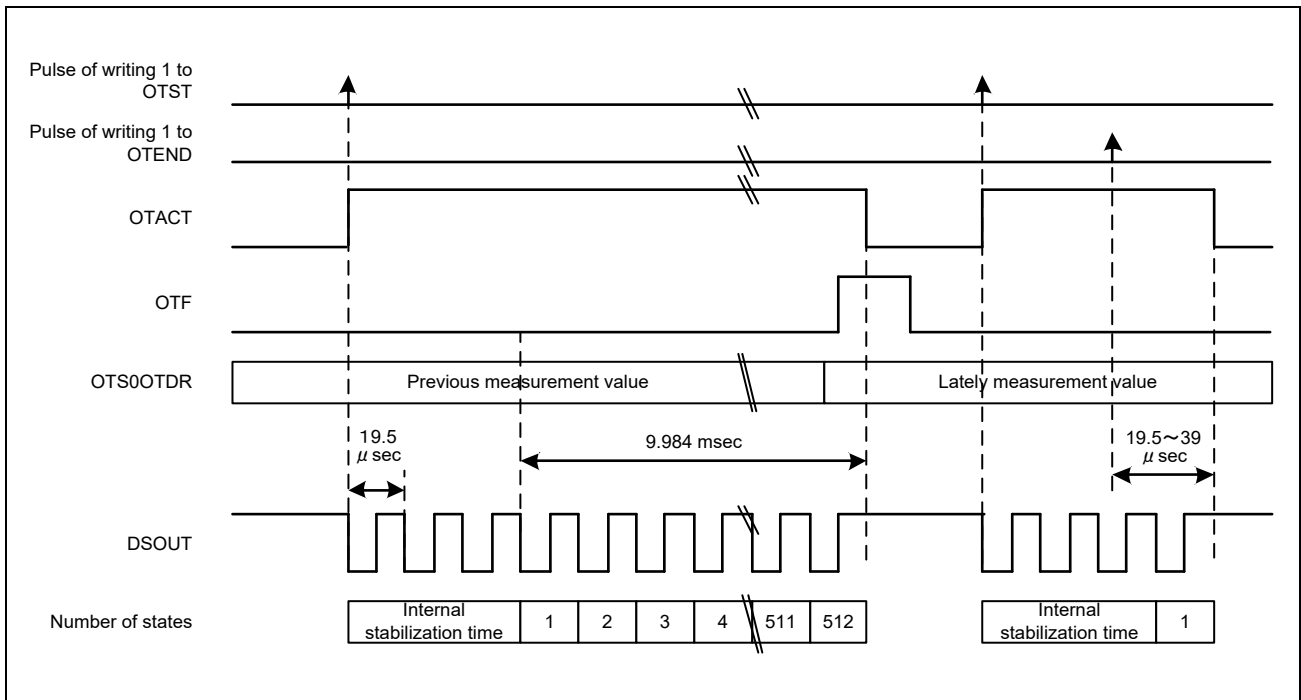


Figure 13.3 Example of Temperature Measurement Operation (Single Measurement Mode)

NOTE

If "1" is written to OTENDCR.OTEND at the time of internal cycle 511 or 512, request INTOTSOTI is issued and OTS0OTFR.OTF is set after updating the temperature measurement result as with the case of automatic stop after the temperature measurement.

DSOUT is the operation clock of the temperature sensor obtained by dividing CLK_LSB.

The period of DSOUT is 19.5 μs (When CLK_LSB 40 MHz)

The following figure shows temperature measurement operation in continuous measurement mode. There is no internal stabilization time in the second and subsequent temperature measurements.

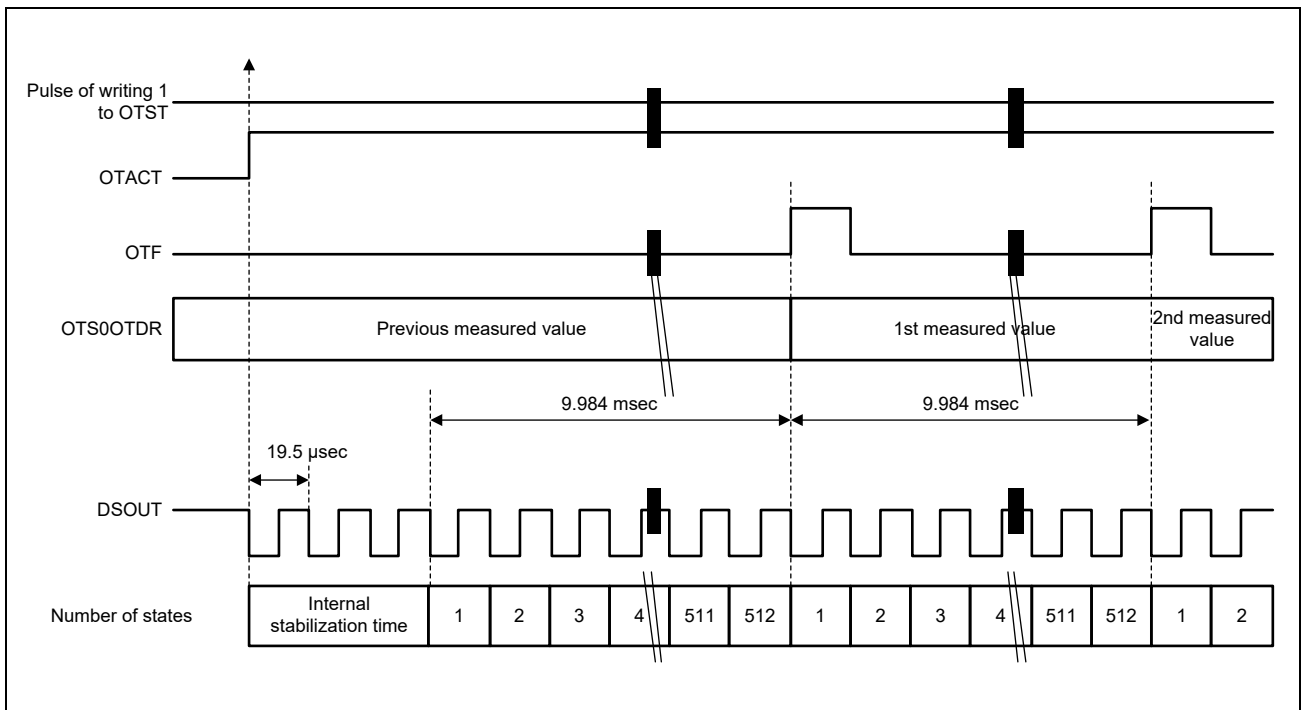


Figure 13.4 Example of Temperature Measurement Operation (Continuous Measurement Mode)

13.4.3 Temperature Measurement End Interrupt Request

The temperature sensor can generate a temperature measurement end interrupt request (INTOTSOTI) to be sent to the INTC. At this time, the OTS0OTFR.OTF is set.

The table below shows the assertion condition and negated conditions of INTOTSOTI and OTS0OTFR.OTF.

Table 13.20 Temperature Measurement End Conditions

Measurement End Notification Signal	Assert Conditions	Negated Conditions	Enable Control
Temperature measurement end interrupt (INTOTSOTI)	Each time temperature measurement ends	After the assertion, one cycle after the register access clock	No enable control bit.
Temperature measurement end flag (OTS0OTFR.OTF)	Each time temperature measurement ends	1) Reading of the OTS0OTDR register 2) Write 1 _B in OTS0OTFCR.OTFC	No enable control bit.

For the OTS0OTFR.OTF bit, if the assertion conditions and negated conditions conflict, negated condition takes precedence.

If the temperature measurement has been stopped by the OTS0OTENDCR.OTEND, neither the INTOTSOTI nor the OTS0OTFR.OTF will occur.

However, after writing “1” to the OTS0OTENDCR.OTEND, when the measurement is within 2 OTS clock cycles of being completed, the measurement result is updated and the INTOTSOTI and OTS0OTFR.OTF are asserted.

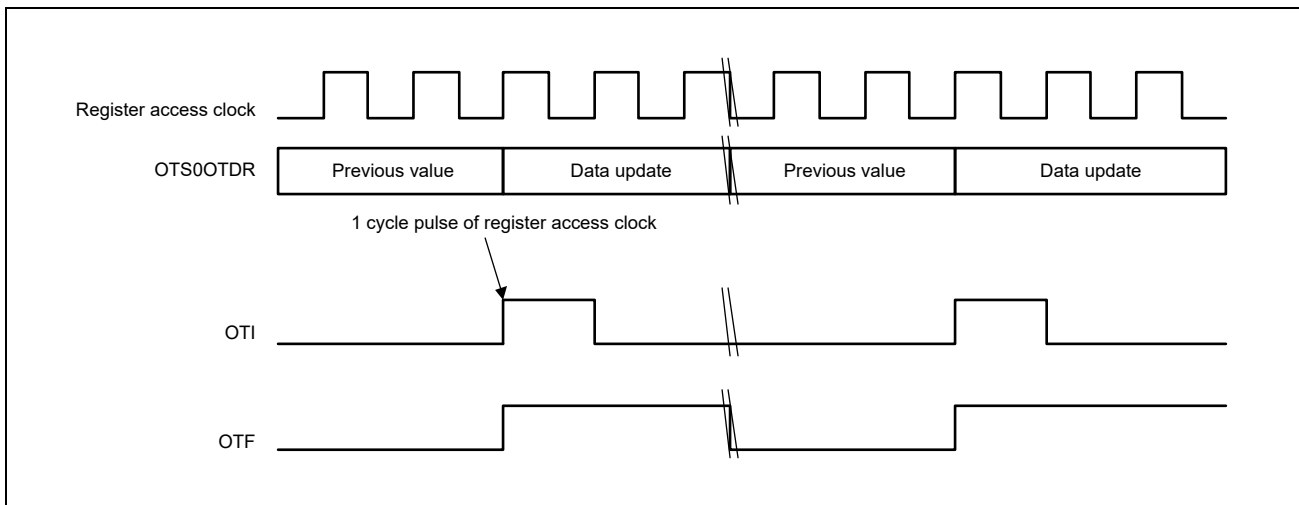


Figure 13.5 Example of Occurrence of a Temperature Measurement End Interrupt

13.4.4 Temperature Alarm Error and Temperature Rise/Drop Interrupt and Temperature Sensor Error Interrupt Requests

The temperature sensor can generate a temperature rise or drop interrupt request (INTOTSOTULI), a temperature sensor error interrupt request (INTOTSOTE) and a temperature alarm error (INTOTSOTABE) to be sent to the INTC. Setting OTULIE in OTS0OTCR to 1 enables INTOTSOTULI. Setting OTULIE to 0 disables INTOTSOTULI. Setting OTEE in OTS0OTCR to 1 enables INTOTSOTE. Setting OTEE to 0 disables INTOTSOTE. Setting OTABEE in OTS0OTCR to 1 enables INTOTSOTABE. Setting OTABEE to 0 disables INTOTSOTABE.

The table below shows the assert conditions and negated conditions of INTOTSOTULI and INTOTSOTABE.

Table 13.21 Temperature Sensor Interrupts

Temperature State Transition Notification Signal	Assert Conditions	Negated Conditions	Enable Control
Temperature rise or drop interrupt (INTOTSOTULI)	Temperature state transition from a state other than the high temperature B to high temperature B. Temperature state transition from a state other than the normal temperature to normal temperature.	After the assertion, one cycle after the register access clock	OTS0OTCR.OTULIE
Temperature alarm error (INTOTSOTABE)	Temperature state transition from a state other than the high temperature A to high temperature A. Temperature state transition from a state other than the low temperature A to low temperature A.	After the assertion, one cycle after the register access clock	OTS0OTCR.OTABEE
Temperature sensor error interrupt (INTOTSOTE)	After the OTDR register is updated, when the difference between the temperature measurements exceeds a set value TDL.	After the assertion, one cycle after the register access clock	OTS0OTCR.OTEE and OTS0OTCR.SDE
Diagnostic error flag (OTS0OTFR.SDER)	After the OTDR register is updated, when the difference between the temperature measurements exceeds a set value TDL.	Write 1 _B in OTS0OTFCR.SDERC	OTS0OTCR.SDE

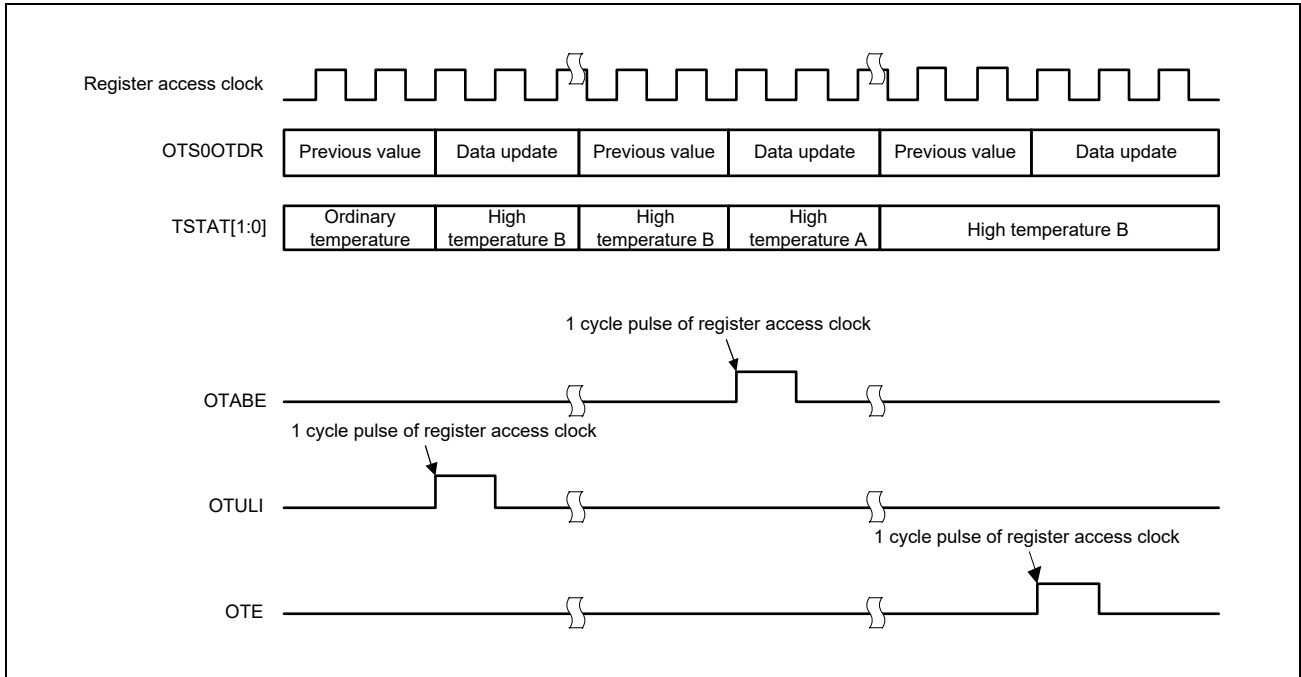


Figure 13.6 Example of Occurrence of a Temperature Alarm Error, a Temperature Rise/Drop Interrupt and a Temperature Sensor Error Interrupt

Section 14 Clock Monitor

14.1 Features

This section contains a generic description of clock monitor A (CLMA).

The first part of this section indicates all this product's specific properties, such as the number of channels and register base addresses.

The remainder of the section describes the functions and registers of CLMA.

14.1.1 Number of Channels

Table 14.1 Number of Channels

Clock Monitor	
Number of Channels	7
Name	CLMA _n (n = 0 to 6)

Note: CLMA4 is for the ICU-M. For details, refer to the relevant document.

14.1.2 Register Base Addresses

Table 14.2 Register Base Addresses

Base Address Name	Base Address	Bus group
<CLMAC_base>	FF70 2000 _H	Peripheral Group 6
<CLMA0_base>	FF70 2100 _H	Peripheral Group 6
<CLMA1_base>	FF70 2200 _H	Peripheral Group 6
<CLMA2_base>	FF70 2300 _H	Peripheral Group 6
<CLMA3_base>	FF70 2400 _H	Peripheral Group 6
<CLMA5_base>	FF70 2500 _H	Peripheral Group 6
<CLMA6_base>	FF70 2600 _H	Peripheral Group 6

14.1.3 Clock Supply

The clocks monitored by CLMA and the CLMA sampling clocks are indicated below.

For each clock frequency, see **Section 15, Clock Controller**.

Table 14.3 Clock Supply

Unit Name	CLMATMON (monitored clock)	CLMATSMP (sampling clock)	Register access clock
CLMA0	CLK_IOSC	CLK_MOSC / 8	CLK_LSB
CLMA1	CLK_MOSC	CLK_IOSC / 400	CLK_LSB
CLMA2	CLK_LSB	CLK_MOSC / 8	CLK_LSB
CLMA3	CLK_UHSB	CLK_MOSC / 8	CLK_LSB
CLMA5	CLK_CPU (PE0) / 4	CLK_MOSC / 8	CLK_LSB
CLMA6	CLK_CPU (PE1) / 4	CLK_MOSC / 8	CLK_LSB

14.1.4 Interrupt Requests

This module has no interrupt requests.

14.1.5 External Input and Output Pins

This module has no external input/output pins.

14.2 Overview

This product incorporates the clock monitors (CLMA) to monitor the clock operation by detecting the abnormal frequency of the clock to be monitored. The clock monitors provide the following functions.

- Monitors to see if the frequency of the clock to be monitored is within the specified range based on the sampling clock.
- Issues an error notice to the ECM upon detection of the abnormal state of the clock.

14.2.1 Functional Overview

Clock monitor CLMA detects frequency abnormalities in the monitored clock. It uses sampling clock CLMATSMPL to monitor whether the frequency of input clock CLMATMON is within a specific range.

Upon detection of an abnormal clock, it outputs a reset request signal.

14.2.2 Block Diagram

Figure 14.1 shows the block diagram of each clock monitor.

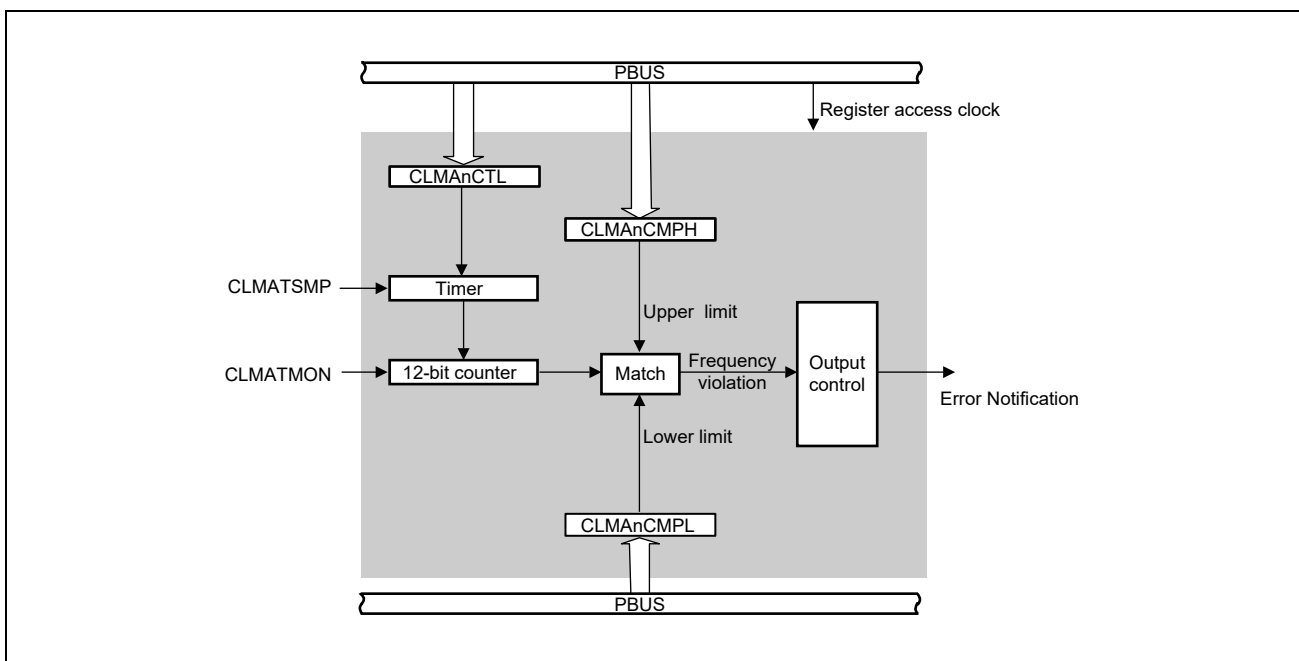


Figure 14.1 Block Diagram of Each Clock Monitor

14.3 Register

14.3.1 Register Protection

Write protected registers are protected from inadvertent write access due to erroneous program execution, etc.

By releasing the protection of CLMAKCPROT, they can be written. For details, see **Section 38, Functional Safety**.

14.3.2 List of Registers

The following table lists the CLMA registers.

<CLMAC_base> and <CLMA_n_base> are defined in **Section 14.1.2, Register Base Addresses**.

Table 14.4 List of Registers

Unit Name	Register Name	Symbol	Address	Access Size	Access Protection
CLMA _n	CLMA _n Control Register	CLMA _n CTL	<CLMA _n _base> + 000 _H	8	CLMAKCPROT
CLMA _n	CLMA _n Compare Register L	CLMA _n CMPL	<CLMA _n _base> + 008 _H	16	—
CLMA _n	CLMA _n Compare Register H	CLMA _n CMPH	<CLMA _n _base> + 00C _H	16	—
CLMAC	Clock Monitor Test Register	CLMATEST	<CLMAC_base> + 000 _H	32	CLMAKCPROT
CLMAC	Clock Monitor Test Status Register	CLMATESTS	<CLMAC_base> + 004 _H	32	—
CLMAC	Clock Monitor Backup Clock Enable Register	CLMABCE	<CLMAC_base> + 008 _H	8	CLMAKCPROT
CLMAC	Clock Monitor Register Key Code Protection Register	CLMAKCPROT	<CLMAC_base> + F00 _H	32	—

Note: (n = 0 to 3, 5 to 6)

14.3.3 Resetting the Registers

Table 14.5 Register Reset Conditions

Register Name	Reset Source					
	Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
All registers	✓	✓	✓	✓	—	—

Each clock monitor (CLMA_n) can also be reset by CLMATEST.RESCLM. In this case, it is necessary to perform a special procedure and reconfigure the CLMA_n parameters before the clock frequency is changed by the register for the operating clock.

For details about this special procedure, see **Section 14.5.2, Reset Procedure Using CLMATEST.RESCLM.**

Table 14.6 Individual Reset Conditions

Channel Name	CLMATEST.RESCLM Reset Target					
	CLMA0TESEN = 1	CLMA1TESEN = 1	CLMA2TESEN = 1	CLMA3TESEN = 1	CLMA5TESEN = 1	CLMA6TESEN = 1
CLMA0CTL	✓	—	—	—	—	—
CLMA0CMPL	✓	—	—	—	—	—
CLMA0CMPH	✓	—	—	—	—	—
CLMA1CTL	—	✓	—	—	—	—
CLMA1CMPL	—	✓	—	—	—	—
CLMA1CMPH	—	✓	—	—	—	—
CLMA2CTL	—	—	✓	—	—	—
CLMA2CMPL	—	—	✓	—	—	—
CLMA2CMPH	—	—	✓	—	—	—
CLMA3CTL	—	—	—	✓	—	—
CLMA3CMPL	—	—	—	✓	—	—
CLMA3CMPH	—	—	—	✓	—	—
CLMA5CTL	—	—	—	—	✓	—
CLMA5CMPL	—	—	—	—	✓	—
CLMA5CMPH	—	—	—	—	✓	—
CLMA6CTL	—	—	—	—	—	✓
CLMA6CMPL	—	—	—	—	—	✓
CLMA6CMPH	—	—	—	—	—	✓
CLMATEST	—	—	—	—	—	—
CLMATESTS	✓	✓	✓	✓	✓	✓
CLMAKCPROT	—	—	—	—	—	—

14.3.4 CLMACTL — CLMA Control Register

This register enables the clock monitor CLMA.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLMA _n CLME
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 14.7 CLMACTL Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CLMA _n CLME	Enables or disables the clock monitor: 0: Disables CLMA. 1: Enables CLMA. This bit can only be cleared by a reset

14.3.5 CLMA_nCMPL — CLMA_n Compare Register L

This register specifies the lower limit of the monitored clock frequency.

Write access is permitted only when CLMA_n is disabled (CLMA_nCTL.CLMA_nCLME = 0).

Value after reset: 0001_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CLMA _n CMPL[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.8 CLMA_nCMPL Register Contents

Bit Position	Bit Name	Function
15 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	CLMA _n CMPL[11:0]	Specify the lower threshold. When CLMA is used, these bits must be set appropriately. CLMA0: 25F _H CLMA1: 4BB _H CLMA2: FE _H CLMA3: 3FB _H CLMA5: 27D _H (400 MHz), 17E _H (240 MHz) CLMA6: 27D _H (400 MHz), 17E _H (240 MHz) The above is an example of setting when using EXTAL. (Clock condition EXTAL +0%/–0.5%, Internal OSC ±5%, PLL ±0.05% (Long term jitter)) When CLMA is not used, these bits must retain their values after a reset.

14.3.6 CLMA_nCMPH — CLMA_n Compare Register H

This register specifies the upper limit of monitored clock frequency.

Write access is permitted only when the CLMA_n is disabled (CLMA_nCTL.CLMA_nCLME = 0).

Value after reset: 03FF_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CLMA _n CMPH[11:0]											
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.9 CLMA_nCMPH Register Contents

Bit Position	Bit Name	Function
15 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	CLMA _n CMPH[11:0]	Specify the upper threshold. When CLMA is used, these bits must be set appropriately.. CLMA0: 2A5 _H CLMA1: 545 _H CLMA2: 102 _H CLMA3: 40A _H CLMA5: 286 _H (400 MHz), 182 _H (240 MHz) CLMA6: 286 _H (400 MHz), 182 _H (240 MHz) The above is an example of setting when using EXTAL. (Clock condition EXTAL +0%/-0.5%, Internal OSC ±5%, PLL ±0.05% (Long term jitter)) When CLMA is not used, these bits must retain their values after a reset.

14.3.7 CLMATEST — Clock Monitor Test Register

This register is used for the self-test of the clock monitors. Each Clock Monitor can be tested individually.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLMA6 TESEN	CLMA5 TESEN	CLMA3 TESEN	CLMA2 TESEN	CLMA1 TESEN	CLMA0 TESEN	ERR MSK	MONCL KMSK	RES CLM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.10 CLMATEST Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	CLMA6TESEN	CLMA6 self-test enable/disable 0: Test disabled. 1: Test enabled.
7	CLMA5TESEN	CLMA5 self-test enable/disable 0: Test disabled. 1: Test enabled.
6	CLMA3TESEN	CLMA3 self-test enable/disable 0: Test disabled. 1: Test enabled.
5	CLMA2TESEN	CLMA2 self-test enable/disable 0: Test disabled. 1: Test enabled.
4	CLMA1TESEN	CLMA1 self-test enable/disable 0: Test disabled. 1: Test enabled.
3	CLMA0TESEN	CLMA0 self-test enable/disable 0: Test disabled. 1: Test enabled.
2	ERRMSK* ¹	Masks an error notification to the ECM when CLMA _n detects an error. When the ERRMSK is set for a certain CLMA _n , the associated CLMA _n does not issue an error notification to the ECM even if it detects an error. 0: Does not mask an error notification to the ECM. 1: Masks an error notification to the ECM.
1	MONCLKMSK* ¹	Fixes the level of the clock input to the CLMA _n that should be monitored, to the low level. 0: Does not fix the clock input to the CLMA _n that should be monitored to the low level. 1: Fixes the clock input to the CLMA _n that should be monitored to the low level.
0	RESCLM* ¹	Initializes CLMA _n forcibly. 0: Does not initialize CLMA _n . 1: Initializes CLMA _n .

Note 1. These bits are valid for CLMA_n that is in self-diagnosis mode by setting “1” to CLMA_nTESEN.

14.3.8 CLMATESTS — Clock Monitor Test Status Register

This register is used for the self-test of the clock monitors. It monitors the error detection flags that are otherwise forwarded to the ECM module. Once error is detected, this register keeps the status until CLMA_n is reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	CLMA6 ERRS	CLMA5 ERRS	CLMA3 ERRS	CLMA2 ERRS	CLMA1 ERRS	CLMA0 ERRS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.11 CLMATESTS Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved When read, the value after reset is returned.
5	CLMA6ERRS	CLMA6 error status 0: No error detected 1: Error detected
4	CLMA5ERRS	CLMA5 error status 0: No error detected 1: Error detected
3	CLMA3ERRS	CLMA3 error status 0: No error detected 1: Error detected
2	CLMA2ERRS	CLMA2 error status 0: No error detected 1: Error detected
1	CLMA1ERRS	CLMA1 error status 0: No error detected 1: Error detected
0	CLMA0ERRS	CLMA0 error status 0: No error detected 1: Error detected

14.3.9 CLMABCE — Clock Monitor Backup Clock Enable Register

This register is a backup clock switching control register.

When CLMA1 or CLMA5 detects an error, the selector to select the startup clock source (CLK_IOOSC) or PLL Clock will be switched to choose the startup clock source (CLK_IOOSC).

For switching points, see **Section 15, Clock Controller**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLMABCE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 14.12 CLMABCE Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CLMABCE	Backup clock switching control 0: Disable the backup clock function 1: Enable the backup clock function

14.3.10 CLMAKCPROT — Clock Monitor Register Key Code Protection Register

The CLMAKCPROT register is used for protection against writing operation to the registers that may have a material effect on the system so that the application system is not incorrectly stopped due to program malfunction and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 14.13 CLMAKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0. *1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5 A500_H to this register to disable writing to protected registers.
Write A5A5 A501_H to this register to enable writing to protected registers.

14.4 Functional Description

The Clock Monitor CLMAn is used to verify whether the frequency of a clock (CLMATMON) is within the specified range.

14.4.1 Detection of Abnormal Clock Frequencies

Detection Method

- (1) CLMAn counts the rising edges of the monitored clock CLMATMON within 16 cycles of the sampling clock CLMATSMPL and then compares the count with the configured thresholds:
 - CLMAnCMPL.CLMAnCMPL[11:0] defines the lower threshold.
 - CLMAnCMPH.CLMAnCMPH[11:0] defines the upper threshold.
- (2) When the frequency of CLMATMON is too low*¹, the count falls below CLMAnCMPL.CLMAnCMPL.
- (3) When the frequency of CLMATMON is too high, the count exceeds CLMAnCMPH.CLMAnCMPH.

Note 1. If the monitored clock stops completely, an error may not be detected.

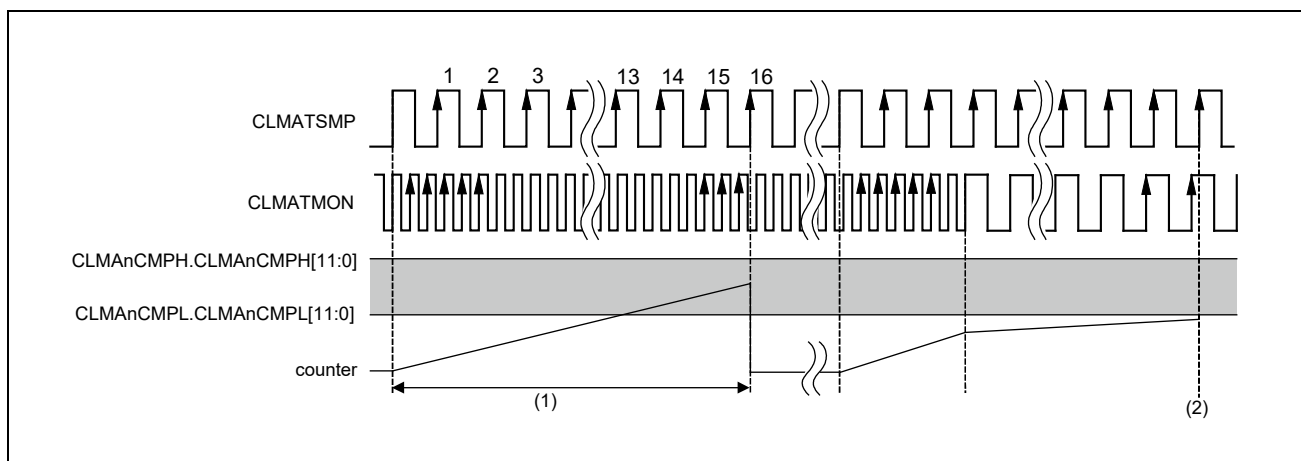


Figure 14.2 Example: f_{CLMATMON} is Lower than the Specified Limit.

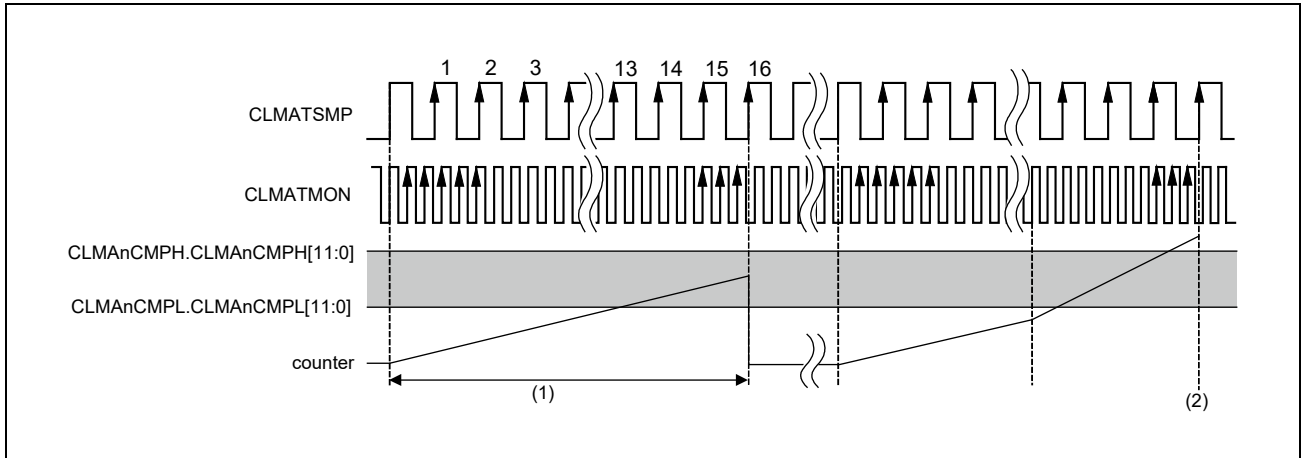


Figure 14.3 Example: $f_{CLMATMON}$ is Higher than the Specified Limit.

NOTE

Even if the $f_{CLMATMON}$ exceeds or falls below the specified limits during a sampling period, the count value may be within the valid range.
 Abnormal $f_{CLMATMON}$ is detected in the next sampling period.

Calculation of thresholds

For the compare registers CLMAnCMPL and CLMAnCMPH, specify the minimum and maximum number of CLMATMON clock cycles to occur within 16 cycles of the sampling clock CLMATSMPL that defines the normal range of CLMATMON.

The number of CLMATMON clock cycles to occur within 16 cycles of CLMATSMPL is denoted by N.

$$\frac{16}{f_{CLMATSMPL}} = \frac{N}{f_{CLMATMON}}$$

$$N = \frac{f_{CLMATMON}}{f_{CLMATSMPL}} \times 16$$

Considering the allowed frequency deviations of CLMATMON and CLMATSMPL, the threshold values can be calculated by the following formulas:

$$\begin{aligned} \text{Lower threshold} &= N_{min} \\ &= \frac{f_{CLMATMON(min)}}{f_{CLMATSMPL(max)}} \times 16 - 1 \end{aligned}$$

$$\begin{aligned} \text{Upper threshold} &= N_{max} \\ &= \frac{f_{CLMATMON(max)}}{f_{CLMATSMPL(min)}} \times 16 + 1 \end{aligned}$$

<Example>

For $f_{CLMATSMPL} = 250 \text{ kHz} (\pm 5\%)$ and $f_{CLMATMON} = 20 \text{ MHz} (\pm 0.1\%)$, the recommended threshold values are the following:

$$\begin{aligned} N_{min} &= 19.98 / 0.2625 \times 16 - 1 \\ &= 1216.83 \\ \text{CLMAnCMPL} &= 1216 = 04C0_H \\ \\ N_{max} &= 20.02 / 0.2375 \times 16 + 1 \\ &= 1348.72 \\ \text{CLMAnCMPH} &= 1349 = 0545_H \end{aligned}$$

<Minimum thresholds>

The following restrictions must be taken into account:

- CLMAnCMPL \geq 0001_H
- CLMAnCMPH \geq CLMAnCMPL + 0003_H

14.4.2 Error Notification

When the frequency of the monitored clock (CLMATMON) becomes too low or too high and an error is detected, CLMA_n sends an error notification to ECM. In addition, the error notification is not canceled until CLMA_n is reset.

When CLMA1 or CLMA5 detects an error, the selector to select the startup clock source (CLK_IOSC) or PLL Clock will be switched to choose the startup clock source (CLK_IOSC).

After the clock source is switched to CLK_IOSC when CLMA1 or CLMA5 detect an error, the clock source cannot be switched with the selection of CKSC0C.

NOTES

1. When CLK_IOSC is selected due to the back-up clock function, CLMA2 to CLMA6 will begin to detect errors.
2. Even if the monitor clock stops completely, the clock source may not be switched to startup clock source (CLK_IOSC). In addition, reset or interrupt may not occur from the ECM.

14.4.3 Self-Diagnosis

This LSI implements a self-diagnosis function for the clock monitors. It allows isolating the clock monitor from the system and executing functional test patterns via software. Each clock monitor can be tested individually. For performing these tests, the clock monitor inputs can be controlled by dedicated control registers, while its outputs can be observed in status registers.

Two registers are implemented for the self-diagnosis of the clock monitors, the Clock Monitor Test Register (CLMATEST) and the Clock Monitor Test Status Register (CLMATESTS). Refer to the corresponding register descriptions for details.

14.5 Operation

14.5.1 Procedures to Enable CLMA_n

[Procedure]

- (1) Enable write access of protected registers by setting CLMAKCPROT = A5A5 A501_H according to CLMA targets.
- (2) Set the minimum and maximum number of clock cycles to CLMA_nCMPL and CLMA_nCMPLH.
- (3) Set enable/disable of back up clock switching to the CLMABCE register.
- (4) Enable CLMA_n by setting 01_H to CLMA_nCTL.CLMA_nCLME.
- (5) Disable write access of protected registers by setting CLMAKCPROT.KCE = A5A5 A500_H according to CLMA targets.

CAUTION

- CLMA_n (n = 0, 1) operation must be started after users check stable state of MainOSC by referring to MOSCS.MOSCSTAB = 1.
- CLMA_n (n = 2 to 6) operation must be started after users check that the clock gear-up procedure has been finished.
- When Startup BIST is not executed by flash option at the reset of Power Up Reset, Standby Reset and VMON Reset with using crystal resonator, as soon as CPU boot up and wait until MOSCS.MOSCSTAB = 1, it is necessary to initialize CLMA_n (n = 0, 2 to 6) according to the procedure of **Section 14.5.2, Reset Procedure Using CLMATEST.RESCLM** and then clear all CLMA_n (n = 0, 2 to 6) related error flags presented in ECM master/checker error source status registers. For details of these error registers and how to clear them, please see **Section 39, Error Control Module (ECM)**.

14.5.2 Reset Procedure Using CLMATEST.RESCLM

Each clock monitor (CLMA0/1/2/3/5/6) can also be reset by CLMATEST.RESCLM. In this case, it is necessary to perform the following procedure and reconfigure the CLMA_n parameters before the clock frequency is changed by the register for the operating clock.

[Procedure]

Example: Resetting CLMA0

1. CLMATEST.CLMA0TESEN = 1 (write data: 0000 0008_H)
2. CLMATEST.ERRMSK = 1 (write data: 0000 000C_H)
3. CLMATEST.MONCLKMSK = 1 (write data: 0000 000E_H)
4. CLMATEST.RESCLM = 1 (write data: 0000 000F_H)
5. CLMATEST.RESCLM = 0 (write data: 0000 000E_H)
6. CLMATEST.MONCLKMSK = 0 (write data: 0000 000C_H)
7. CLMATEST.ERRMSK = 0 (write data: 0000 0008_H)
8. CLMATEST.CLMA0TESEN = 0 (write data: 0000 0000_H)

14.5.3 Procedures to do Self-Diagnosis

Self-diagnosis of the clock monitor is available as described below. In the description, the clock monitor to be self-diagnosed is operating.

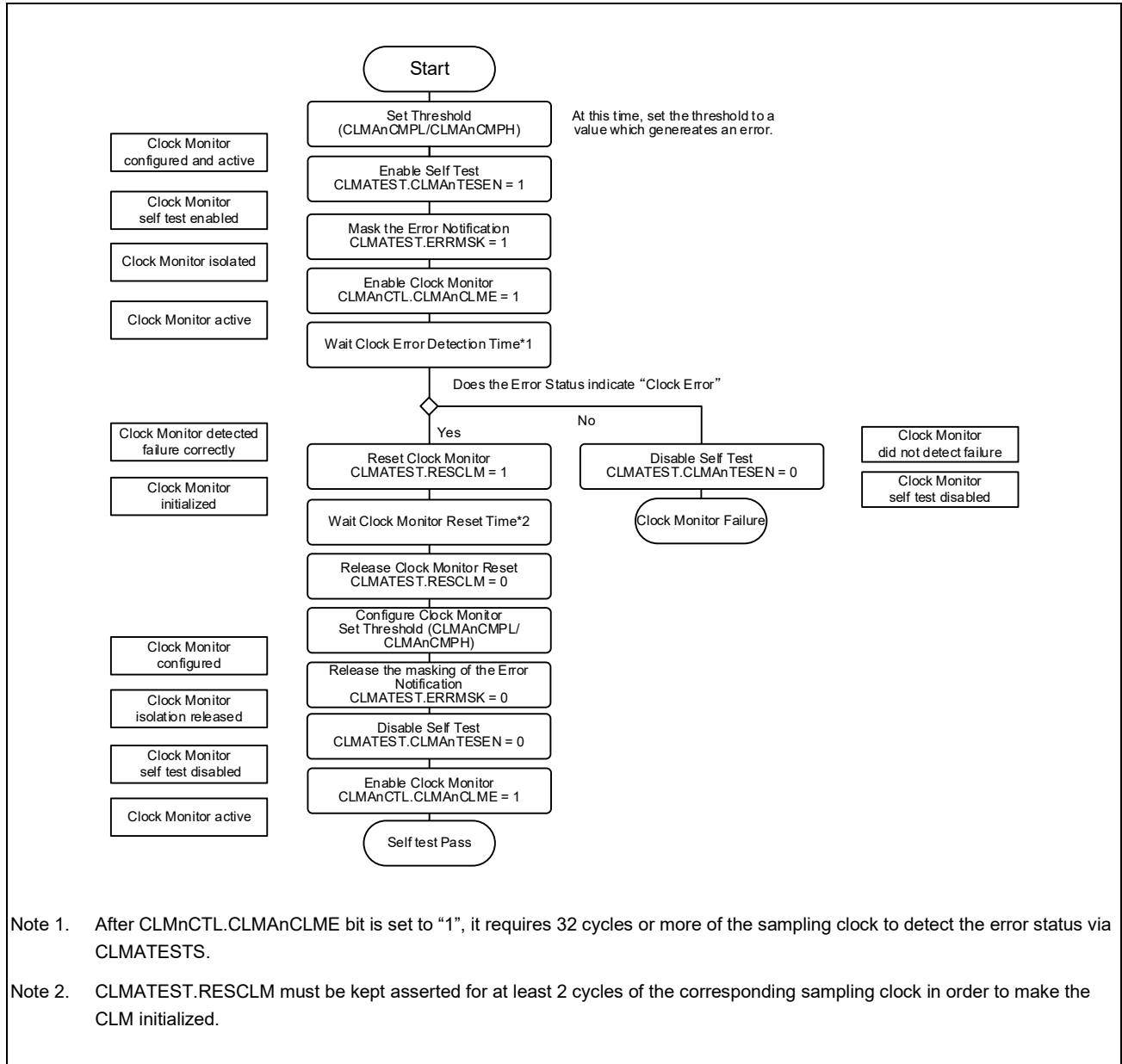


Figure 14.4 Flow Chart of Self-Diagnosis

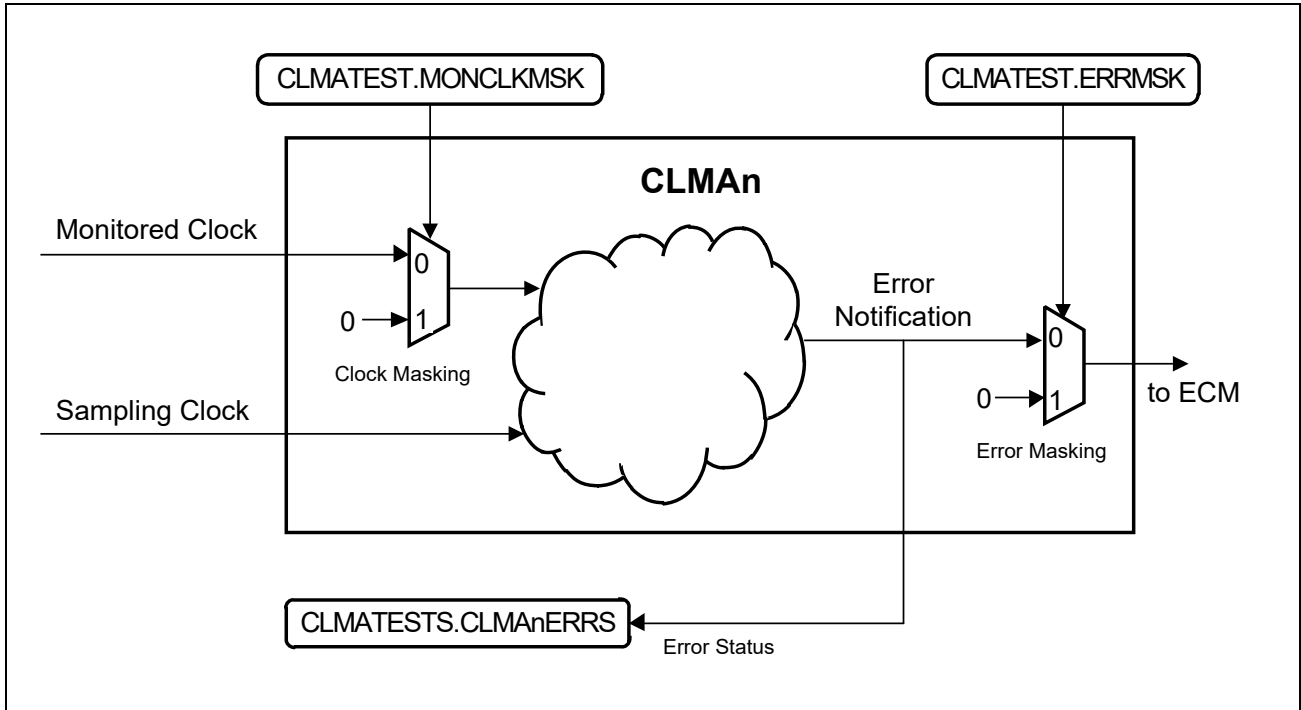


Figure 14.5 Self-Diagnosis of the Clock Monitor

Section 15 Clock Controller

The clock controller generates several clocks that are supplied to peripherals and to external pins for use by other devices. The clock controller consists of a main oscillator circuit (Main OSC), an internal oscillator, a Phase Locked Loop circuit (PLL), clock dividers and clock selectors. PLL uses the Main OSC as a reference clock. Use of Main OSC is mandatory.

15.1 Features

- Crystal resonator circuit (Main OSC), used as a PLL reference clock
- Internal oscillator, used as the startup and backup clock
- PLL circuit to generate high speed internal clocks by multiplying the Main OSC input
- Software configurable external clock output
- Clock controller can generate clock pulses from internal oscillator, main oscillator and PLL
- MainOsc Configuration is variable by OPBT7. Please refer to **Section 15.4.3, Main OSC Configuration and Stabilization Sequence** and **Section 15.4.4, OSC Monitor Mode**

15.1.1 Register Base Addresses

The clock controller base address is shown in the following table.

Clock controller register addresses are given as offsets from the base addresses in general.

Table 15.1 Register Base Addresses

Base Address Name	Base Address	Bus Group
<CLK_base>	FF70 3000 _H	Peripheral Group 6

15.1.2 External Input / Output Pins

Table 15.2 shows the pins related to the clock controller.

Table 15.2 Pins Related to the Clock Controller

Pin Function Name	Direction	Function
EXTAL	Input	Main OSC crystal resonator / external clock input
XTAL	Output	Main OSC crystal resonator
CK	Output	External clock out (divider 2)

15.2 Overview

15.2.1 Types of Clocks

The clock sources are listed in **Table 15.3**, and the clock frequencies based on different clock sources are listed in **Table 15.4** and **Table 15.5**.

Table 15.3 List of Clock Sources

Clock Name	Symbol	Clock Frequency (MHz)	Remarks
EXTAL, XTAL	EXTAL, XTAL	20/40	The frequency of EXTAL can be selected by using an option byte.* ¹ Main OSC crystal resonator or external clock input
MainOSC	CLK_MOSC	20	CLK_MOSC is the 20MHz clock after selected by using an option byte.* ¹
Internal OSC	CLK_IOSC	100	
WDTB/SWDT counter clock	CLK_WDT	0.25	WDTB / SWDT counter clock source can be selected by using an option byte. * ¹ Clock source: CLK_MOSC* ² or CLK_IOSC
ICU-M (WDTA) counter clock / ECM counter clock	CLK_WDTICUM	0.25	Clock source: CLK_IOSC
HVIOSC (VMON filter clock)	CLK_HVIOSC	16	
External clock out	CK	Divider 2 setting Maximum output frequency: 4 MHz	Clock source: CLK_MOSC or CLK_LSB

Note 1. See **Section 41, Flash Memory** for target values.

Note 2. WDTB auto-start by CLK_MOSC is prohibited if BIST is not executed.

Table 15.4 Clock Frequencies with PLL as the Clock Source

Clock Name	Symbol	Clock Frequency Mode (MHz)		Remarks
		400 MHz Mode	240 MHz Mode	
CPU clock	CLK_CPU	400	240	The clock frequency can be divided by Divider 0 (CLKD0DIV) settings. Clock source: PLL clock
System Bus clock	CLK_SBUS	200	80	
H-Bus clock	CLK_HBUS	100	80	
Peripheral ultra high speed clock	CLK_UHSB	160	160	
Peripheral high speed clock	CLK_HSB	80	80	
Peripheral low speed clock	CLK_LSB	40	40	
RHSB clock	CLK_RHSB	80/160	80/160	The RHSB clock source can be selected by using an option byte.* ¹ Clock source: CLK_HSB or CLK_UHSB

Note 1. See **Section 41, Flash Memory** for target values.

Table 15.5 Clock Frequencies with Internal OSC as the Clock Source

Clock Name	Symbol	Clock Frequency Mode (MHz)		Remarks
		400 MHz Mode	240 MHz Mode	
CPU clock	CLK_CPU	100	100	If the source clock is CLK_IOSC, then the CPU clock is 100 MHz in all modes, but the peripheral clocks are different in each mode. This is because the division ratio between the CPU clock and peripheral clocks is fixed in each mode.
System Bus clock	CLK_SBUS	50	33.33	
H-Bus clock	CLK_HBUS	25	33.33	
Peripheral ultra high speed clock	CLK_UHSB	40	66.67	
Peripheral high speed clock	CLK_HSB	20	33.33	
Peripheral low speed clock	CLK_LSB	10	16.67	
RHSB clock	CLK_RHSB	20/40	33.33/66.67	The RHSB clock source can be selected by using the option bytes.* ¹ Clock source: CLK_HSB or CLK_UHSB

Note 1. Refer to **Section 41, Flash Memory** for the target values

15.2.2 Block Diagram

A block diagram of the clock controller is shown in **Figure 15.1**.

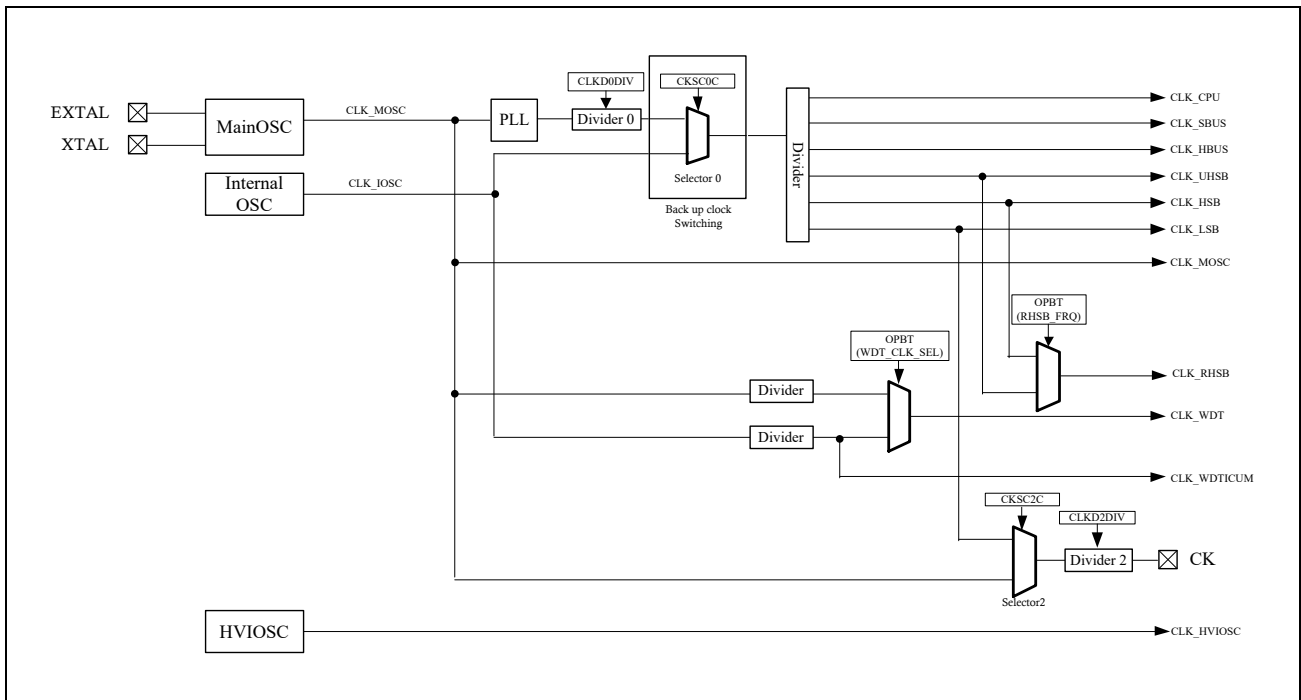


Figure 15.1 Block Diagram of Clock Controller

15.3 Register Description

15.3.1 Register Protection

Write protection prevents inadvertent changing of register content due to erroneous software execution.

The CLKKCPROT0 and CLKKCPROT1 registers have to be written first in order to unlock protected registers and allow changing the register content.

15.3.2 List of Registers

The following table lists the clock controller registers.

Table 15.6 List of Clock Controller Registers

Unit Name	Register Name	Symbol	Address	Access Size	Access Protection
SYSCTRL	Clock Divider 0 Divisor Register	CLKD0DIV	<CLK_base> + 000 _H	32	CLKKCPROT0
SYSCTRL	Clock Divider 0 Status Register	CLKD0STAT	<CLK_base> + 004 _H	32	—
SYSCTRL	Clock Divider 2 Divisor Register	CLKD2DIV	<CLK_base> + 010 _H	32	CLKKCPROT0
SYSCTRL	Clock Divider 2 Status Register	CLKD2STAT	<CLK_base> + 014 _H	32	—
SYSCTRL	Clock Selector 0 Control Register	CKSC0C	<CLK_base> + 100 _H	32	CLKKCPROT0
SYSCTRL	Clock Selector 0 Status Register	CKSC0S	<CLK_base> + 108 _H	32	—
SYSCTRL	Clock Selector 2 Control Register	CKSC2C	<CLK_base> + 180 _H	32	CLKKCPROT0
SYSCTRL	Clock Selector 2 Status Register	CKSC2S	<CLK_base> + 188 _H	32	—
SYSCTRL	PLL Oscillation Status Register	PLLCLKS	<CLK_base> + 200 _H	32	—
SYSCTRL	Main Oscillation Status Register	MOSCS	<CLK_base> + 400 _H	32	—
SYSCTRL	OSC Monitor Mode Entry Register	OSCMONMD	<CLK_base> + 404 _H	32	CLKKCPROT1
SYSCTRL	Clock Controller Register Key Code Protection Register 0	CLKKCPROT0	<CLK_base> + 700 _H	32	—
SYSCTRL	Clock Controller Register Key Code Protection Register 1	CLKKCPROT1	<CLK_base> + 710 _H	32	—

15.3.3 Reset of Registers

Table 15.7 Reset Sources Applicable to Each Register

Symbol	Reset Source					
	Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
CLKD0DIV	✓	✓	✓	—	—	—
CLKD0STAT	✓	✓	✓	—	—	—
CLKD2DIV	✓	✓	✓	✓	—	—
CLKD2STAT	✓	✓	✓	✓	—	—
CKSC0C	✓	✓	✓	—	—	—
CKSC0S	✓	✓	✓	—	—	—
CKSC2C	✓	✓	✓	✓	—	—
CKSC2S	✓	✓	✓	✓	—	—
PLLCLKS	✓	✓	✓	—	—	—
MOSCS	✓	✓	✓	—	—	—
OSCMONMD	✓	✓	✓	—	—	—
CLKKCPROT0	✓	✓	✓	✓	—	—
CLKKCPROT1	✓	✓	✓	—	—	—

15.3.4 CLKD0DIV — Clock Divider 0 Divisor Register

This register defines the division ratio of the clock divider 0 used for system clock generation.

This register must not be written with a new value while CLKD0SYNC is deasserted.

Value after reset: 0000 0002_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKD0DIV[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 15.8 CLKD0DIV Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	CLKD0DIV[2:0]	These bits select the frequency division ratio of divider 0. 001 _B : No division 010 _B : Divided by 2 Settings other than the above are prohibited.

15.3.5 CLKD0STAT — Clock Divider 0 Status Register

This register indicates the status of clock divider 0 used for system clock generation.

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKD0 SYNC	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15.9 CLKD0STAT Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned.
1	CLKD0SYNC	Divider clock synchronized 0: The system and peripheral clock do not correspond to the division ratio setting in CLKD0DIV. 1: The system and peripheral clock correspond to the division ratio setting in CLKD0DIV.
0	—	Reserved When read, the value after reset is returned.

15.3.6 CLKDnDIV — Clock Divider Divisor Register (n = 2)

This register defines the division ratio of clock divider n (n = 2) used for the external clock.

This register must not be written with a new value while CLKDnSYNC is deasserted.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	CLKDnDIV[4:0]				—	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R

Table 15.10 CLKDnDIV Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5 to 1	CLKDnDIV[4:0]	Division ratio 0: The external clock is stopped. Other bits: For the settings, see Table 15.11 .
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 15.11 Mapping of CLKDnDIV Settings to Output Frequencies

CLKDnDIV[4]	CLKDnDIV[3]	CLKDnDIV[2]	CLKDnDIV[1]	CLKDnDIV[0]	Output Frequency [MHz] (Each Clock Source)	
					CLK_MOSC	CLK_LSB
0	0	1	0	1	2.00	4.00
0	0	1	1	0	1.67	3.33
0	0	1	1	1	1.43	2.86
0	1	0	0	0	1.25	2.50
0	1	0	0	1	1.11	2.22
0	1	0	1	0	1.00	2.00
0	1	0	1	1	0.91	1.82
0	1	1	0	0	0.83	1.67
0	1	1	0	1	0.77	1.54
0	1	1	1	0	0.71	1.43
0	1	1	1	1	0.67	1.33
1	0	0	0	0	0.63	1.25
1	0	0	0	1	0.59	1.18
1	0	0	1	0	0.56	1.11
1	0	0	1	1	0.53	1.05
1	0	1	0	0	0.50	1.00

Note: Settings other than above are prohibited.

15.3.7 CLKDnSTAT — Clock Divider Status Register (n = 2)

This register indicates the status of clock divider n (n = 2) used for the external clock .

Value after reset: 0000 0002_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKDn SYNC	CLKDn CLKACT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15.12 CLKDnSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned.
1	CLKDnSYNC	Divider clock synchronized 0: The external clock does not correspond to the division ratio setting in CLKDnDIV. 1: The external clock corresponds to the division ratio setting in CLKDnDIV.
0	CLKDnCLKACT	Divider clock active 0: The external clock is inactive (CLKDnDIV[4:0] = 00000 _B) 1: The external clock is active

15.3.8 CKSC0C — Clock Selector 0 Control Register

This register selects the clock that drives the system and peripheral clock trees.

This register must not be written with a new value while CKSC0S shows a different ID.

Value after reset: 0000 0002_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CKSC0[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 15.13 CKSC0C Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	CKSC0[2:0]	Clock source control These bits select the ID of a clock source for the system and peripheral clock. ID = 001 _B : PLL ID = 010 _B : Internal OSC Other than the above: Setting prohibited.

15.3.9 CKSC0S — Clock Selector 0 Status Register

This register indicates the status of clock selector 0.

Value after reset: 0000 0002_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKACT0[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15.14 CKSC0S Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is returned.
2 to 0	CLKACT0[2:0]	Clock source status Indicates the ID of the selected clock source of the system and peripheral clock. The value indicated by these bits shows the actual ID that is currently selected or currently waiting to be changed to another ID. ID = 001 _B : PLL ID = 010 _B : Internal OSC

15.3.10 CKSCnC — Clock Selector Control Register (n = 2)

This register selects the clock that drives the CK clock .

This register must not be written with a new value while CKSCnS shows a different ID.

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CKSCn[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 15.15 CKSCnC Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	CKSCn[2:0]	Clock source control These bits select the ID of a clock source for the external clock. ID = 011 _B : CLK_MOSC ID = 100 _B : CLK_LSB Other than above: Setting prohibited.

15.3.11 CKSCnS — Clock Selector Status Register (n = 2)

This register indicates the status of clock selector n.

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKACTn[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15.16 CKSCnS Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is returned.
2 to 0	CLKACTn[2:0]	Clock source status Indicates the ID of the selected clock source of the external clock. The value indicated by these bits shows the actual ID that is currently selected or currently waiting to be changed to another ID. ID = 011 _B : CLK_MOSC ID = 100 _B : CLK_LSB

15.3.12 PLLCLKS — PLL Oscillation Status Register

This register indicates the PLL oscillation status.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLK STAB	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 ^{*1}	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. Depends on the read timing after the CPU starts operation.

Table 15.17 PLLCLKS Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned.
1	CLKSTAB	PLL clock stable state 0: PLL is unstable. 1: PLL is stable.
0	—	Reserved When read, the value after reset is returned.

15.3.13 MOSCS — Main Oscillation Status Register

This register indicates the main oscillator status.

Value after reset: 0000 000x_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSC STAB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 ^{*1}
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. Depends on the read timing after the CPU starts operation.

Table 15.18 MOSCS Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned.
0	MOSCSTAB	Main OSC clock stable state 0: Main OSC clock is unstable. 1: Main OSC clock is stable.

15.3.14 OSCMONMD — OSC Monitor Mode Entry Register

This register enables OSC monitor mode.

Writing to this register is enabled when CKSC0C.CKSC0[2:0] = 2_H.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OSCMONMD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 15.19 OSCMONMD Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	OSCMONMD	OSC Monitor mode entry bit 0: Disable OSC monitor mode 1: Enable OSC monitor mode

15.3.15 CLKKCPROT0 — Clock Controller Register Key Code Protection Register 0

The CLKKCPROT0 register is used to protect against writing to registers that may have a material effect on the system, so that the application system is not incorrectly stopped by a program malfunction, etc.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 15.20 CLKKCPROT0 Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5 A500_H to this register to disable writing protected registers.
Write A5A5 A501_H to this register to enable writing protected registers.

15.3.16 CLKKCPROT1 — Clock Controller Register Key Code Protection Register 1

The CLKKCPROT1 register is used to protect against writing to registers that may have a material effect on the system, so that the application system is not incorrectly stopped by a program malfunction, etc.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 15.21 CLKKCPROT1 Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0. *1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5 A500_H to this register to disable writing protected registers.
Write A5A5 A501_H to this register to enable writing protected registers.

15.4 Operation

15.4.1 Sequence for Shifting the Clock Gear Up

To switch the clock signals, follow the procedure shown in **Figure 15.2**.

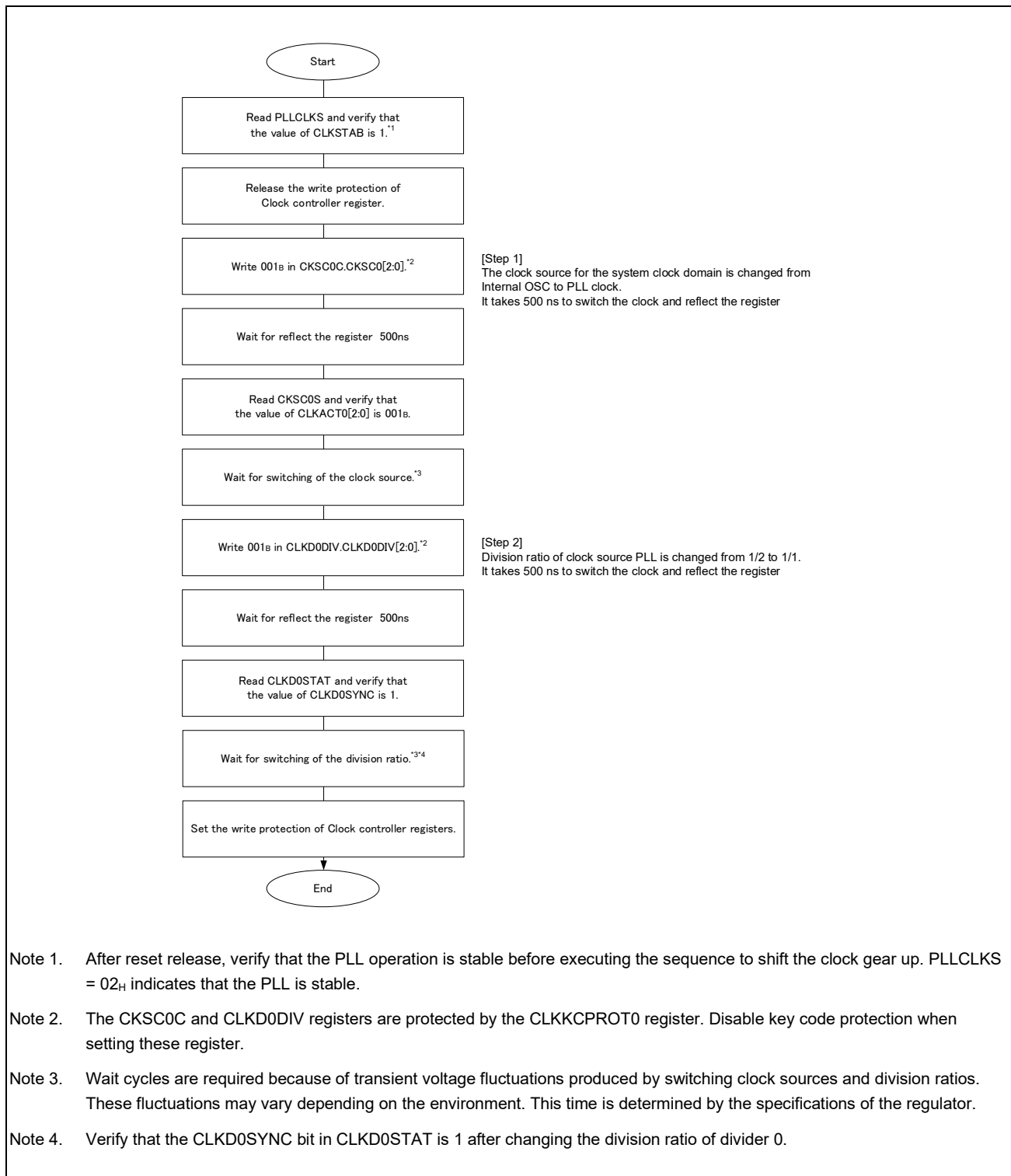


Figure 15.2 Example of Shifting Clock Gear Up

15.4.2 External Clock Output Pins (CK)

The device provides an external clock that can be used as clock supply for other external circuits. For details, see **Section 15.3.6, CLKDnDIV — Clock Divider Divisor Register (n = 2)**, **Section 15.3.7, CLKDnSTAT — Clock Divider Status Register (n = 2)**, **Section 15.3.10, CKSCnC — Clock Selector Control Register (n = 2)**, and **Section 15.3.11, CKSCnS — Clock Selector Status Register (n = 2)**.

- Clock signals can be output from the external pin CK.
- Output clock frequencies can be divided by the divider by configuring register settings.
- The source of the external clock can be configured as CLK_MOSC or CLK_LSB.
- It is recommended that the external clock be configured after clock gearup is complete.

NOTES

1. Stop the external clock before changing the clock source.
2. CLK_LSB becomes the backup clock if changing to a backup clock was triggered by an error detected in CLMA.

15.4.3 Main OSC Configuration and Stabilization Sequence

The block diagram of MainOSC setting and the sequence of MainOSC stabilization are shown in **Figure 15.3** and **Figure 15.4**, respectively.

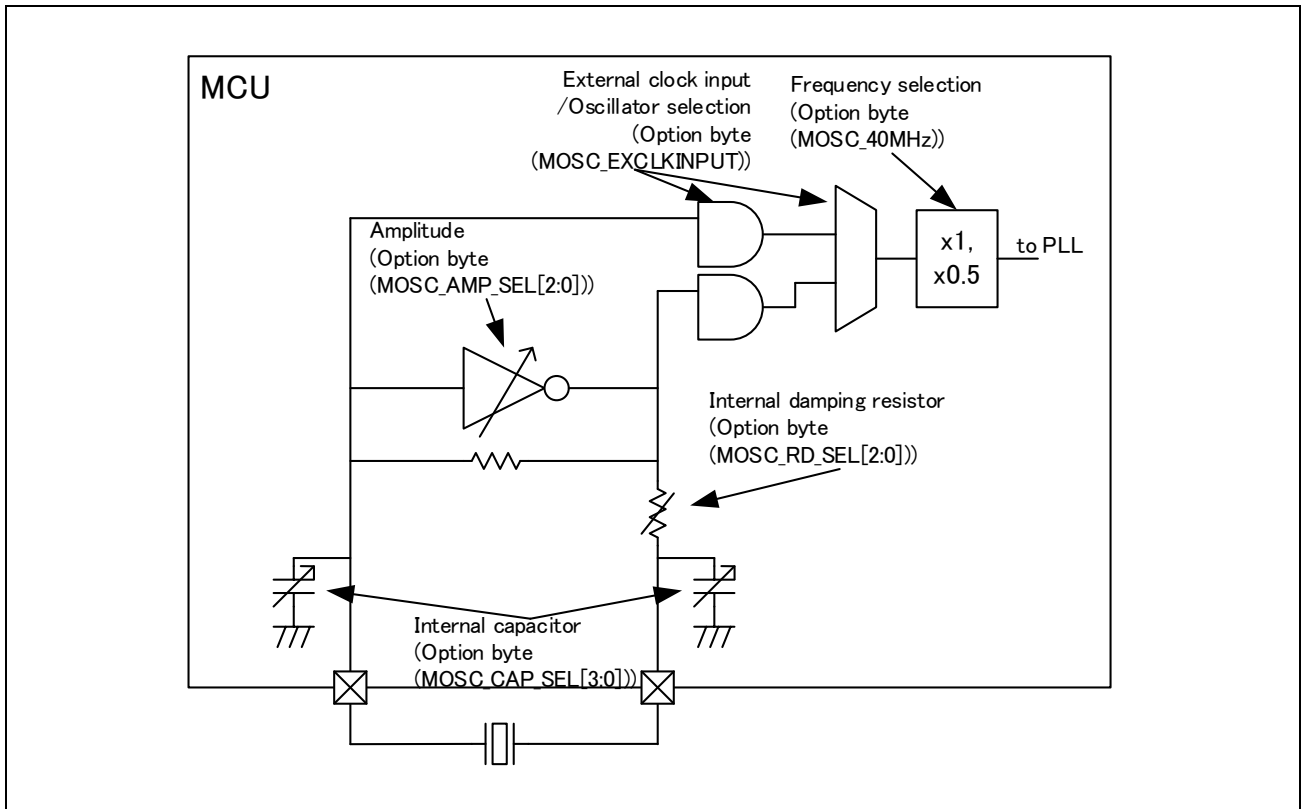


Figure 15.3 Block Diagram of MainOSC Configuration

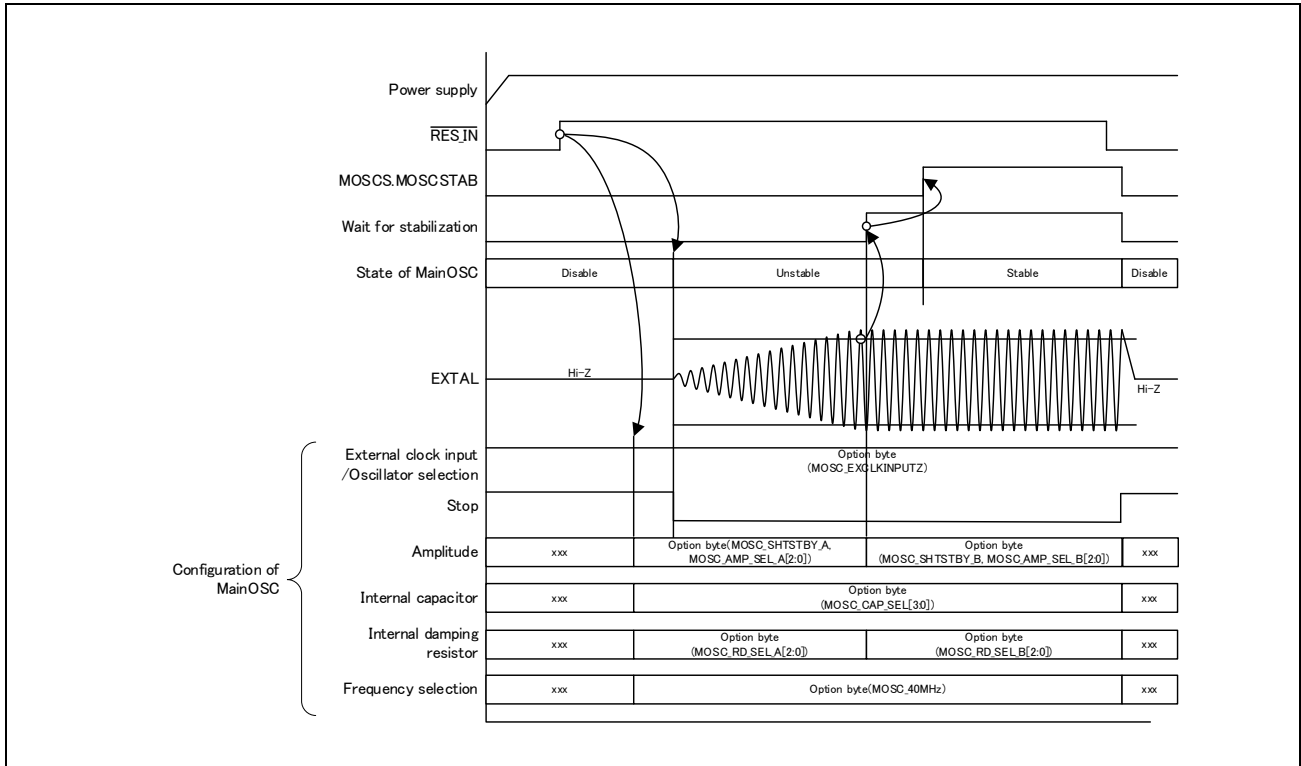


Figure 15.4 Sequence of MainOSC Stabilization

15.4.4 OSC Monitor Mode

This mode is used for characterization of the main oscillator crystal.

OSC monitor mode is enabled if the OSCMONMD.OSCMONMD bit is set to 1.

Steps required to prevent the system from stopping when in OSC monitor mode:

- Disable auto start of WDTB and SWDT.
- Wait six cycles (CLK_LSB) to transition to OSC monitor mode.
- Be sure to execute the HALT instruction in each CPU after transitioning to OSC monitor mode.
- Return from OSC monitor mode should be done by External Reset.

The recommended characterization flow is shown in **Figure 15.6**.

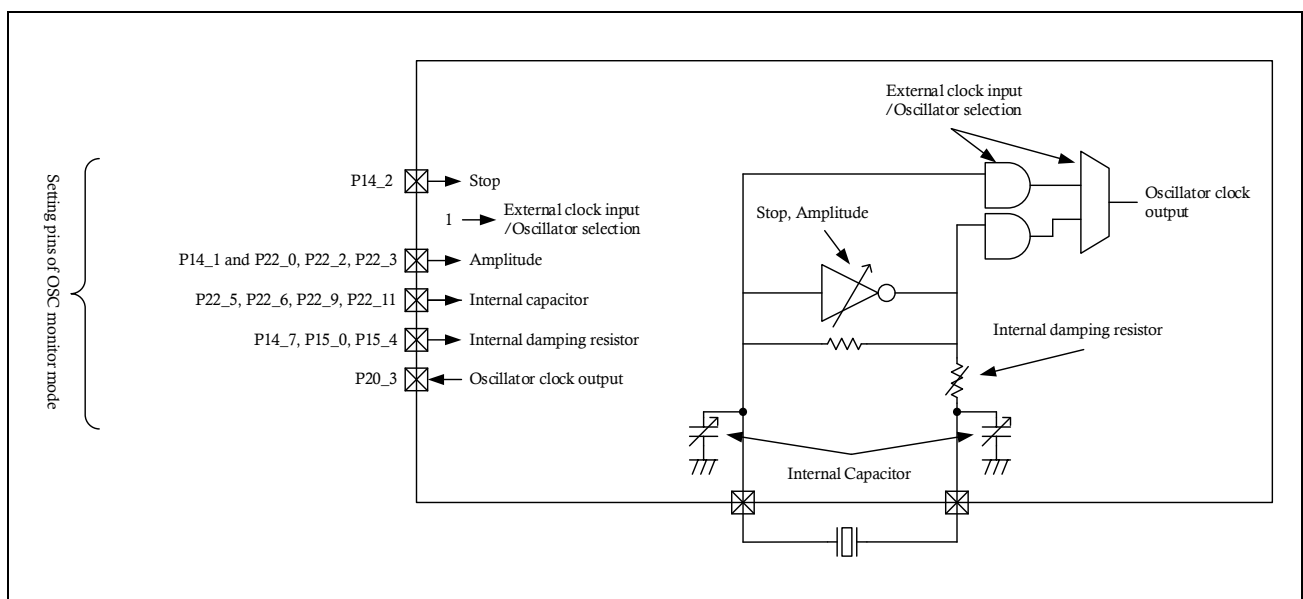


Figure 15.5 Block Diagram of OSC Monitor Mode

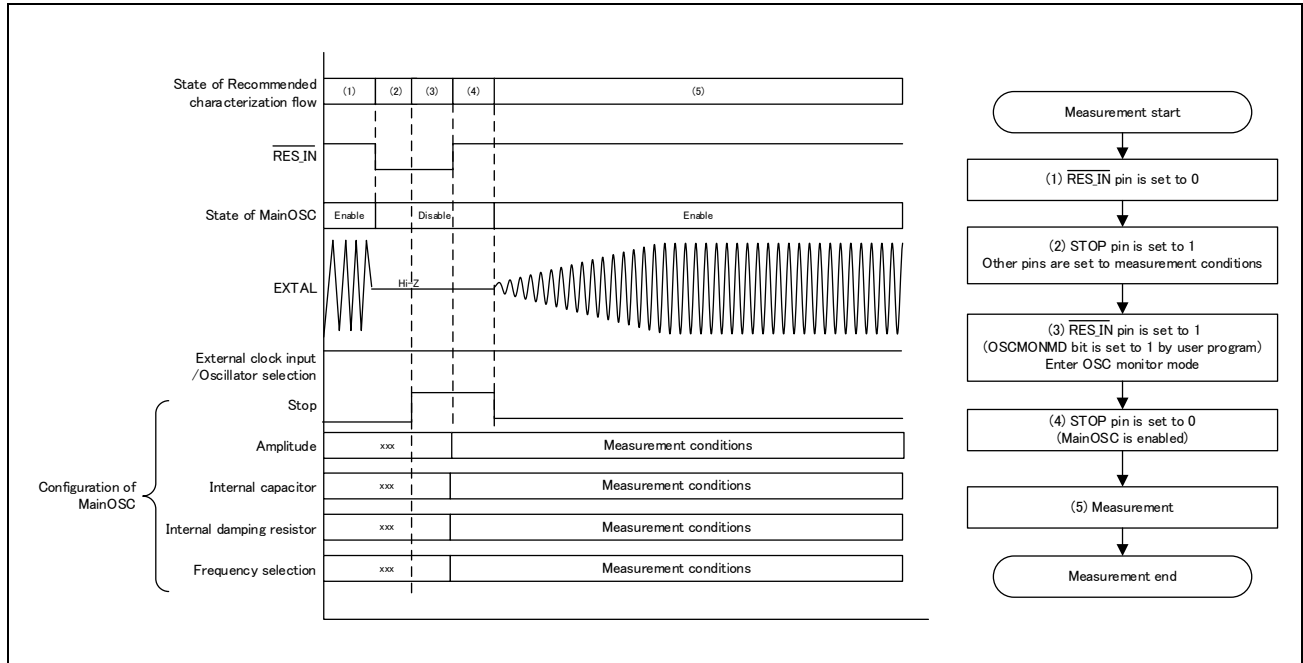


Figure 15.6 Recommended Characterization Flow

Table 15.22 OSC Monitor Mode Pins

Signal Name	Option Byte* ¹	OSC Monitor Mode		Description
		Setting Pins	I/O	
STOP	—	P14_2	Input	MOSC enable/disable 0: MOSC is enabled. 1: MOSC is stopped.
EXCLK	MOSC_EXCLKINPUTZ	Oscillator	Input	External clock input/Oscillator selection 0: MOSC outputs the clock from EXTAL as is, without going through AMP. (External clock) 1: Normal oscillation (Oscillator)
FREQ_SEL	MOSC_40MHz	—	Input	Frequency selection 0: 20 MHz 1: 40 MHz
CAPSEL[3:0]	MOSC_CAP_SEL[3:0]	P22_5, P22_6, P22_9, P22_11	Input	This bit controls the OSC internal capacitance.* ²
SHTSTBY_A	MOSC_SHTSTBY_A	P14_1	Input	Amplitude (x_A: unstable / x_B: stable) This bit controls OSC drivability during oscillation stabilization.* ²
SHTSTBY_B	MOSC_SHTSTBY_B			
AMPSEL_A[2:0]	MOSC_AMP_SEL_A[2:0]	P22_0, P22_2, P22_3	Input	
AMPSEL_B[2:0]	MOSC_AMP_SEL_B[2:0]			
RDSEL_A[2:0]	MOSC_RD_SEL_A[2:0]	P14_7, P15_0, P15_4	Input	Damping resistor configuration. (x_A: Unstable / x_B: Stable) This bit controls the damping resistor.* ²
RDSEL_B[2:0]	MOSC_RD_SEL_B[2:0]			
OSCOUT	—	P20_3	Output	MainOSC clock output

Note 1. For details on the option byte, see **Section 41, Flash Memory**.

Note 2. For details, see *Section 1.3.4, Clock Timing in the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

15.5 Usage Notes

15.5.1 Main OSC setting

In the initial state of the product to be shipped, Option Bytes are fixed.

In case of the mismatch between the EXTAL/XTAL parameter condition and Main OSC Option Bytes setting relationship, Option Bytes should be written in Serial programming mode to be matched against the EXTAL/XTAL parameter condition.

15.5.2 Clock Gear Up

The operational settings for each peripheral function should be configured after clock gearup is complete.

(Except for when each function has a specified usage.)

The operation of each peripheral function is guaranteed only in the clock gear up state.

(Except when permitted by each peripheral function)

It must be set according to the procedure in **Figure 15.2**.

It is recommended to set the external clock output after clock gearup because CLK_MOSC is not stable and CLK_LSB has a different frequency.

15.5.3 FPU Operation Notes

Operations that require the FPU or FPSIMD should not be used when the division ratio is being changed. Increased current fluctuation as a result of using the FPU or FPSIMD can make the division ratio change unstable.

15.5.4 Board Design Notes

Other signal lines should not cross over EXTAL and XTAL because induction may prevent proper oscillation. See **Figure 15.7**.

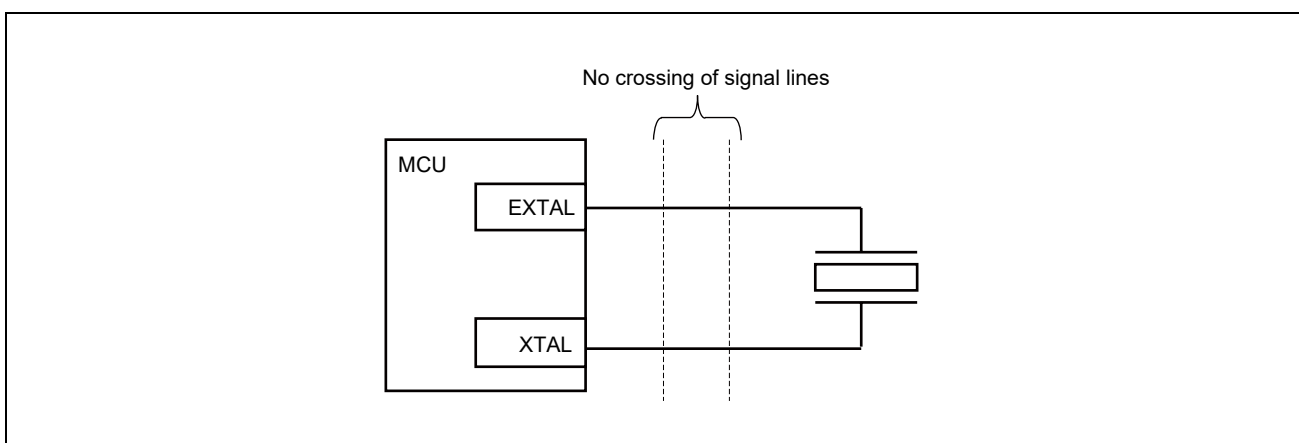


Figure 15.7 Board Design Notes

Section 16 Standby Controller

16.1 Features

This product supports various power-down modes such as power off standby mode, Emulation/Instrumentation RAM and Aurora retention mode (E2x-FCC1 Only), HALT mode, and module standby mode.

The power consumption of the LSI can be reduced by applying a suitable power-down mode for the application.

16.1.1 Register Base Addresses

Standby controller base address is listed in the following table.

Standby controller register addresses are given as offsets from the base addresses in general.

Table 16.1 Register Base Addresses

Base Address Name	Base Address	Bus Group
<MSTB_base>	FF70 1000 _H	Peripheral Group 6

16.2 Power Down Modes

16.2.1 Power Off Standby Mode

In Power Off Standby Mode, only the retention RAM of the Global (Cluster) RAM is powered. All clocks are stopped. All power supplies except SYSVCC can be turned off.

Power Off Standby Mode is entered by asserting the RES_IN pin low, and then turning off all power supplies except SYSVCC.

Power Off Standby Mode is exited by turning all the power supplies on, waiting for the required time for clock stabilization and then negating RES_IN. See *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware* for timing details.

If the EMUVDD is supplied, the data in Emulation/Instrumentation RAM is also retained. For RAM retention targets, see the section about RAM.

16.2.2 Emulation/Instrumentation RAM and Aurora Retention Mode (E2x-FCC1 Only)

In Emulation/Instrumentation RAM and Aurora Retention Mode, only the Emulation/Instrumentation RAM and Aurora configuration data is retained. All power supplies except EMUVCC and EMUVDD can be turned off.

Emulation/Instrumentation RAM and Aurora Retention Mode is entered by asserting the RES_IN pin low, and then turning off all power supplies except EMUVCC and EMUVDD.

Emulation/Instrumentation RAM and Aurora Retention Mode is exited by turning all the power supplies on, waiting for the required time for clock stabilization and then negating RES_IN.

See *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware* for timing details.

16.2.3 HALT Mode

When the HALT instruction is executed, the CPU transits to HALT mode and stops instruction execution.

Each CPU (PE0, PE1) can be controlled individually.

The CPU returns from this state by the occurrence of an all resets (except Module Reset and JTAG Reset), interrupt or exception.

16.2.4 Module Standby

This function stops the clocks for the selected peripherals to reduce power consumption. After reset (see **Table 16.2, Table 16.3**) is released, All the peripheral modules except for WDTB, $\Delta\Sigma$ -ADC and Cyclic-ADC enter the module standby mode.

Set the bit of the target register to 0 to release the module standby mode

Set the bit of the target register to 1 to set the module standby mode

NOTES

1. It is recommended to set WDTB module stanby before clock gear-up (if WDTB has not been auto-started or has not triggered by software yet)
2. These registers should only be written one time after reset. Operation is not guaranteed for subsequent writes.

16.3 Description of Registers

16.3.1 Register Protection

Write-protected registers are protected from inadvertent write access due to erroneous program execution, etc.

Write-protected registers can be written by releasing the protection of MSRKCPROT.

16.3.2 List of Registers

The following table lists the standby controller registers.

Table 16.2 List of Registers

Unit Name	Register Name	Symbol	Address	Access size	Access Protection
MSTB	Module Standby Register for ATU5	MSR_ATU* ¹	<MSTB_base> + 000 _H	32	MSRKCPROT
MSTB	Module Standby Register for GTM	MSR_GTM* ¹	<MSTB_base> + 000 _H	32	MSRKCPROT
MSTB	Module Standby Register for OSTM	MSR_OSTM	<MSTB_base> + 010 _H	32	MSRKCPROT
MSTB	Module Standby Register for WDTB	MSR_WDT	<MSTB_base> + 020 _H	32	MSRKCPROT
MSTB	Module Standby Register for SCI3	MSR_SCI	<MSTB_base> + 030 _H	32	MSRKCPROT
MSTB	Module Standby Register for HSSPI	MSR_HSSPI	<MSTB_base> + 040 _H	32	MSRKCPROT
MSTB	Module Standby Register for RLIN3	MSR_RLIN	<MSTB_base> + 050 _H	32	MSRKCPROT
MSTB	Module Standby Register for CSIH	MSR_CSIH	<MSTB_base> + 060 _H	32	MSRKCPROT
MSTB	Module Standby Register for RHSIF	MSR_RHSIF	<MSTB_base> + 070 _H	32	MSRKCPROT
MSTB	Module Standby Register for RHSB	MSR_RHSB	<MSTB_base> + 080 _H	32	MSRKCPROT
MSTB	Module Standby Register for CAN	MSR_CAN	<MSTB_base> + 090 _H	32	MSRKCPROT
MSTB	Module Standby Register for FlexRay	MSR_FLXA	<MSTB_base> + 0A0 _H	32	MSRKCPROT
MSTB	Module Standby Register for Ethernet controller	MSR_ETHER	<MSTB_base> + 0B0 _H	32	MSRKCPROT
MSTB	Module Standby Register for RSENT	MSR_RSENT	<MSTB_base> + 0C0 _H	32	MSRKCPROT
MSTB	Module Standby Register for PSI5S	MSR_PSI5S	<MSTB_base> + 0E0 _H	32	MSRKCPROT
MSTB	Module Standby Register for DFE	MSR_DFE	<MSTB_base> + 100 _H	32	MSRKCPROT
MSTB	Module Standby Register for DS-ADC	MSR_DAD	<MSTB_base> + 110 _H	32	MSRKCPROT
MSTB	Module Standby Register for Cyclic-ADC	MSR_CAD	<MSTB_base> + 120 _H	32	MSRKCPROT
MSTB	Module Standby Register for SAR-ADC	MSR_SAD	<MSTB_base> + 130 _H	32	MSRKCPROT
MSTB	Module Standby Register for ENCA	MSR_ENCA	<MSTB_base> + 140 _H	32	MSRKCPROT
MSTB	Module Standby Register Key Code Protection Register	MSRKCPROT	<MSTB_base> + 700 _H	32	—

Note 1. MSR_ATU, MSR_GTM switch in OPBT. See **Section 41, Flash Memory**.

16.3.3 Reset of Registers

Table 16.3 Register Reset Conditions

	Reset Source					
	Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
All registers	✓	✓	✓	✓	—	—

16.3.4 MSR_ATU — Module Standby Register for ATU5

This register is used to control the Stop Mode of the ATU5.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_ATU
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.4 MSR_ATU Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
0	MS_ATU	Setting this bit to 1 stops clock supply to ATU5. 0: Put ATU5 in operating mode. 1: Put ATU5 in standby mode (stop clocks to ATU5).

16.3.5 MSR_GTM — Module Standby Register for GTM

This register is used to control the Stop Mode of the GTM.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_GTM
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.5 MSR_GTM Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
0	MS_GTM	Setting this bit to 1 stops clock supply to GTM. 0: Put GTM in operating mode. 1: Put GTM in standby mode (stop clocks to GTM).

16.3.6 MSR_OSTM — Module Standby Register for OSTM

This register is used to control the Stop Mode of the OSTM.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_	MS_	MS_
														OSTM2	OSTM1	OSTM0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 16.6 MSR_OSTM Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
2	MS_OSTM2	Setting this bit to 1 stops clock supply to OSTM2. 0: Put OSTM2 in operating mode. 1: Put OSTM2 in standby mode (stop clocks to OSTM2).
1	MS_OSTM1	Setting this bit to 1 stops clock supply to OSTM1. 0: Put OSTM1 in operating mode. 1: Put OSTM1 in standby mode (stop clocks to OSTM1).
0	MS_OSTM0	Setting this bit to 1 stops clock supply to OSTM0. 0: Put OSTM0 in operating mode. 1: Put OSTM0 in standby mode (stop clocks to OSTM0).

16.3.7 MSR_WDT — Module Standby Register for WDTB

This register is used to control the Stop Mode of the WDTB.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_WDT1	MS_WDT0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 16.7 MSR_WDT Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
1	MS_WDT1	Setting this bit to 1 stops clock supply to WDTB1. 0: Put WDTB1 in operating mode. 1: Put WDTB1 in standby mode (stop clocks to WDTB1).
0	MS_WDT0	Setting this bit to 1 stops clock supply to WDTB0. 0: Put WDTB0 in operating mode. 1: Put WDTB0 in standby mode (stop clocks to WDTB0).

16.3.8 MSR_SCI — Module Standby Register for SCI3

This register is used to control the Stop Mode of the SCI3.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	MS_SCI33	MS_SCI32	MS_SCI31	MS_SCI30
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 16.8 MSR_SCI Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
3	MS_SCI33	Setting this bit to 1 stops clock supply to SCI33. 0: Put SCI3 in operating mode. 1: Put SCI3 in standby mode (stop clocks to SCI33).
2	MS_SCI32	Setting this bit to 1 stops clock supply to SCI32. 0: Put SCI2 in operating mode. 1: Put SCI2 in standby mode (stop clocks to SCI32).
1	MS_SCI31	Setting this bit to 1 stops clock supply to SCI31. 0: Put SCI1 in operating mode. 1: Put SCI1 in standby mode (stop clocks to SCI31).
0	MS_SCI30	Setting this bit to 1 stops clock supply to SCI30. 0: Put SCI0 in operating mode. 1: Put SCI0 in standby mode (stop clocks to SCI30).

16.3.9 MSR_HSSPI — Module Standby Register for HSSPI

This register is used to control the Stop Mode of the HSSPI.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_HSSPI0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.9 MSR_HSSPI Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
0	MS_HSSPI0	Setting this bit to 1 stops clock supply to HSSPI0. 0: Put HSSPI0 in operating mode. 1: Put HSSPI0 in standby mode (stop clocks to HSSPI0).

16.3.10 MSR_RLIN — Module Standby Register for RLIN3

This register is used to control the Stop Mode of the RLIN3.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	MS_RLIN5	MS_RLIN4	MS_RLIN3	MS_RLIN2	MS_RLIN1	MS_RLIN0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.10 MSR_RLIN Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
5	MS_RLIN5	Setting this bit to 1 stops clock supply to RLIN5. 0: Put RLIN5 in operating mode. 1: Put RLIN5 in standby mode (stop clocks to RLIN5).
4	MS_RLIN4	Setting this bit to 1 stops clock supply to RLIN4. 0: Put RLIN4 in operating mode. 1: Put RLIN4 in standby mode (stop clocks to RLIN4).
3	MS_RLIN3	Setting this bit to 1 stops clock supply to RLIN3. 0: Put RLIN3 in operating mode. 1: Put RLIN3 in standby mode (stop clocks to RLIN3).
2	MS_RLIN2	Setting this bit to 1 stops clock supply to RLIN2. 0: Put RLIN2 in operating mode. 1: Put RLIN2 in standby mode (stop clocks to RLIN2).
1	MS_RLIN1	Setting this bit to 1 stops clock supply to RLIN1. 0: Put RLIN1 in operating mode. 1: Put RLIN1 in standby mode (stop clocks to RLIN1).
0	MS_RLIN0	Setting this bit to 1 stops clock supply to RLIN0. 0: Put RLIN0 in operating mode. 1: Put RLIN0 in standby mode (stop clocks to RLIN0).

16.3.11 MSR_CSIH — Module Standby Register for CSIH

This register is used to control the Stop Mode of the CSIH.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	MS_CSIH5	MS_CSIH4	MS_CSIH3	MS_CSIH2	MS_CSIH1	MS_CSIH0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.11 MSR_CSIH Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
5	MS_CSIH5	Setting this bit to 1 stops clock supply to CSIH5. 0: Put CSIH5 in operating mode. 1: Put CSIH5 in standby mode (stop clocks to CSIH5).
4	MS_CSIH4	Setting this bit to 1 stops clock supply to CSIH4. 0: Put CSIH4 in operating mode. 1: Put CSIH4 in standby mode (stop clocks to CSIH4).
3	MS_CSIH3	Setting this bit to 1 stops clock supply to CSIH3. 0: Put CSIH3 in operating mode. 1: Put CSIH3 in standby mode (stop clocks to CSIH3).
2	MS_CSIH2	Setting this bit to 1 stops clock supply to CSIH2. 0: Put CSIH2 in operating mode. 1: Put CSIH2 in standby mode (stop clocks to CSIH2).
1	MS_CSIH1	Setting this bit to 1 stops clock supply to CSIH1. 0: Put CSIH1 in operating mode. 1: Put CSIH1 in standby mode (stop clocks to CSIH1).
0	MS_CSIH0	Setting this bit to 1 stops clock supply to CSIH0. 0: Put CSIH0 in operating mode. 1: Put CSIH0 in standby mode (stop clocks to CSIH0).

16.3.12 MSR_RHSIF — Module Standby Register for RHSIF

This register is used to control the Stop Mode of the RHSIF.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_RHSIF0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.12 MSR_RHSIF Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
0	MS_RHSIF0*1.	Setting this bit to 1 stops clock supply to RHSIF0. 0: Put RHSIF0 in operating mode. 1: Put RHSIF0 in standby mode (stop clocks to RHSIF0).

Note 1. Access to RHSIF is required waiting period of at least 12 cycles in the CPU clock after standby release.

16.3.13 MSR_RHSB — Module Standby Register for RHSB

This register is used to control the Stop Mode of the RHSB.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_RHSB1	MS_RHSB0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 16.13 MSR_RHSB Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
1	MS_RHSB1	Setting this bit to 1 stops clock supply to RHSB1. 0: Put RHSB1 in operating mode. 1: Put RHSB1 in standby mode (stop clocks to RHSB1).
0	MS_RHSB0	Setting this bit to 1 stops clock supply to RHSB0. 0: Put RHSB0 in operating mode. 1: Put RHSB0 in standby mode (stop clocks to RHSB0).

16.3.14 MSR_CAN — Module Standby Register for CAN

This register is used to control the Stop Mode of the CAN.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_RSCFD0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.14 MSR_CAN Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
0	MS_RSCFD0	Setting this bit to 1 stops clock supply to RSCFD0. 0: Put RSCFD0 in operating mode. 1: Put RSCFD0 in standby mode (stop clocks to RSCFD0).

16.3.15 MSR_FLXA — Module Standby Register for FlexRay

This register is used to control the Stop Mode of the FlexRay.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_FLXA0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.15 MSR_FLXA Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
0	MS_FLXA0	Setting this bit to 1 stops clock supply to FLXA0. 0: Put FLXA0 in operating mode. 1: Put FLXA0 in standby mode (stop clocks to FLXA0).

16.3.16 MSR_ETHER — Module Standby Register for Ethernet controller

This register is used to control the Stop Mode of the Ethernet controller.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_ETHER
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.16 MSR_ETHER Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
0	MS_ETHER	Setting this bit to 1 stops clock supply to Ethernet controller. 0: Put ETHER in operating mode. 1: Put ETHER in standby mode (stop clocks to Ethernet controller).

16.3.17 MSR_RSENT — Module Standby Register for RSENT

This register is used to control the Stop Mode of the RSENT.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_RS ENT16
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MS_RS ENT15	MS_RS ENT14	MS_RS ENT13	MS_RS ENT12	MS_RS ENT11	MS_RS ENT10	MS_RS ENT9	MS_RS ENT8	MS_RS ENT7	MS_RS ENT6	MS_RS ENT5	MS_RS ENT4	MS_RS ENT3	MS_RS ENT2	MS_RS ENT1	MS_RS ENT0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.17 MSR_RSENT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 17	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
16	MS_RSENT16	Setting this bit to 1 stops clock supply to RSENT16. 0: Put RSENT16 in operating mode. 1: Put RSENT16 in standby mode (stop clocks to RSENT16).
15	MS_RSENT15	Setting this bit to 1 stops clock supply to RSENT15. 0: Put RSENT15 in operating mode. 1: Put RSENT15 in standby mode (stop clocks to RSENT15).
14	MS_RSENT14	Setting this bit to 1 stops clock supply to RSENT14. 0: Put RSENT14 in operating mode. 1: Put RSENT14 in standby mode (stop clocks to RSENT14).
13	MS_RSENT13	Setting this bit to 1 stops clock supply to RSENT13. 0: Put RSENT13 in operating mode. 1: Put RSENT13 in standby mode (stop clocks to RSENT13).
12	MS_RSENT12	Setting this bit to 1 stops clock supply to RSENT12. 0: Put RSENT12 in operating mode. 1: Put RSENT12 in standby mode (stop clocks to RSENT12).
11	MS_RSENT11	Setting this bit to 1 stops clock supply to RSENT11. 0: Put RSENT11 in operating mode. 1: Put RSENT11 in standby mode (stop clocks to RSENT11).

Table 16.17 MSR_RSENT Register Contents (2/2)

Bit Position	Bit Name	Function
10	MS_RSENT10	Setting this bit to 1 stops clock supply to RSENT10. 0: Put RSENT10 in operating mode. 1: Put RSENT10 in standby mode (stop clocks to RSENT10).
9	MS_RSENT9	Setting this bit to 1 stops clock supply to RSENT9. 0: Put RSENT9 in operating mode. 1: Put RSENT9 in standby mode (stop clocks to RSENT9).
8	MS_RSENT8	Setting this bit to 1 stops clock supply to RSENT8. 0: Put RSENT8 in operating mode. 1: Put RSENT8 in standby mode (stop clocks to RSENT8).
7	MS_RSENT7	Setting this bit to 1 stops clock supply to RSENT7. 0: Put RSENT7 in operating mode. 1: Put RSENT7 in standby mode (stop clocks to RSENT7).
6	MS_RSENT6	Setting this bit to 1 stops clock supply to RSENT6. 0: Put RSENT6 in operating mode. 1: Put RSENT6 in standby mode (stop clocks to RSENT6).
5	MS_RSENT5	Setting this bit to 1 stops clock supply to RSENT5. 0: Put RSENT5 in operating mode. 1: Put RSENT5 in standby mode (stop clocks to RSENT5).
4	MS_RSENT4	Setting this bit to 1 stops clock supply to RSENT4. 0: Put RSENT4 in operating mode. 1: Put RSENT4 in standby mode (stop clocks to RSENT4).
3	MS_RSENT3	Setting this bit to 1 stops clock supply to RSENT3. 0: Put RSENT3 in operating mode. 1: Put RSENT3 in standby mode (stop clocks to RSENT3).
2	MS_RSENT2	Setting this bit to 1 stops clock supply to RSENT2. 0: Put RSENT2 in operating mode. 1: Put RSENT2 in standby mode (stop clocks to RSENT2).
1	MS_RSENT1	Setting this bit to 1 stops clock supply to RSENT1. 0: Put RSENT1 in operating mode. 1: Put RSENT1 in standby mode (stop clocks to RSENT1).
0	MS_RSENT0	Setting this bit to 1 stops clock supply to RSENT0. 0: Put RSENT0 in operating mode. 1: Put RSENT0 in standby mode (stop clocks to RSENT0).

16.3.18 MSR_PSI5S — Module Standby Register for PSI5S

This register is used to control the Stop Mode of the PSI5S.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_PSI5S
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.18 MSR_PSI5S Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
0	MS_PSI5S	Setting this bit to 1 stops clock supply to PSI5S. 0: Put PSI5S in operating mode. 1: Put PSI5S in standby mode (stop clocks to PSI5S).

16.3.19 MSR_DFE — Module Standby Register for DFE

This register is used to control the Stop Mode of the DFE.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_DFE
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.19 MSR_DFE Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
0	MS_DFE	Setting this bit to 1 stops clock supply to DFE. 0: Put DFE in operating mode. 1: Put DFE in standby mode (stop clocks to DFE).

16.3.20 MSR_DAD — Module Standby Register for DS-ADC

This register is used to control the Stop Mode of the $\Delta\Sigma$ -ADC.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_DAD 13_11	MS_DAD 20_12	MS_DAD 00_10
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 16.20 MSR_DAD Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
2	MS_DAD13_11	Setting this bit to 1 stops clock supply to units DSADC13 and DSADC11. 0: Put units DSADC13 and DSADC11 of $\Delta\Sigma$ AD in operating mode. 1: Put units DSADC13 and DSADC11 of $\Delta\Sigma$ AD in standby mode (stop clocks to units DSADC13 and DSADC11 of $\Delta\Sigma$ AD).
1	MS_DAD20_12	Setting this bit to 1 stops clock supply to units DSADC20 and DSADC12 0: Put units DSADC20 and DSADC12 of $\Delta\Sigma$ AD in operating mode. 1: Put units DSADC20 and DSADC12 of $\Delta\Sigma$ AD in standby mode (stop clocks to units DSADC20 and DSADC12 of $\Delta\Sigma$ AD).
0	MS_DAD00_10	Setting this bit to 1 stops clock supply to units DSADC00 and DSADC10. 0: Put units DSADC00 and DSADC10 of $\Delta\Sigma$ AD in operating mode. 1: Put units DSADC00 and DSADC10 of $\Delta\Sigma$ AD in standby mode (stop clocks to units DSADC00 and DSADC10 of $\Delta\Sigma$ AD).

16.3.21 MSR_CAD — Module Standby Register for Cyclic-ADC

This register is used to control the Stop Mode of the Cyclic-ADC.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_CAD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.21 MSR_CAD Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
0	MS_CAD	Setting this bit to 1 stops clock supply to Cyclic-ADC. 0: Put Cyclic-ADC in operating mode. 1: Put Cyclic-ADC in standby mode (stop clocks to Cyclic-ADC).

16.3.22 MSR_SAD — Module Standby Register for SAR-ADC

This register is used to control the Stop Mode of the SAR-ADC.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	MS_SAD3	MS_SAD2	MS_SAD1	MS_SAD0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 16.22 MSR_SAD Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
3	MS_SAD3	Setting this bit to 1 stops clock supply to ADCH3. 0: Put ADCH3 in operating mode. 1: Put ADCH3 in standby mode (stop clocks to ADCH3).
2	MS_SAD2	Setting this bit to 1 stops clock supply to ADCH2. 0: Put ADCH2 in operating mode. 1: Put ADCH2 in standby mode (stop clocks to ADCH2).
1	MS_SAD1	Setting this bit to 1 stops clock supply to ADCH1. 0: Put ADCH1 in operating mode. 1: Put ADCH1 in standby mode (stop clocks to ADCH1).
0	MS_SAD0	Setting this bit to 1 stops clock supply to ADCH0. 0: Put ADCH0 in operating mode. 1: Put ADCH0 in standby mode (stop clocks to ADCH0).

16.3.23 MSR_ENCA — Module Standby Register for ENCA

This register is used to control the Stop Mode of the ENCA.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_ENCA1	MS_ENCA0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 16.23 MSR_ENCA Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When reading, the value after reset is returned. When writing, write the value after reset.
1	MS_ENCA1	Setting this bit to 1 stops clock supply to ENCA1. 0: Put ENCA1 in operating mode. 1: Put ENCA1 in standby mode (stop clocks to ENCA1).
0	MS_ENCA0	Setting this bit to 1 stops clock supply to ENCA0. 0: Put ENCA0 in operating mode. 1: Put ENCA0 in standby mode (stop clocks to ENCA0).

16.3.24 MSRKCPROT — Module Standby Register Key Code Protection Register

The MSRKCPROT register is used to protect against writing to the registers that may have a material effect on the system so that the application system is not incorrectly stopped due to program malfunction and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 16.24 MSRKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

Section 17 Clocked Serial Interface H (CSIH)

17.1 Features of CSIH

17.1.1 Units and Channels

This microcontroller has the following number of CSIH channels.

Table 17.1 Number of Channels (For E2x-FCC1)

Product Name	RH850/E2x-FCC1	
	373 pins	292 pins
Number of Channels	6	6
Unit Name	CSIHn (n = 0 to 5)	

Table 17.2 Number of Channels (For E2M)

Product Name	RH850/E2M	
	373 pins	292 pins
Number of Channels	6	6
Unit Name	CSIHn (n = 0 to 5)	

Table 17.3 Index

Index	Description
m	A variable used for explanation is identified by the index "m": for example, CSIHnETXDAm and CSIHnERXDAm are a non-specified baud rate setting register of CSIHn.
n	Throughout this section, the individual CSIH units are identified by the index "n" ("n" is channel numbers.): for example, CSIHnCTL0 is the CSIHn control register 0.
x	CSIHn has a maximum of 6 chip select signals. Throughout this section, the individual chip select signals are identified by the index "x": that is, CSx denotes a non-specified chip select signal.
y	A variable used for explanation is identified by the index "y": for example, CSIHnBRSy is the baud rate setting register of CSIHn.

The numbers of chip select signals for each of the CSIH units are listed in the following table.

Table 17.4 Number of Chip Select Signals

Unit Name	Chip Select Index
CSIH0	CSx (x = 0 to 5)
CSIH1	CSx (x = 0 to 3)
CSIH2	CSx (x = 0 to 3)
CSIH3	CSx (x = 0 to 3)
CSIH4	CSx (x = 0 to 3)
CSIH5	CSx (x = 0 to 3)

17.1.2 Register Base Address

CSIH base addresses are listed in the following table.

CSIH register addresses are given as offsets from the base addresses.

Table 17.5 Register Base Addresses

Base Address Name	Base Address	Bus Group
<CSIH0_base>	FFD8 0000 _H	Peripheral Group 4
<CSIH1_base>	FFD8 2000 _H	Peripheral Group 4
<CSIH2_base>	FFD8 4000 _H	Peripheral Group 4
<CSIH3_base>	FFD8 6000 _H	Peripheral Group 4
<CSIH4_base>	FFD8 8000 _H	Peripheral Group 3
<CSIH5_base>	FFD8 A000 _H	Peripheral Group 3

17.1.3 Clock Supply

The CSIH clock supply is shown in the following table.

Table 17.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
CSIHn	PCLK	CLK_HSB

17.1.4 Interrupt Requests

CSIH interrupt requests are listed in the following table.

Table 17.7 Interrupt Requests

Interrupt symbol name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
CSIH0					
INTCSIH0TIC	INTCSIHTIC	Communication status interrupt	343	group0 93	group0 93
INTCSIH0TIR	INTCSIHTIR	Receive status interrupt	344	group0 92	group0 92
INTCSIH0TIRE	INTCSIHTIRE	Communication error interrupt	345	—	—
INTCSIH0TIJC	INTCSIHTIJC	Job completion interrupt	346	group0 94	group0 94
CSIH1					
INTCSIH1TIC	INTCSIHTIC	Communication status interrupt	347	group0 96	group0 96
INTCSIH1TIR	INTCSIHTIR	Receive status interrupt	348	group0 95	group0 95
INTCSIH1TIRE	INTCSIHTIRE	Communication error interrupt	349	—	—
INTCSIH1TIJC	INTCSIHTIJC	Job completion interrupt	350	group0 97	group0 97
CSIH2					
INTCSIH2TIC	INTCSIHTIC	Communication status interrupt	351	group0 99	group0 99
INTCSIH2TIR	INTCSIHTIR	Receive status interrupt	352	group0 98	group0 98
INTCSIH2TIRE	INTCSIHTIRE	Communication error interrupt	353	—	—
INTCSIH2TIJC	INTCSIHTIJC	Job completion interrupt	354	group0 100	group0 100
CSIH3					
INTCSIH3TIC	INTCSIHTIC	Communication status interrupt	355	group0 102	group0 102
INTCSIH3TIR	INTCSIHTIR	Receive status interrupt	356	group0 101	group0 101
INTCSIH3TIRE	INTCSIHTIRE	Communication error interrupt	357	—	—
INTCSIH3TIJC	INTCSIHTIJC	Job completion interrupt	358	group0 103	group0 103
CSIH4					
INTCSIH4TIC	INTCSIHTIC	Communication status interrupt	359	group0 105	group0 105
INTCSIH4TIR	INTCSIHTIR	Receive status interrupt	360	group0 104	group0 104
INTCSIH4TIRE	INTCSIHTIRE	Communication error interrupt	361	—	—
INTCSIH4TIJC	INTCSIHTIJC	Job completion interrupt	362	group0 106	group0 106
CSIH5					
INTCSIH5TIC	INTCSIHTIC	Communication status interrupt	363	group0 108	group0 108
INTCSIH5TIR	INTCSIHTIR	Receive status interrupt	364	group0 107	group0 107
INTCSIH5TIRE	INTCSIHTIRE	Communication error interrupt	365	—	—
INTCSIH5TIJC	INTCSIHTIJC	Job completion interrupt	366	group0 109	group0 109

17.1.5 Reset Sources

CSIH reset sources are listed in the following table. CSIH is initialized by these reset sources.

Table 17.8 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
CSIHn	All registers	✓	✓	✓	✓	—	—

17.1.6 External Input/Output Signals

External input/output signals of CSIH are listed below.

Table 17.9 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
CSIH0		
CSIH0TCK	Serial clock signal	CSIH0TCK
CSIH0TSI	Serial data input signal	CSIH0TSI
$\overline{\text{CSIH0TSSI}}$	Slave select input signal	$\overline{\text{CSIH0TSSI}}$
CSIH0TRYI	Ready/busy input signal	CSIH0TRY
CSIH0TSO	Serial data output signal	CSIH0TSO
CSIH0TRYO	Ready/busy output signal	CSIH0TRY
CSIH0TCSS[5:0]*1	Chip select signal	CSIH0TCSS[5:0]*1
CSIH1		
CSIH1TCK	Serial clock signal	CSIH1TCK
CSIH1TSI	Serial data input signal	CSIH1TSI
$\overline{\text{CSIH1TSSI}}$	Slave select input signal	$\overline{\text{CSIH1TSSI}}$
CSIH1TRYI	Ready/busy input signal	CSIH1TRY
CSIH1TSO	Serial data output signal	CSIH1TSO
CSIH1TRYO	Ready/busy output signal	CSIH1TRY
CSIH1TCSS[3:0]*1	Chip select signal	CSIH1TCSS[3:0]*1
CSIH2		
CSIH2TCK	Serial clock signal	CSIH2TCK
CSIH2TSI	Serial data input signal	CSIH2TSI
$\overline{\text{CSIH2TSSI}}$	Slave select input signal	$\overline{\text{CSIH2TSSI}}$
CSIH2TRYI	Ready/busy input signal	CSIH2TRY
CSIH2TSO	Serial data output signal	CSIH2TSO
CSIH2TRYO	Ready/busy output signal	CSIH2TRY
CSIH2TCSS[3:0]*1	Chip select signal	CSIH2TCSS[3:0]*1
CSIH3		
CSIH3TCK	Serial clock signal	CSIH3TCK
CSIH3TSI	Serial data input signal	CSIH3TSI
$\overline{\text{CSIH3TSSI}}$	Slave select input signal	$\overline{\text{CSIH3TSSI}}$
CSIH3TRYI	Ready/busy input signal	CSIH3TRY
CSIH3TSO	Serial data output signal	CSIH3TSO
CSIH3TRYO	Ready/busy output signal	CSIH3TRY
CSIH3TCSS[3:0]*1	Chip select signal	CSIH3TCSS[3:0]*1
CSIH4		
CSIH4TCK	Serial clock signal	CSIH4TCK
CSIH4TSI	Serial data input signal	CSIH4TSI
$\overline{\text{CSIH4TSSI}}$	Slave select input signal	$\overline{\text{CSIH4TSSI}}$
CSIH4TRYI	Ready/busy input signal	CSIH4TRY
CSIH4TSO	Serial data output signal	CSIH4TSO
CSIH4TRYO	Ready/busy output signal	CSIH4TRY
CSIH4TCSS[3:0]*1	Chip select signal	CSIH4TCSS[3:0]*1

Table 17.9 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
CSIH5		
CSIHTSCK	Serial clock signal	CSIH5TSCK
CSIHTSI	Serial data input signal	CSIH5TSI
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH5TSSI}}$
CSIHTRYI	Ready/busy input signal	CSIH5TRY
CSIHTSO	Serial data output signal	CSIH5TSO
CSIHTRYO	Ready/busy output signal	CSIH5TRY
CSIHTCSS[3:0]*1	Chip select signal	CSIH5TCSS[3:0]*1

Note 1. For the number of chip select signals, see **Table 17.4**.

CAUTION

This AC specification is only applied to the specific pin groups. For details, refer to Appendix file "Limited_conditions_for_AC_specification.xlsx".

17.1.7 Data Consistency Check

The ports and the alternative functions for data consistency check of CSIHnSO (CSIHTSO) output are shown in the following table. See **Section 17.5.11, Error Detection** for details on data consistency checking.

Table 17.10 Port Pins for Data Consistency Checking

Unit Signal Name	Port Pin Name	Alternative Function
CSIH0		
CSIHTSO	P11_9	ALT_OUT1
	P20_6	ALT_OUT1
	P20_7	ALT_OUT1
	P22_4	ALT_OUT1
	P22_7	ALT_OUT1
	P22_8	ALT_OUT1
CSIH1		
CSIHTSO	P10_1	ALT_OUT1
	P10_2	ALT_OUT1
	P10_3	ALT_OUT1
	P10_6	ALT_OUT6
	P11_7	ALT_OUT2
	P12_5	ALT_OUT6
	P12_6	ALT_OUT1
	P12_7	ALT_OUT1
CSIH2		
CSIHTSO	P14_1	ALT_OUT4
	P15_3	ALT_OUT1
	P15_5	ALT_OUT1
	P15_7	ALT_OUT1
	P15_8	ALT_OUT1
	P34_3	ALT_OUT1
	P34_4	ALT_OUT1
CSIH3		
CSIHTSO	P01_5	ALT_OUT1
	P01_6	ALT_OUT1
	P02_5	ALT_OUT1
	P02_6	ALT_OUT1
	P10_6	ALT_OUT1
	P10_8	ALT_OUT1
CSIH4		
CSIHTSO	P22_2	ALT_OUT1
	P22_3	ALT_OUT1
	P33_10	ALT_OUT1
	P33_12	ALT_OUT1
CSIH5		
CSIHTSO	P14_10	ALT_OUT1
	P14_11	ALT_OUT1
	P23_2	ALT_OUT1
	P23_6	ALT_OUT1

17.1.8 Notes on Pin Combinations

CSIH uses multiple pins for one channel and the pins should be used in groups as shown in the following table.

Table 17.11 Pin combinations of CSIH

	Group	CSIHn TSCK	CSIHn TSI	CSIHn TSO	CSIHn TSSI	CSIHn TRY	CSIHn TCSS0	CSIHn TCSS1	CSIHn TCSS2	CSIHn TCSS3	CSIHn TCSS4	CSIHn TCSS5
CSIH0	1	P11_5	P11_0 P11_4 P12_7	P11_9	P11_7	P22_10	P11_7	P11_10	P11_8	P11_6 P12_8	P11_3 P12_9	P11_4 P11_1 P12_5
	2	P20_3 P20_5	P20_4 P20_6 P20_7	P20_6 P20_7	P20_5 P20_4	P22_10	P20_2 P20_3 P20_7	P20_4	P20_5	—	—	—
	3	P22_6 P22_9	P22_4 P22_7 P22_8	P22_4 P22_7 P22_8	—	P22_10	—	—	—	—	—	—
CSIH1	1	P10_4	P10_1 P10_2 P10_3	P10_1 P10_2 P10_3 P10_6	P10_1	P10_0	P10_1	P10_5	P10_6	P10_8	—	—
	2	P11_0 P11_2	P11_4	P11_7	P11_1 P11_10	P10_0	P11_10	P11_2	P11_4	P12_8	—	—
	3	P12_9	P12_6 P12_7 P12_8	P12_5 P12_6 P12_7	P12_8	P10_0	P12_5	P11_2	P11_4	P12_8	—	—
CSIH2	1	P14_0 P14_1 P15_4 P15_6	P14_2 P15_2 P15_4 P15_5 P15_7 P15_8	P14_1 P15_3 P15_5 P15_7 P15_8	P14_3 P15_0 P15_1 P15_2	P14_6 P15_0	P14_2 P15_2	P14_7 P14_11	P14_5 P15_0 P15_8	P14_3 P14_12	—	—
	2	P32_2 P33_9	P34_3 P34_4	P34_3 P34_4	—	P14_6 P15_0	—	—	—	—	—	—
CSIH3	1	P01_7 P02_9	P01_5 P01_6 P02_5 P02_6	P01_5 P01_6 P02_5 P02_6	P01_4 P02_3	P01_3	P02_3	P02_0	P02_2	P00_5 P02_1	—	—
	2	P10_7	P10_6 P10_8	P10_6 P10_8	P02_3	—	P02_3	—	—	—	—	—
CSIH4	1	P22_1	P22_2 P22_3	P22_2 P22_3	P22_0	P32_1 P33_8	P22_5	—	P22_7	P22_0	—	—
	2	P33_6	P33_10 P33_12	P33_10 P33_12	P33_11	P32_1 P33_8	P33_11	P33_4	P33_7	P33_8	—	—
CSIH5	1	P14_12	P14_10 P14_11	P14_10 P14_11	—	P23_0	—	—	—	P15_5	—	—
	2	P23_1 P23_7	P23_2 P23_6	P23_2 P23_6	P23_3	P23_0	P23_5	P23_4	P23_3	P23_7	—	—

17.2 Overview

17.2.1 Functional Overview

- Three-wire serial synchronous data transfer well known as SPI
- Either master mode or slave mode can be selected
- Multiple slaves configuration and RCB (Recessive Configuration for Broadcasting) are possible since there are six configurable chip select output signals
- Slave select input signal ($\overline{\text{CSIHTSSI}}$) is usable
- Built-in baud rate generator
- Transfer clock frequency is adjustable in master mode, whereas it is determined by the input clock in slave mode.
- Maximum transfer clock frequency:
 - Master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/8)
 - Slave mode: 4.0 MHz (however, it must be equal to or lower than PCLK/20)
- Clock phase and data phase are selectable
- Data transfer with MSB first or LSB first is selectable
- Transfer data length is selectable from 2 to 16 bits in 1-bit units
- Built-in EDL (extended data length) function for transferring more than 16 bits of data
- Three selectable transfer modes:
 - transmit-only mode
 - receive-only mode
 - transmit/receive mode
- Built-in handshake function
- Built-in error detection (data consistency check, parity, and overrun)
- Support of job concept
- Four different interrupt request signals (INTCSIHTIC, INTCSIHTIR, INTCSIHTIRE, INTCSIHTIJC)
- Three different DMA request signals (INTCSIHTIC, INTCSIHTIR, INTCSIHTIJC)
- Built-in LBM (loop back mode) function for self-test
- Enforced chip select idle setting
- Built-in JOB enable control bit for AUTOSAR
- CSIH has an extended communication function that can perform 8, 16, 32-bit SPI communication.
 - Extended communication data registers can use up to 16 stages.

17.2.2 Functional Overview Description

The CSIH uses three signals for communication:

- Transmission clock CSIH_TSCK (output in master mode, input in slave mode)
- Data output signal CSIH_TSO
- Data input signal CSIH_TSI

Additional signals are available for external control and monitoring:

- $\overline{\text{CSIH}}\text{TSSI}$: Slave select input signal
- CSIH_TRYO: Ready/busy output signal (handshake signal)
- CSIH_TRYI: Ready/busy input signal (handshake signal)
- CSIH_TCSS[5:0]: Chip select signals

Data transmission is bit-wise and serial, and performed synchronously with the transmission clock.

The following table shows the most important registers for setting up the CSIH.

Table 17.12 Main Registers of CSIH

Register	Function
CSIH _n CTL0	Enables/disables serial clock, and permits/prohibits data transmission and data reception. Defines end-of-job behavior and enables/disables (bypass) buffering.
CSIH _n CTL1	Controls options like interrupt timing, extended data length, job feature, loop-back mode, handshake, etc. such as
CSIH _n CTL2	Selects master or slave mode, and the transfer clock frequency of the built-in baud rate generator (BRG) in master mode.
CSIH _n BRSy	Specifies the transfer clock frequency for each chip select signal.
CSIH _n CFGx	Configures the communication protocol for each chip select signal.

17.2.3 Block Diagram

The following block diagram shows the main components of the CSIH.

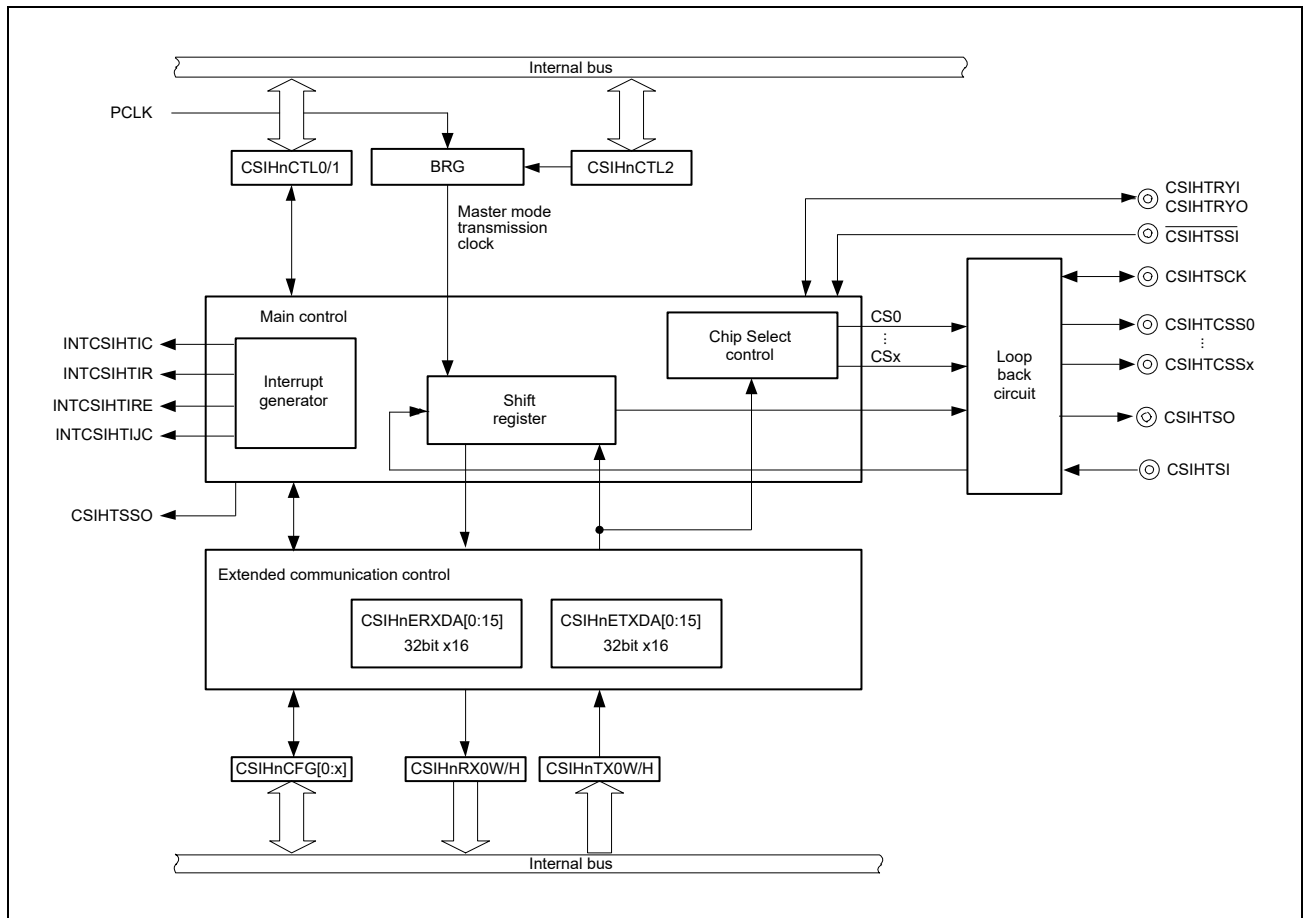


Figure 17.1 CSIH Block Diagram

In master mode, the transmission clock CSIHnTSSCK is generated by the built-in baud rate generator (BRG). In slave mode, the transmission clock is supplied by an external source.

The loop back circuit disconnects the CSIH completely from the ports and supports internal self-test.

NOTE

This section describes the following modes:

- The “operating mode” is either master mode or and slave mode. In this context, only a master can control and communicate with several slaves (for details, see **Section 17.5.1, Operating Modes (Master/Slave)**).
- The “job mode” is related to the AUTOSAR job concept (for details, see **Section 17.5.3.3, Job Concept**).
- The “data transfer mode” specifies the type of communication – transmit-only, receive-only, or transmit/receive (for details, see **Section 17.5.6, Data Transfer Modes**).

17.3 Registers

17.3.1 List of Registers

CSIH registers are listed in the following table.

For details about <CSIHn_base>, see **Section 17.1.2, Register Base Address**.

Table 17.13 List of Registers (1/3)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection
CSIHn	CSIHn Control Register 0	CSIHnCTL0	<CSIHn_base> + 0000 _H	8	—
	CSIHn Control Register 1	CSIHnCTL1	<CSIHn_base> + 0010 _H	32	—
	CSIHn Control Register 2	CSIHnCTL2	<CSIHn_base> + 0014 _H	16	—
	CSIHn Status Register 0	CSIHnSTR0	<CSIHn_base> + 0004 _H	32	—
	CSIHn Status Clear Register 0	CSIHnSTCR0	<CSIHn_base> + 0008 _H	16	—
	CSIHn Configuration Register 0	CSIHnCFG0	<CSIHn_base> + 1044 _H	32	—
	CSIHn Configuration Register 1	CSIHnCFG1	<CSIHn_base> + 1048 _H	32	—
	CSIHn Configuration Register 2	CSIHnCFG2	<CSIHn_base> + 104C _H	32	—
	CSIHn Configuration Register 3	CSIHnCFG3	<CSIHn_base> + 1050 _H	32	—
	CSIHn Configuration Register 4	CSIHnCFG4	<CSIHn_base> + 1054 _H	32	—
	CSIHn Configuration Register 5	CSIHnCFG5	<CSIHn_base> + 1058 _H	32	—
	CSIHn Transmit Data Register 0 for Word Access	CSIHnTX0W	<CSIHn_base> + 1008 _H	32	—
	CSIHn Transmit Data Register 0 for Half Word Access	CSIHnTX0H	<CSIHn_base> + 100C _H	16	—
	CSIHn Receive Data Register 0 for Word Access	CSIHnRX0W	<CSIHn_base> + 1010 _H	32	—
	CSIHn Receive Data Register 0 for Half Word Access	CSIHnRX0H	<CSIHn_base> + 1014 _H	16	—
	CSIHn Baud Rate Setting Register 0	CSIHnBRS0	<CSIHn_base> + 1068 _H	16	—
	CSIHn Baud Rate Setting Register 1	CSIHnBRS1	<CSIHn_base> + 106C _H	16	—
	CSIHn Baud Rate Setting Register 2	CSIHnBRS2	<CSIHn_base> + 1070 _H	16	—
	CSIHn Baud Rate Setting Register 3	CSIHnBRS3	<CSIHn_base> + 1074 _H	16	—
	CSIHn Interrupt Control Register	CSIHnEIE	<CSIHn_base> + 1100 _H	8,16,32	—
	CSIHn Extended Communication Control Register 0	CSIHnECTL0	<CSIHn_base> + 1104 _H	8,16,32	—
	CSIHn Extended Communication Control Register 1	CSIHnECTL1	<CSIHn_base> + 1108 _H	8,16,32	—
	CSIHn Extended Communication Transmit Data Register 0	CSIHnETXDA0	<CSIHn_base> + 110C _H	8,16,32	—
	CSIHn Extended Communication Transmit Data Register 1	CSIHnETXDA1	<CSIHn_base> + 1110 _H	8,16,32	—
	CSIHn Extended Communication Transmit Data Register 2	CSIHnETXDA2	<CSIHn_base> + 1114 _H	8,16,32	—
	CSIHn Extended Communication Transmit Data Register 3	CSIHnETXDA3	<CSIHn_base> + 1118 _H	8,16,32	—
	CSIHn Extended Communication Transmit Data Register 4	CSIHnETXDA4	<CSIHn_base> + 111C _H	8,16,32	—
	CSIHn Extended Communication Transmit Data Register 5	CSIHnETXDA5	<CSIHn_base> + 1120 _H	8,16,32	—
	CSIHn Extended Communication Transmit Data Register 6	CSIHnETXDA6	<CSIHn_base> + 1124 _H	8,16,32	—

Table 17.13 List of Registers (2/3)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection
CSIHn	CSIHn Extended Communication Transmit Data Register 7	CSIHnETXDA7	<CSIHn_base> + 1128 _H	8,16,32	—
	CSIHn Extended Communication Transmit Data Register 8	CSIHnETXDA8	<CSIHn_base> + 112C _H	8,16,32	—
	CSIHn Extended Communication Transmit Data Register 9	CSIHnETXDA9	<CSIHn_base> + 1130 _H	8,16,32	—
	CSIHn Extended Communication Transmit Data Register 10	CSIHnETXDA10	<CSIHn_base> + 1134 _H	8,16,32	—
	CSIHn Extended Communication Transmit Data Register 11	CSIHnETXDA11	<CSIHn_base> + 1138 _H	8,16,32	—
	CSIHn Extended Communication Transmit Data Register 12	CSIHnETXDA12	<CSIHn_base> + 113C _H	8,16,32	—
	CSIHn Extended Communication Transmit Data Register 13	CSIHnETXDA13	<CSIHn_base> + 1140 _H	8,16,32	—
	CSIHn Extended Communication Transmit Data Register 14	CSIHnETXDA14	<CSIHn_base> + 1144 _H	8,16,32	—
	CSIHn Extended Communication Transmit Data Register 15	CSIHnETXDA15	<CSIHn_base> + 1148 _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 0	CSIHnERXDA0	<CSIHn_base> + 1150 _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 1	CSIHnERXDA1	<CSIHn_base> + 1154 _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 2	CSIHnERXDA2	<CSIHn_base> + 1158 _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 3	CSIHnERXDA3	<CSIHn_base> + 115C _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 4	CSIHnERXDA4	<CSIHn_base> + 1160 _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 5	CSIHnERXDA5	<CSIHn_base> + 1164 _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 6	CSIHnERXDA6	<CSIHn_base> + 1168 _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 7	CSIHnERXDA7	<CSIHn_base> + 116C _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 8	CSIHnERXDA8	<CSIHn_base> + 1170 _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 9	CSIHnERXDA9	<CSIHn_base> + 1174 _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 10	CSIHnERXDA10	<CSIHn_base> + 1178 _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 11	CSIHnERXDA11	<CSIHn_base> + 117C _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 12	CSIHnERXDA12	<CSIHn_base> + 1180 _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 13	CSIHnERXDA13	<CSIHn_base> + 1184 _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 14	CSIHnERXDA14	<CSIHn_base> + 1188 _H	8,16,32	—
	CSIHn Extended Communication Receive Data Register 15	CSIHnERXDA15	<CSIHn_base> + 118C _H	8,16,32	—
	CSIHn Extended Communication Stop Trigger Register	CSIHnESTT	<CSIHn_base> + 1190 _H	8,16,32	—

Table 17.13 List of Registers (3/3)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection
CSIHn	CSIHn Extended Communication Status Register	CSIHnESTF	<CSIHn_base> + 1194 _H	8,16,32	—

17.3.2 CSIHnCTL0 — CSIHn Control Register 0

This register controls the operation clock, enables/disables transmission/reception, and enables/disables the memory allocated for transmission and/or reception. It forces the stop of communication at the end of the current job.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CSIHnPWR	CSIHnTXE	CSIHnRXE	—	—	—	CSIHnJOBE	CSIHnMBS
Value after reset	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R	R	R	R/W	R/W

Table 17.14 CSIHnCTL0 Register Contents

Bit Position	Bit Name	Function
7	CSIHnPWR	Controls the operation clock. 0: Stops operation clock. 1: Supplies operation clock. Clearing CSIHnPWR to 0 resets the internal circuits, stops operation, and sets CSIH to standby state. Clock supply to internal circuits stops. If CSIHnPWR is cleared to 0 during communication, ongoing communication is immediately aborted. In this case, the communication setting must be reconfigured.
6	CSIHnTXE	Permits or prohibits transmission. 0: Prohibits transmission. 1: Permits transmission.
5	CSIHnRXE	Permits or prohibits reception. 0: Prohibits reception. 1: Permits reception.
4 to 2	—	Reserved When read, the value after reset is read. When writing to these bits, write the value after reset.
1	CSIHnJOBE	Stops communication at the end of the current job (communication ends if data is written to the transmission buffer when CSIHnTX0W.CSIHnEOJ = 1 (job completion)). 0: Communication stop is not requested. 1: Stops communication. This bit can be used to abort an ongoing job. This bit is cleared to 0 automatically. Even if this bit is set to 1, the read value is always 0.
0	CSIHnMBS	Enables direct access mode. 0: CSIH transmit/receive operation is prohibited. 1: Direct access mode Write 1 in CSIHnMBS bit for transmission/reception via the CSIH.

CAUTIONS

- Do not modify any of CSIHnTXE, CSIHnRXE, CSIHnJOBE, and CSIHnMBS bits while CSIHnPWR = 0.
Do not modify the CSIHnMBS bit when CSIHnPWR = 1.
The CSIHnTXE, CSIHnRXE, and CSIHnMBS bits can be modified when the CSIHnPWR bit is changed from 0 to 1.
To change the CSIHnMBS bit, do so at the same time as when the CSIHnPWR bit is changed from 0 to 1.
- Do not modify CSIHnTXE, CSIHnRXE, or CSIHnMBS while data transmission is pending or going on, i.e. if CSIHnSTR0.CSIHnTSF = 1.

When setting this register, see **Table 17.35, Notes on Setting Registers**.

17.3.3 CSIHnCTL1 — CSIHn Control Register 1

This register specifies the interrupt timing and the interrupt delay mode. It also enables/disables extended data length control, loop-back mode, handshake functionality, and job mode. It selects the active output level of each chip select signal and the behavior of the chip select signals after the transfer of the final data.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CSIHnCKR	CSIHnSLIT	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	CSIHnCSL5	CSIHnCSL4	CSIHnCSL3	CSIHnCSL2	CSIHnCSL1	CSIHnCSL0	CSIHnEDLE	CSIHnJE	CSIHnD	CSIHnCS	CSIHnCSRI	CSIHnLBM	CSIHnSIT	CSIHnHSE	CSIHnSSE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.15 CSIHnCTL1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 18	—	Reserved When read, the value after reset is read. When writing to these bits, write the value after reset.
17	CSIHnCKR	CSIHnTSCCK Clock Inversion Function 0: The default level of CSIHnTSCCK is high 1: The default level of CSIHnTSCCK is low For details, see Section 17.3.7, CSIHnCFGx — CSIHn Configuration Register x . CAUTION: Clear the CSIHnCKPx bit in CSIHnCFGx to 0 when this bit is used without the chip select function.
16	CSIHnSLIT	Selects the timing of interrupt INTCSIHnTIC. 0: Normal interrupt timing (interrupt is generated after the transfer) 1: As soon as the contents of the CSIHnTX0W/H register are transferred to the shift register, an interrupt is generated. For details, see Section 17.4.3, INTCSIHnTIC (Communication Status Interrupt) .
15 to 14	—	Reserved When read, the value after reset is read. When writing to these bits, write the value after reset.
13 to 8	CSIHnCSLx	Selects the active output level of chip select signal x (CSIHnTSCSSx). 0: Chip select is active low. 1: Chip select is active high. For details, see Section 17.5.3, Chip Selection (CS) Features .
7	CSIHnEDLE	Enables/disables extended data length (EDL) mode. 0: Disables extended data length mode. 1: Enables extended data length mode. For details, see Section 17.5.7.2, Data Length Greater than 16 Bits .

Table 17.15 CSIHnCTL1 Register Contents (2/2)

Bit Position	Bit Name	Function
6	CSIHnJE	<p>Enables/disables job mode.</p> <p>0: Disables job mode. 1: Enables job mode.</p> <p>For details, see Section 17.5.3.3, Job Concept.</p> <p>The CSIHnCTL0.CSIHnJOB_E, and CSIHnTX0W.CSIHnEOJ bits are enabled only when CSIHnJE = 1.</p> <p>Setting this bit in slave mode is prohibited.</p> <p>When CSIHnJE = 0, interrupt INTCSIHTIC is generated after one unit of data transmission is completed.</p> <p>When CSIHnJE = 1, interrupt INTCSIHTIC is also generated after one unit of data transmission is completed.</p> <p>In the case of CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOB_E = 1, however, interrupt INTCSIHTIJC is generated instead of INTCSIHTIC.</p>
5	CSIHnDCS	<p>Enables/disables data consistency check.</p> <p>0: Disables data consistency check. 1: Enables data consistency check.</p> <p>For details, see Section 17.5.11.1, Data Consistency Check.</p>
4	CSIHnCSRI	<p>Defines chip select signal behavior after last data transfer.</p> <p>0: Chip select signal retains the active level. 1: Chip select signal returns to the inactive level.</p> <p>The last data is determined at the interrupt timing when CSIHnCTL1.CSIHnSLIT is 1.</p>
3	CSIHnLBM	<p>Controls loop-back mode (LBM).</p> <p>0: Deactivates loop-back mode. 1: Activates loop-back mode.</p> <p>For details, see Section 17.5.12, Loop-Back Mode.</p>
2	CSIHnSIT	<p>Selects interrupt delay mode.</p> <p>0: No delay is generated. 1: Half clock delay is generated for all interrupts.</p> <p>This bit is only valid in master mode. In slave mode, no delay is generated.</p> <p>For details, see Section 17.4.2, Interrupt Delay.</p>
1	CSIHnHSE	<p>Enables/disables the handshake function.</p> <p>0: Disables the handshake function. 1: Enables the handshake function.</p> <p>For details, see Section 17.5.10, Handshake Function.</p>
0	CSIHnSSE	<p>Enables/disables the slave select function.</p> <p>0: Input signal $\overline{\text{CSIHTSSI}}$ is disabled. 1: Input signal $\overline{\text{CSIHTSSI}}$ is recognized.</p> <p>If the slave select function is not used, this bit must be set to 0 (see also Section 17.5.2, Master/Slave Connections).</p>

Details about CSIHnCTL1.CSIHnSSE are shown in the following tables.

Table 17.16 Operation of the Slave Select Function during Reception

CSIHnCTL0. CSIHnRXE	CSIHnCTL1. CSIHnSSE	$\overline{\text{CSIHnTSSI}}$	Receive Operation
0	—	—	Reception is prohibited
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

Table 17.17 Operation of the Slave Select Function during Transmission

CSIHnCTL0. CSIHnTXE	CSIHnCTL1. CSIHnSSE	$\overline{\text{CSIHnTSSI}}$	Transmit Operation
0	—	—	Transmission is prohibited
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

CAUTION

When setting this register, see **Table 17.35, Notes on Setting Registers**.

17.3.4 CSIHnCTL2 — CSIHn Control Register 2

This register selects operating mode and the reference clock value, and specifies the transfer clock frequency.

For details, see **Section 17.5.5, Transmission Clock Selection**.

Value after reset: E000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnPRS[2:0]			—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 17.18 CSIHnCTL2 Register Contents

Bit Position	Bit Name	Function																																				
15 to 13	CSIHnPRS[2:0]	These bits select the operation mode and the reference clock value. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>CSIHnPRS2</th> <th>CSIHnPRS1</th> <th>CSIHnPRS0</th> <th>Selection of Reference Clock (PRSOUT)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>PCLK (Master mode)</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>PCLK/2 (Master mode)</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>PCLK/4 (Master mode)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>PCLK/8 (Master mode)</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>PCLK/16 (Master mode)</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>PCLK/32 (Master mode)</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>PCLK/64 (Master mode)</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>External clock via CSIHnTSCk(in) (Slave mode)</td> </tr> </tbody> </table>	CSIHnPRS2	CSIHnPRS1	CSIHnPRS0	Selection of Reference Clock (PRSOUT)	0	0	0	PCLK (Master mode)	0	0	1	PCLK/2 (Master mode)	0	1	0	PCLK/4 (Master mode)	0	1	1	PCLK/8 (Master mode)	1	0	0	PCLK/16 (Master mode)	1	0	1	PCLK/32 (Master mode)	1	1	0	PCLK/64 (Master mode)	1	1	1	External clock via CSIHnTSCk(in) (Slave mode)
CSIHnPRS2	CSIHnPRS1	CSIHnPRS0	Selection of Reference Clock (PRSOUT)																																			
0	0	0	PCLK (Master mode)																																			
0	0	1	PCLK/2 (Master mode)																																			
0	1	0	PCLK/4 (Master mode)																																			
0	1	1	PCLK/8 (Master mode)																																			
1	0	0	PCLK/16 (Master mode)																																			
1	0	1	PCLK/32 (Master mode)																																			
1	1	0	PCLK/64 (Master mode)																																			
1	1	1	External clock via CSIHnTSCk(in) (Slave mode)																																			
12 to 0	—	Reserved When read, the value after reset is read. When writing to these bits, write the value after reset.																																				

CAUTION: Max baud rate configuration is as follows.

- Master mode is PCLK/8.
- Slave mode is PCLK/20.

In master mode, the following bits are used to set the transfer clock frequency:

CSIHnCTL2.CSIHnPRS[2:0], CSIHnCFGx.CSIHnBRSSx[1:0], CSIHnBRSy.CSIHnBRS[11:0]

In addition, any of the four different transfer clock frequency settings that are specified by the CSIHnBRSy.CSIHnBRS[11:0] bits are selected for each chip select signal. To select the transfer clock frequency setting for each chip select signal, use the CSIHnCFGx.CSIHnBRSSx[1:0] bits.

The following table shows the relationship between CSIHnCFGx.CSIHnBRSSx[1:0] and CSIHnBRSy.CSIHnBRS[11:0].

CSIHnCFGx.CSIHnBRSSx[1:0]	Transfer Clock Frequency Setting Bit to be Selected
00	CSIHnBRS0.CSIHnBRS[11:0]
01	CSIHnBRS1.CSIHnBRS[11:0]
10	CSIHnBRS2.CSIHnBRS[11:0]
11	CSIHnBRS3.CSIHnBRS[11:0]

The following table shows the relationship between the transfer clock frequency and the transfer clock frequency setting (CSIHnBRSy[11:0]) selected by the CSIHnBRSSx[1:0] bits when the bit value of the CSIHnPRS[2:0] bits is α .

CSIHnBRSy[11:0]	Transfer clock frequency
0	BRG stopped
1	$PCLK / (2^\alpha \times 1 \times 2)$
2	$PCLK / (2^\alpha \times 2 \times 2)$
3	$PCLK / (2^\alpha \times 3 \times 2)$
4	$PCLK / (2^\alpha \times 4 \times 2)$
...	...
4095	$PCLK / (2^\alpha \times 4095 \times 2)$

CAUTION

Changing the contents of this register is only permitted when CSIHnCTL0.CSIHnPWR = 0.

When setting this register, see **Table 17.35, Notes on Setting Registers**.

17.3.5 CSIHnSTR0 — CSIHn Status Register 0

This register indicates the status of CSIH.

Value after reset: 0000 0010_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CSIHn TSF	—	—	—	CSIHn DCE	—	CSIHn PE	CSIHn OVE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 17.19 CSIHnSTR0 Register Contents (1/2)

Bit Position	Bit Name	Function																		
31 to 8	—	Reserved When read, the value after reset is read. When writing to these bits, write the value after reset.																		
7	CSIHnTSF	Transfer Status Flag 0: Idle state 1: Communication is in progress or being prepared. The timing to set or clear this bit is as follows: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Master Mode</th> <th>Timing to Set</th> <th>Timing to Clear</th> </tr> </thead> <tbody> <tr> <td>Transmit-only mode</td> <td rowspan="3">Data is written to a transmit register (CSIHnTX0W/CSIHnTX0H)</td> <td rowspan="3">Within a half clock cycle of the last serial clock edge</td> </tr> <tr> <td>Transmit/receive mode</td> </tr> <tr> <td>Receive-only mode</td> </tr> <tr> <th>Slave Mode</th> <th>Timing to Set</th> <th>Timing to Clear</th> </tr> <tr> <td>Transmit-only mode</td> <td rowspan="2">Data is written to a transmit register (CSIHnTX0W/CSIHnTX0H)</td> <td rowspan="2">Within a half clock cycle of the last serial clock edge</td> </tr> <tr> <td>Transmit/receive mode</td> </tr> <tr> <td>Receive-only mode</td> <td>Input timing of CSIHnSCK</td> <td></td> </tr> </tbody> </table>	Master Mode	Timing to Set	Timing to Clear	Transmit-only mode	Data is written to a transmit register (CSIHnTX0W/CSIHnTX0H)	Within a half clock cycle of the last serial clock edge	Transmit/receive mode	Receive-only mode	Slave Mode	Timing to Set	Timing to Clear	Transmit-only mode	Data is written to a transmit register (CSIHnTX0W/CSIHnTX0H)	Within a half clock cycle of the last serial clock edge	Transmit/receive mode	Receive-only mode	Input timing of CSIHnSCK	
Master Mode	Timing to Set	Timing to Clear																		
Transmit-only mode	Data is written to a transmit register (CSIHnTX0W/CSIHnTX0H)	Within a half clock cycle of the last serial clock edge																		
Transmit/receive mode																				
Receive-only mode																				
Slave Mode	Timing to Set	Timing to Clear																		
Transmit-only mode	Data is written to a transmit register (CSIHnTX0W/CSIHnTX0H)	Within a half clock cycle of the last serial clock edge																		
Transmit/receive mode																				
Receive-only mode	Input timing of CSIHnSCK																			
6 to 4	—	Reserved When read, the value after reset is read.																		
3	CSIHnDCE	Data Consistency Check Error Flag 0: No data consistency error is detected. 1: Data consistency error is detected. This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnDCEC. When this bit is set to 1 and cleared to 0 simultaneously, setting to 1 is prioritized. This bit is initialized when CSIHnCTL0.CSIHnPWR is changed from 0 to 1 or from 1 to 0.																		
2	—	Reserved When read, the value after reset is read.																		
1	CSIHnPE	Parity Error Flag 0: No parity error is detected. 1: Parity error is detected. This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnPEC. When this bit is set to 1 and cleared to 0 simultaneously, setting to 1 is prioritized. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.																		

Table 17.19 CSIHnSTR0 Register Contents (2/2)

Bit Position	Bit Name	Function
0	CSIHnOVE	Overrun Error Flag 0: No overrun error is detected. 1: Overrun error is detected. For details, see Section 17.5.11.2 , Parity Check. This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnOVEC. When this bit is set to 1 and cleared to 0 simultaneously, setting to 1 is prioritized. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.

Table 17.20 Behaviors in Various Direct Access Modes

Bit Name	Bit Position	Direct Access Mode
CSIHnTSF	7	0: Idle state 1: Communication is in progress or being prepared
CSIHnDCE	3	0: No error is detected. 1: An error is detected.
CSIHnPE	1	0: No error is detected. 1: An error is detected.
CSIHnOVE	0	0: No error is detected. 1: An error is detected.

CAUTION

When setting this register, see **Table 17.35, Notes on Setting Registers**.

17.3.6 CSIHnSTCR0 — CSIHn Status Clear Register 0

This register clears the status flags of the CSIHnSTR0 status register.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CSIHn DCEC	—	CSIHn PEC	CSIHn OVEC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W

Table 17.21 CSIHnSTCR0 Register Contents

Bit Position	Bit Name	Function
15 to 4	—	Reserved When read, the value after reset is read. When writing to these bits, write the value after reset.
3	CSIHnDCEC	Controls the data consistency error flag clear command. 0: No operation. The read value is always 0. 1: Clears the data consistency error flag (CSIHnSTR0.CSIHnDCE).
2	—	Reserved When read, the value after reset is read. When writing to these bits, write the value after reset.
1	CSIHnPEC	Controls the parity error flag clear command. 0: No operation. The read value is always 0. 1: Clears the parity error flag (CSIHnSTR0.CSIHnPE).
0	CSIHnOVEC	Controls the overrun error flag clear command. 0: No operation. The read value is always 0. 1: Clears the overrun error flag (CSIHnSTR0.CSIHnOVE).

CAUTION

When setting this register, see **Table 17.35, Notes on Setting Registers**.

17.3.7 CSIHnCFGx — CSIHn Configuration Register x (x = 0 to 5)

These eight registers configure the prescaler, parity, data length, recessive configuration for broadcasting, serial data direction, clock phase and data phase, idle enforcement configuration, idle time, hold time, inter-data time, and setup for each chip select signal, CSIHnCFGx.

Slave mode

In slave mode, the transmission protocol setting of the CSIHnCFG0 register is effective.

- CSIHnPS0[1:0]: parity usage
- CSIHnDLS0[3:0]: data length selection
- CSIHnDIR0: data direction
- CSIHnCKP0, CSIHnDAP0: clock phase and data phase

In slave mode, set all bits other than above in the CSIHnCFG0 register, and the CSIHnCFG1 to CSIHnCFG5 registers to 0.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHnBRSSx [1:0]		CSIHnPSx[1:0]		CSIHnDLSx[3:0]				—	—	—	—	CSIHn RCBx	CSIHn DIRx	CSIHn CKPx	CSIHn DAPx
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHn IDLx	CSIHnIDx[2:0]			CSIHnHDx[3:0]				CSIHnINx[3:0]				CSIHnSPx[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CAUTION

Rewrite is possible only when CSIHnCTL0.CSIHnPWR = 0 (for write access with the same value, rewrite is possible when CSIHnCTL0.CSIHnPWR = 1).

Table 17.22 CSIHnCFGx Register Contents (1/4)

Bit Position	Bit Name	Function																				
31, 30	CSIHnBRSSx[1:0]	These bits select the baud rate setting register (CSIHnBRSy).																				
		<table border="1"> <thead> <tr> <th>CSIHnBRSSx1</th> <th>CSIHnBRSSx0</th> <th>Baud Rate Setting Register Selection</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>The transfer clock frequency is set according to the CSIHnBRS0 setting.</td> </tr> <tr> <td>0</td> <td>1</td> <td>The transfer clock frequency is set according to the CSIHnBRS1 setting.</td> </tr> <tr> <td>1</td> <td>0</td> <td>The transfer clock frequency is set according to the CSIHnBRS2 setting.</td> </tr> <tr> <td>1</td> <td>1</td> <td>The transfer clock frequency is set according to the CSIHnBRS3 setting.</td> </tr> </tbody> </table>	CSIHnBRSSx1	CSIHnBRSSx0	Baud Rate Setting Register Selection	0	0	The transfer clock frequency is set according to the CSIHnBRS0 setting.	0	1	The transfer clock frequency is set according to the CSIHnBRS1 setting.	1	0	The transfer clock frequency is set according to the CSIHnBRS2 setting.	1	1	The transfer clock frequency is set according to the CSIHnBRS3 setting.					
		CSIHnBRSSx1	CSIHnBRSSx0	Baud Rate Setting Register Selection																		
		0	0	The transfer clock frequency is set according to the CSIHnBRS0 setting.																		
		0	1	The transfer clock frequency is set according to the CSIHnBRS1 setting.																		
		1	0	The transfer clock frequency is set according to the CSIHnBRS2 setting.																		
		1	1	The transfer clock frequency is set according to the CSIHnBRS3 setting.																		
		The maximum value for setting the transfer clock frequency, in accordance with the CSIHnCTL2.CSIHnPRS[2:0] setting, must be as follows:																				
		Master mode: PCLK/8																				
		Slave mode: PCLK/20																				
29, 28	CSIHnPSx[1:0]	Selects the parity for transmitting or receiving chip select signal x.																				
		<table border="1"> <thead> <tr> <th>CSIHnPSx1</th> <th>CSIHnPSx0</th> <th>Transmission</th> <th>Reception</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Does not transmit any parity bit.</td> <td>Does not wait for reception of the parity bit.</td> </tr> <tr> <td>0</td> <td>1</td> <td>Adds a parity bit fixed to 0.</td> <td>Waits for reception of the parity bit but does not evaluate it.</td> </tr> <tr> <td>1</td> <td>0</td> <td>Adds an odd parity bit.</td> <td>Waits for the odd parity bit.</td> </tr> <tr> <td>1</td> <td>1</td> <td>Adds an even parity bit.</td> <td>Waits for the even parity bit.</td> </tr> </tbody> </table>	CSIHnPSx1	CSIHnPSx0	Transmission	Reception	0	0	Does not transmit any parity bit.	Does not wait for reception of the parity bit.	0	1	Adds a parity bit fixed to 0.	Waits for reception of the parity bit but does not evaluate it.	1	0	Adds an odd parity bit.	Waits for the odd parity bit.	1	1	Adds an even parity bit.	Waits for the even parity bit.
		CSIHnPSx1	CSIHnPSx0	Transmission	Reception																	
		0	0	Does not transmit any parity bit.	Does not wait for reception of the parity bit.																	
		0	1	Adds a parity bit fixed to 0.	Waits for reception of the parity bit but does not evaluate it.																	
		1	0	Adds an odd parity bit.	Waits for the odd parity bit.																	
1	1	Adds an even parity bit.	Waits for the even parity bit.																			
Selects the data length for chip select signal x.																						
<table border="1"> <thead> <tr> <th>CSIHnDLSx[3:0]</th> <th>Data Length</th> </tr> </thead> <tbody> <tr> <td>0000_B</td> <td>16 bits</td> </tr> <tr> <td>0001_B</td> <td>1 bit</td> </tr> <tr> <td>0010_B</td> <td>2 bits</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>1111_B</td> <td>15 bits</td> </tr> </tbody> </table>		CSIHnDLSx[3:0]	Data Length	0000 _B	16 bits	0001 _B	1 bit	0010 _B	2 bits	1111 _B	15 bits									
CSIHnDLSx[3:0]	Data Length																					
0000 _B	16 bits																					
0001 _B	1 bit																					
0010 _B	2 bits																					
...	...																					
1111 _B	15 bits																					
CAUTION: When CSIHnTX0W.CSIHnEDL = 1, the setting of this bit has no effect.																						
When CSIHnTX0W.CSIHnEDL = 0 (the data length is 16 bits), the setting of this bit is valid. Setting "1 bit" is only enabled if the previous transmit data was 16 bits with CSIHnEDL = 1.																						
23 to 20	—	Reserved When read, the value after reset is read. When writing to these bits, write the value after reset.																				
19	CSIHnRCBx	Selects the recessive configuration for broadcasting for chip select signal x. 0: Dominant (higher priority) 1: Recessive (lower priority) For details, see Section 17.5.3.1, Configuration Registers .																				
18	CSIHnDIRx	Selects the serial data direction of chip select signal x. 0: Data is transmitted/received with MSB first. 1: Data is transmitted/received with LSB first. For details, see Section 17.5.8, Serial Data Direction Selection .																				

Table 17.22 CSIHnCFGx Register Contents (2/4)

Bit Position	Bit Name	Function
17	CSIHnCKPx	CSIHnCKPx: Clock phase selection bit
16	CSIHnDAPx	CSIHnDAPx: Data phase selection bit
<ul style="list-style-type: none"> • CSIHnCTL1.CSIHnCKR = 0 		
CSIHnCKPx	CSIHnDAPx	Clock Phase and Data Phase Selection
0	0	
0	1	
1	0	
1	1	
<ul style="list-style-type: none"> • CSIHnCTL1.CSIHnCKR = 1 		
CSIHnCKPx	CSIHnDAPx	Clock Phase and Data Phase Selection
0	0	
0	1	
1	—	Setting prohibited
15	CSIHnIDLx	<p>Selects the idle enforcement configuration for chip select signal x.</p> <p>0: If the CSIHnTX0W.CSIHnCSx settings of two consecutive transfers are different, an idle state is inserted between two transfers. If the CSIHnTX0W.CSIHnCSx settings of two consecutive transfers are the same, an idle state is not inserted between two transfers.</p> <p>1: Regardless of the CSIHnTX0W.CSIHnCSx settings of two consecutive transfers, an idle state is inserted between two transfers.</p> <p>This bit is only available in master mode.</p> <p>For details about the enforced idle state, see Section 17.5.13, Enforced Chip Select Idle Setting.</p>

Table 17.22 CSIHnCFGx Register Contents (3/4)

Bit Position	Bit Name	Function		
14 to 12	CSIHnIDx[2:0]	Selects the idle time for chip select signal x.		
		CSIHnIDx[2:0]	Idle time when	
		000 _B	0.5 transmission clock cycles	
		001 _B	1.0 transmission clock cycle	
		010 _B	1.5 transmission clock cycles	
		011 _B	2.5 transmission clock cycles	
		100 _B	3.5 transmission clock cycles	
		101 _B	4.5 transmission clock cycles	
		110 _B	6.5 transmission clock cycles	
		111 _B	8.5 transmission clock cycles	
These bits are only available in master mode.				
11 to 8	CSIHnHDx[3:0]	Specifies the hold time for chip select signal x in transmission clock cycles.		
		CSIHnHDx[3:0]	Hold time when CSIHnCTL1.CSIHnSIT is 0	Hold time when CSIHnCTL1.CSIHnSIT is 1
		0000 _B	0.5 transmission clock cycles	1.0 transmission clock cycle
		0001 _B	1.0 transmission clock cycle	1.5 transmission clock cycles
		0010 _B	1.5 transmission clock cycles	2.0 transmission clock cycles
		0011 _B	2.5 transmission clock cycles	3.0 transmission clock cycles
		0100 _B	3.5 transmission clock cycles	4.0 transmission clock cycles
		0101 _B	4.5 transmission clock cycles	5.0 transmission clock cycles
		0110 _B	6.5 transmission clock cycles	7.0 transmission clock cycles
		0111 _B	8.5 transmission clock cycles	9.0 transmission clock cycles
		1000 _B	9.5 transmission clock cycles	10.0 transmission clock cycles
		1001 _B	10.5 transmission clock cycles	11.0 transmission clock cycles
		1010 _B	11.5 transmission clock cycles	12.0 transmission clock cycles
		1011 _B	12.5 transmission clock cycles	13.0 transmission clock cycles
		1100 _B	14.5 transmission clock cycles	15.0 transmission clock cycles
		1101 _B	16.5 transmission clock cycles	17.0 transmission clock cycles
		1110 _B	18.5 transmission clock cycles	19.0 transmission clock cycles
		1111 _B	20.5 transmission clock cycles	21.0 transmission clock cycles
These bits are only available in master mode.				

Table 17.22 CSIHnCFGx Register Contents (4/4)

Bit Position	Bit Name	Function																																																			
7 to 4	CSIHnINx[3:0]	Specifies the inter-data time for chip select signal x in transmission clock cycles.																																																			
		<table border="1"> <thead> <tr> <th>CSIHnINx[3:0]</th> <th>Inter-Data Time when CSIHnCTL1.CSIHnSIT is 0</th> <th>Inter-Data Time when CSIHnCTL1.CSIHnSIT is 1</th> </tr> </thead> <tbody> <tr><td>0000_B</td><td>0.0 transmission clock cycles</td><td>0.5 transmission clock cycles</td></tr> <tr><td>0001_B</td><td>0.5 transmission clock cycles</td><td>1.0 transmission clock cycle</td></tr> <tr><td>0010_B</td><td>1.0 transmission clock cycle</td><td>1.5 transmission clock cycles</td></tr> <tr><td>0011_B</td><td>2.0 transmission clock cycles</td><td>2.5 transmission clock cycles</td></tr> <tr><td>0100_B</td><td>3.0 transmission clock cycles</td><td>3.5 transmission clock cycles</td></tr> <tr><td>0101_B</td><td>4.0 transmission clock cycles</td><td>4.5 transmission clock cycles</td></tr> <tr><td>0110_B</td><td>6.0 transmission clock cycles</td><td>6.5 transmission clock cycles</td></tr> <tr><td>0111_B</td><td>8.0 transmission clock cycles</td><td>8.5 transmission clock cycles</td></tr> <tr><td>1000_B</td><td>9.0 transmission clock cycles</td><td>9.5 transmission clock cycles</td></tr> <tr><td>1001_B</td><td>10.0 transmission clock cycles</td><td>10.5 transmission clock cycles</td></tr> <tr><td>1010_B</td><td>11.0 transmission clock cycles</td><td>11.5 transmission clock cycles</td></tr> <tr><td>1011_B</td><td>12.0 transmission clock cycles</td><td>12.5 transmission clock cycles</td></tr> <tr><td>1100_B</td><td>14.0 transmission clock cycles</td><td>14.5 transmission clock cycles</td></tr> <tr><td>1101_B</td><td>16.0 transmission clock cycles</td><td>16.5 transmission clock cycles</td></tr> <tr><td>1110_B</td><td>18.0 transmission clock cycles</td><td>18.5 transmission clock cycles</td></tr> <tr><td>1111_B</td><td>20.0 transmission clock cycles</td><td>20.5 transmission clock cycles</td></tr> </tbody> </table>	CSIHnINx[3:0]	Inter-Data Time when CSIHnCTL1.CSIHnSIT is 0	Inter-Data Time when CSIHnCTL1.CSIHnSIT is 1	0000 _B	0.0 transmission clock cycles	0.5 transmission clock cycles	0001 _B	0.5 transmission clock cycles	1.0 transmission clock cycle	0010 _B	1.0 transmission clock cycle	1.5 transmission clock cycles	0011 _B	2.0 transmission clock cycles	2.5 transmission clock cycles	0100 _B	3.0 transmission clock cycles	3.5 transmission clock cycles	0101 _B	4.0 transmission clock cycles	4.5 transmission clock cycles	0110 _B	6.0 transmission clock cycles	6.5 transmission clock cycles	0111 _B	8.0 transmission clock cycles	8.5 transmission clock cycles	1000 _B	9.0 transmission clock cycles	9.5 transmission clock cycles	1001 _B	10.0 transmission clock cycles	10.5 transmission clock cycles	1010 _B	11.0 transmission clock cycles	11.5 transmission clock cycles	1011 _B	12.0 transmission clock cycles	12.5 transmission clock cycles	1100 _B	14.0 transmission clock cycles	14.5 transmission clock cycles	1101 _B	16.0 transmission clock cycles	16.5 transmission clock cycles	1110 _B	18.0 transmission clock cycles	18.5 transmission clock cycles	1111 _B	20.0 transmission clock cycles	20.5 transmission clock cycles
		CSIHnINx[3:0]	Inter-Data Time when CSIHnCTL1.CSIHnSIT is 0	Inter-Data Time when CSIHnCTL1.CSIHnSIT is 1																																																	
		0000 _B	0.0 transmission clock cycles	0.5 transmission clock cycles																																																	
		0001 _B	0.5 transmission clock cycles	1.0 transmission clock cycle																																																	
		0010 _B	1.0 transmission clock cycle	1.5 transmission clock cycles																																																	
		0011 _B	2.0 transmission clock cycles	2.5 transmission clock cycles																																																	
		0100 _B	3.0 transmission clock cycles	3.5 transmission clock cycles																																																	
		0101 _B	4.0 transmission clock cycles	4.5 transmission clock cycles																																																	
		0110 _B	6.0 transmission clock cycles	6.5 transmission clock cycles																																																	
		0111 _B	8.0 transmission clock cycles	8.5 transmission clock cycles																																																	
		1000 _B	9.0 transmission clock cycles	9.5 transmission clock cycles																																																	
		1001 _B	10.0 transmission clock cycles	10.5 transmission clock cycles																																																	
		1010 _B	11.0 transmission clock cycles	11.5 transmission clock cycles																																																	
		1011 _B	12.0 transmission clock cycles	12.5 transmission clock cycles																																																	
		1100 _B	14.0 transmission clock cycles	14.5 transmission clock cycles																																																	
		1101 _B	16.0 transmission clock cycles	16.5 transmission clock cycles																																																	
		1110 _B	18.0 transmission clock cycles	18.5 transmission clock cycles																																																	
		1111 _B	20.0 transmission clock cycles	20.5 transmission clock cycles																																																	
		These bits are only available in master mode.																																																			
		3 to 0	CSIHnSPx[3:0]	Specifies the setup time for chip select signal x in transmission clock cycles.																																																	
<table border="1"> <thead> <tr> <th>CSIHnSPx[3:0]</th> <th>Setup Time</th> </tr> </thead> <tbody> <tr><td>0000_B</td><td>0.5 transmission clock cycles</td></tr> <tr><td>0001_B</td><td>1.0 transmission clock cycle</td></tr> <tr><td>0010_B</td><td>1.5 transmission clock cycles</td></tr> <tr><td>0011_B</td><td>2.5 transmission clock cycles</td></tr> <tr><td>0100_B</td><td>3.5 transmission clock cycles</td></tr> <tr><td>0101_B</td><td>4.5 transmission clock cycles</td></tr> <tr><td>0110_B</td><td>6.5 transmission clock cycles</td></tr> <tr><td>0111_B</td><td>8.5 transmission clock cycles</td></tr> <tr><td>1000_B</td><td>9.5 transmission clock cycles</td></tr> <tr><td>1001_B</td><td>10.5 transmission clock cycles</td></tr> <tr><td>1010_B</td><td>11.5 transmission clock cycles</td></tr> <tr><td>1011_B</td><td>12.5 transmission clock cycles</td></tr> <tr><td>1100_B</td><td>14.5 transmission clock cycles</td></tr> <tr><td>1101_B</td><td>16.5 transmission clock cycles</td></tr> <tr><td>1110_B</td><td>18.5 transmission clock cycles</td></tr> <tr><td>1111_B</td><td>20.5 transmission clock cycles</td></tr> </tbody> </table>	CSIHnSPx[3:0]			Setup Time	0000 _B	0.5 transmission clock cycles	0001 _B	1.0 transmission clock cycle	0010 _B	1.5 transmission clock cycles	0011 _B	2.5 transmission clock cycles	0100 _B	3.5 transmission clock cycles	0101 _B	4.5 transmission clock cycles	0110 _B	6.5 transmission clock cycles	0111 _B	8.5 transmission clock cycles	1000 _B	9.5 transmission clock cycles	1001 _B	10.5 transmission clock cycles	1010 _B	11.5 transmission clock cycles	1011 _B	12.5 transmission clock cycles	1100 _B	14.5 transmission clock cycles	1101 _B	16.5 transmission clock cycles	1110 _B	18.5 transmission clock cycles	1111 _B	20.5 transmission clock cycles																	
CSIHnSPx[3:0]	Setup Time																																																				
0000 _B	0.5 transmission clock cycles																																																				
0001 _B	1.0 transmission clock cycle																																																				
0010 _B	1.5 transmission clock cycles																																																				
0011 _B	2.5 transmission clock cycles																																																				
0100 _B	3.5 transmission clock cycles																																																				
0101 _B	4.5 transmission clock cycles																																																				
0110 _B	6.5 transmission clock cycles																																																				
0111 _B	8.5 transmission clock cycles																																																				
1000 _B	9.5 transmission clock cycles																																																				
1001 _B	10.5 transmission clock cycles																																																				
1010 _B	11.5 transmission clock cycles																																																				
1011 _B	12.5 transmission clock cycles																																																				
1100 _B	14.5 transmission clock cycles																																																				
1101 _B	16.5 transmission clock cycles																																																				
1110 _B	18.5 transmission clock cycles																																																				
1111 _B	20.5 transmission clock cycles																																																				
These bits are only available in master mode.																																																					

CAUTION

When setting this register, see **Table 17.35, Notes on Setting Registers**.

17.3.8 CSIHnTX0W — CSIHn Transmit Data Register 0 for Word Access

This register stores transmission data. In addition, it specifies the communication interrupt request, the end-of-job, the extended data length, and the chip select activation.

CAUTION

When CSIH0CTL0.CSIH0TXE = CSIH0CTL0.CSIH0RXE = 0 in direct access mode, write access to this register is prohibited.

Value after reset: X0XX XXXX_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CSIHn EOJ	CSIHn EDL	—	—	—	—	—	—	—	CSIHn CS5	CSIHn CS4	CSIHn CS3	CSIHn CS2	CSIHn CS1	CSIHn CS0
Value after reset	0	—	—	0	0	0	0	0	—	—	—	—	—	—	—	—
R/W	R	R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnTX[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.23 CSIHnTX0W Register Contents (1/2)

Bit Position	Bit name	Function
31	—	Reserved When read, the value after reset is read. When writing to these bits, write the value after reset.
30	CSIHnEOJ	Specifies the end of a job. 0: Indicates that it is not end-of-job data. The job continues. 1: Indicates end-of-job data. CAUTION: This bit is only valid when job mode is enabled (CSIHnCTL1.CSIHnJE = 1). This bit must be set to 0 in slave mode.
29	CSIHnEDL	Specifies whether the associated data requires the extended data length (EDL) option. 0: Normal operation 1: Enables the extended data length. The associated data is transmitted as a 16-bit packet. No inter-data time or idle time will be inserted after the data is transmitted. If CSIHnCTL1.CSIHnEDLE = 1 and CSIHnTX0W.CSIHnEDL = 1, the subsequent data must have the same CS selection. If CS is modified for the subsequent data, the correct operation is not assured. CAUTION: This bit is only available if CSIHnCTL1.CSIHnEDLE = 1.
28 to 24	—	Reserved When read, the value after reset is read. When writing to these bits, write the value after reset.
23, 22	—	Reserved When read, the value returned is undefined. The write value should be 1.

Table 17.23 CSIHnTX0W Register Contents (2/2)

Bit Position	Bit name	Function
21 to 16	CSIHnCSx (x =5 to 0)	<p>Activates one or more chip select signals.</p> <p>0: Activates chip select signals x for the associated transmission.</p> <p>1: Deactivates chip select signals x for the associated transmission.</p> <p>Setting bits 23 to 16 = FF_H is prohibited.</p> <p>CAUTION: If several chip select signals are enabled for broadcasting, the configuration of one with CSIHnCFGx.CSIHnRCBx = 0 (dominant) is used. In this case, all dominant chip select signals must be set to precisely the same value.</p> <p>In slave mode, set the bits 23 to 16 are FE_H.</p>
15 to 0	CSIHnTX[15:0]	Stores the transmission data.

CAUTION

When setting this register, see **Table 17.35, Notes on Setting Registers.**

17.3.9 CSIHnTX0H — CSIHn Transmit Data Register 0 for Half Word Access

This register stores the transmission data. This register is the same as bits 15 to 0 of register CSIHnTX0W.

The settings specified by the upper 16 bits of CSIHnTX0W are applied to the transmission. Set transmit data to CSIHnTX0W before using this register because the value of CSIHnTX0W is undefined after reset.

CAUTION

When CSIH0CTL0.CSIH0TXE = CSIH0CTL0.CSIH0RXE = 0 in direct access mode, write access to this register is prohibited.

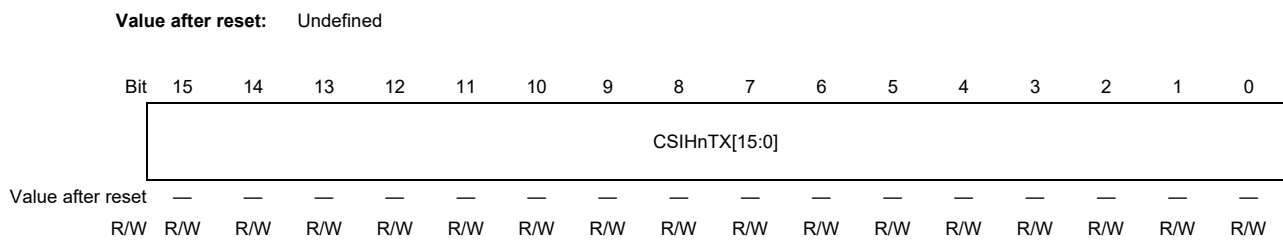


Table 17.24 CSIHnTX0H Register Contents

Bit Position	Bit name	Function
15 to 0	CSIHnTX[15:0]	Stores the transmission data.

CAUTION

When setting this register, see **Table 17.35, Notes on Setting Registers**.

17.3.10 CSIHnRX0W — CSIHn Receive Data Register 0 for Word Access

This register stores the received data.

CAUTION

This register can be read while CSIHnCTL0.CSIHnPWR = 1.

This register is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.

Value after reset: 0XXX XXXX_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	CSIHnRPE	CSIHnTDCE	—	—	CSIHnCS5	CSIHnCS4	CSIHnCS3	CSIHnCS2	CSIHnCS1	CSIHnCS0
Value after reset	0	0	0	0	0	0	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnRX[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 17.25 CSIHnRX0W Register Contents

Bit Position	Bit name	Function
31 to 26	—	Reserved When read, the value after reset is read.
25	CSIHnRPE	Indicates whether a reception data parity error was detected. 0: No parity error was detected on the associated reception data. 1: A parity error was detected on the associated reception data.
24	CSIHnTDCE	Indicates whether a transmission data consistency error was detected. 0: No consistency error was detected on the associated transmission data. 1: A consistency error was detected on the associated transmission data.
23 to 22	—	Reserved When read, the value returned is undefined.
21 to 16	CSIHnCSx (x = 5 to 0)	Indicates which chip select signal was activated. 0: Chip select signal x was activated for the associated reception. 1: Chip select signal x was deactivated for the associated reception.
15 to 0	CSIHnRX[15:0]	Stores the received data.

NOTE

Read this register at least one serial clock cycle before an interrupt is generated.

CAUTION

When setting this register, see **Table 17.35, Notes on Setting Registers**.

17.3.11 CSIHnRX0H — CSIHn Receive Data Register 0 for Half Word Access

This register stores the received data. This register is the same as bits 15 to 0 of register CSIHnRX0W.

CAUTION

This register can be read while CSIHnCTL0.CSIHnPWR = 1.

This register is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.

Value after reset: Undefined

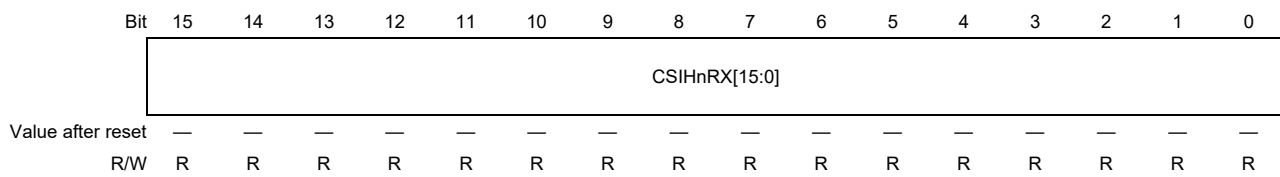


Table 17.26 CSIHnRX0H Register Contents

Bit Position	Bit name	Function
15 to 0	CSIHnRX[15:0]	Stores the received data.

CAUTION

When setting this register, see **Table 17.35, Notes on Setting Registers**.

17.3.12 CSIHnBRSy — CSIHn Baud Rate Setting Register y (y = 0 to 3)

This register sets the transfer clock frequency for each chip select signal.

With CSIHnCFG0 to 5.CSIHnBRSSx[1:0] bits, one of the four types of transfer clock frequency settings can be selected for each chip select signal. For details of transfer clock frequency setting, see **Section 17.5.5, Transmission Clock Selection**.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CSIHnBRS[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.27 CSIHnBRSy Register Contents

Bit Position	Bit Name	Function
15 to 12	—	Reserved When read, the value after reset is read. When writing to these bits, write the value after reset.
11 to 0	CSIHnBRS[11:0]	0: BRG stopped 1: PCLK / ($2^\alpha \times 1 \times 2$) 2: PCLK / ($2^\alpha \times 2 \times 2$) 3: PCLK / ($2^\alpha \times 3 \times 2$) 4: PCLK / ($2^\alpha \times 4 \times 2$) . . . 4095: PCLK / ($2^\alpha \times 4095 \times 2$)

α is the value of CSIHnCTL2.CSIHnPRS[2:0].

CAUTION

When setting this register, see **Table 17.35, Notes on Setting Registers**.

17.3.13 CSIHnEIE — CSIHn Interrupt Control Register

This register controls enabling or disabling the interrupt of CSIH.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CSIHn EIJE	CSIHn EIEE	CSIHn EIRE	CSIHn EICE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 17.28 CSIHnEIE Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is read.
3	CSIHnEIJE	This bit enables or disables the INTCSIHTIJC interrupt. 0: Disable the INTCSIHTIJC interrupt. 1: Enable the INTCSIHTIJC interrupt.
2	CSIHnEIEE	This bit enables or disables the INTCSIHTIRE interrupt. 0: Disable the INTCSIHTIRE interrupt. 1: Enable the INTCSIHTIRE interrupt.
1	CSIHnEIRE	This bit enables or disables the INTCSIHTIR interrupt. 0: Disable the INTCSIHTIR interrupt. 1: Enable the INTCSIHTIR interrupt.
0	CSIHnEICE	This bit enables or disables the INTCSIHTIC interrupt. 0: Disable the INTCSIHTIC interrupt. 1: Enable the INTCSIHTIC interrupt.

CAUTIONS

- Even when the extended communication function is disabled (CSIHnECTL0.CSIHnEEN = 0), enabling or disabling the interrupt of CSIH is effective. When all interrupts of CSIH are used, specify settings to enable all interrupt outputs (CSIHnEIE[3:0] = 1111_B).
- Specify settings to enable or disable interrupt output while communication is not performed. If the settings are specified during communication, the interrupt output operation is not guaranteed.

17.3.14 CSIHnECTL0 — CSIHn Extended Communication Control Register 0

This register is used to control the operation of the extended communication function.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CSIHn EEN	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R

Table 17.29 CSIHnECTL0 Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is read.
7	CSIHnEEN	This bit enables or disables the extended communication function. 0: Disable the extended communication function. 1: Enable the extended communication function.
6 to 0	—	Reserved When read, the value after reset is read.

17.3.15 CSIHnECTL1 — CSIHn Extended Communication Control Register 1

This register is used to control the operation of the extended communication function.

Value after reset: 00FF0000_H

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHnEDIR	—	CSIHnEDS[1:0]		CSIHnETC[3:0]				CSIHnECS[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 17.30 CSIHnECTL1 Register Contents (1/2)

Bit Position	Bit Name	Function
31	CSIHnEDIR	This bit specifies the data transfer direction when the extended communication function is enabled. 0: MSB-first 1: LSB-first
30	—	Reserved When read, the value after reset is read.
29 to 28	CSIHnEDS[1:0]	These bits specify the communication data size when the extended communication function is enabled. 00: 8 bits 01: 16 bits 10: 32 bits 11: Setting prohibited
27 to 24	CSIHnETC[3:0]	These bits specify the number of communications when the extended communication function is enabled. 0000: 1 extended communications 0001: 2 extended communications 0010: 3 extended communications 0011: 4 extended communications 0100: 5 extended communications 0101: 6 extended communications 0110: 7 extended communications 0111: 8 extended communications 1000: 9 extended communications 1001: 10 extended communications 1010: 11 extended communications 1011: 12 extended communications 1100: 13 extended communications 1101: 14 extended communications 1110: 15 extended communications 1111: 16 extended communications

Table 17.30 CSIHnECTL1 Register Contents (2/2)

Bit Position	Bit Name	Function
23 to 16	CSIHnECS[7:0]	<p>When CSIH is in master mode, these bits specify whether to activate the CSx when the extended communication function is enabled. (x = 0 to 5)</p> <p>0: Activate the CSx. 1: Deactivate the CSx.</p> <p>It is prohibited to perform communication with these bits set to FF_H. When CSIH is in slave mode, set these bits set to FE_H. Individual bits correspond to slaves as follows:</p> <p>CSIHnECS7: Setting prohibited CSIHnECS6: Setting prohibited CSIHnECS5: CS5 CSIHnECS4: CS4 CSIHnECS3: CS3 CSIHnECS2: CS2 CSIHnECS1: CS1 CSIHnECS0: CS0</p>
15 to 0	—	<p>Reserved</p> <p>When read, the value after reset is read.</p>

17.3.16 CSIHnETXDAm — CSIHn Extended Communication Transmit Data Register m (m = 0 to 15)

This register is used to control the operation of the extended communication function.

Value after reset: 0000 0000_H

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHnETXDAm[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnETXDAm[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.31 CSIHnETXDAm Register Contents

Bit Position	Bit Name	Function
31 to 0	CSIHnETXDAm[31:0]	These bits retain the data transmitted by the extended communication function.

CAUTION

Access to this register must be done in the same bit mode as the transfer data size.

When the communication data size is 8 bits, access to this register must be 8-bit access.

When the communication data size is 16 bits, access to this register must be 16-bit access.

When the communication data size is 32 bits, access to this register must be 32-bit access.

17.3.17 CSIHnERXDAm — CSIHn Extended Communication Receive Data Register m (m = 0 to 15)

This register is used to control the operation of the extended communication function.

Value after reset: 0000 0000_H

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHnERXDAm[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnERXDAm[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 17.32 CSIHnERXDAm Register Contents

Bit Position	Bit Name	Function
31 to 0	CSIHnERXDAm[31:0]	These bits retain the data received by the extended communication function.

CAUTIONS

- Access to this register must be done in the same bit mode as the transfer data size.
When the communication data size is 8 bits, access to this register must be 8-bit access.
When the communication data size is 16 bits, access to this register must be 16-bit access.
When the communication data size is 32 bits, access to this register must be 32-bit access.
- When transfer completion interrupt "INTCSIHTIJC" occurs, these registers (reception data) must be read only once.

17.3.18 CSIHnESTT — CSIHn Extended Communication Stop Trigger Register

This register is used to control the operation of the extended communication function.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CSIHn ESTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 17.33 CSIHnESTT Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is read.
0	CSIHnESTRG	This bit stops the extended communication of CSIHn. Writing 1 to this bit stops the communication. Writing 0 to this bit is ignored. The value read from this bit is always 0.

For details about the stop trigger register, see **Section 17.5.14.2, Extended Communication Stop Function**.

17.3.19 CSIHnESTF — CSIHn Extended Communication Status Register

This register is used to control the operation of the extended communication function.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CSIHnE TSF	—	—	CSIHnETCF[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 17.34 CSIHnESTF Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is read.
7	CSIHnETSF	This bit is a flag that indicates the status of the extended communication. 0: Extended communication stopped 1: Extended communication in progress This bit is cleared to 0 when CSIHnECTL0.CSIHnEEN changes from 1 to 0.
6 to 5	—	Reserved When read, the value after reset is read.
4 to 0	CSIHnETCF[4:0]	These bits are the flags to indicate the transfer count in extended communication. Flags that indicate the remaining number of 32-bit communications: 0: 0 communications remaining 1: 1 communication remaining 2: 2 communications remaining 3: 3 communications remaining 4: 4 communications remaining 5: 5 communications remaining 6: 6 communications remaining 7: 7 communications remaining 8: 8 communications remaining 9: 9 communications remaining 10: 10 communications remaining 11: 11 communications remaining 12: 12 communications remaining 13: 13 communications remaining 14: 14 communications remaining 15: 15 communications remaining 16: 16 communications remaining This bit is cleared to 0000 _B when CSIHnECTL0.CSIHnEEN changes from 1 to 0.

17.3.20 List of Cautions

Table 17.35 Notes on Setting Registers

Register Name	Bit Name	Cautions
CSIHnCTL0	CSIHnPWR	If this bit is cleared during communication, ongoing communication is aborted. After the communication is aborted, it is necessary to restart the communication.
CSIHnCTL0	CSIHnTXE CSIHnRXE	Do not modify any of these bits while CSIHnCTL0.CSIHnPWR = 0. (These bits can be modified at the same time as the CSIHnCTL0.CSIHnPWR bit.) Do not modify these bits while CSIHnSTR0.CSIHnTSF = 1, because the specified operation is not guaranteed if ongoing communication is aborted.
CSIHnCTL0	CSIHnJOB	Do not modify this bit while CSIHnCTL0.CSIHnPWR = 0. This bit is only valid when CSIHnCTL1.CSIHnJE = 1. Setting this bit is prohibited in slave mode.
CSIHnCTL0	CSIHnMBS	Do not modify this bit while CSIHnCTL0.CSIHnPWR = 0. (This bit can be modified at the same time as the CSIHnCTL0.CSIHnPWR bit.) Modification of this bit is only permitted while CSIHnSTR0.CSIHnTSF = 0.
CSIHnCTL1	CSIHnCKR	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. When CS is not used, use this bit instead of CSIHnCFGx.CSIHnCKPx and set CSIHnCFGx.CSIHnCKPx to 0. This bit must be used in slave mode.
CSIHnCTL1	CSIHnSLIT CSIHnCSL[5:0] CSIHnEDLE CSIHnDCS CSIHnCSRI CSIHnHSE	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0.
CSIHnCTL1	CSIHnJE CSIHnLBM	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting of these bits are prohibited in slave mode.
CSIHnCTL1	CSIHnSSE	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting this bit to 1 is prohibited in master mode.
CSIHnCTL1	CSIHnSIT	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. This bit is only valid in master mode. In slave mode, no delay is generated.
CSIHnCTL2	CSIHnPRS[2:0]	Modification of these bits are only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting of the maximum transfer clock frequency is as follows. Master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/8) Slave mode: 4.0 MHz (however, it must be equal to or lower than PCLK/20)
CSIHnSTR0	CSIHnTSF	Writing to this bit is prohibited, and only reading is permitted.
CSIHnSTR0	CSIHnDCE CSIHnPE CSIHnOVE	These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1 → 0. Writing to these bits are prohibited, and only reading is permitted.
CSIHnCFGx x = 0 to 5	CSIHnBRSSx[1:0] CSIHnRCBx CSIHnIDLx CSIHnIDx[2:0] CSIHnHDx[3:0] CSIHnINx[3:0] CSIHnSPx[3:0]	Modification of these bits are only permitted while CSIHnCTL0.CSIHnPWR = 0. These bits must be set to 0 in slave mode.
CSIHnCFGx x = 0 to 5	CSIHnPSx[1:0] CSIHnDLSx[3:0] CSIHnDIRx CSIHnDAPx	Modification of these bits are only permitted while CSIHnCTL0.CSIHnPWR = 0. In slave mode, the CSIHnCFG0 setting is used for the configuration. Therefore, all the bits in CSIHnCFG1 to CSIHnCFG5 must be set to 0.

Table 17.35 Notes on Setting Registers

Register Name	Bit Name	Cautions
CSIHnCFGx x = 0 to 5	CSIHnCKPx	Modification of these bits are only permitted while CSIHnCTL0.CSIHnPWR = 0. Set these bits to 0 as CSIHnCTL1.CSIHnCKR must be used in slave mode. If CS is not used, use the CSIHnCTL1.CSIHnCKR bit instead of this bit, and clear these bits to 0.
CSIHnTX0W	CSIHnEOJ	This bit is only valid when CSIHnCTL1.CSIHnJE = 1. While CSIHnCTL1.CSIHnJE = 0, values of this bit is ignored even if 1 is read. Set this bit to 0 in slave mode.
CSIHnTX0W	CSIHnEDL	This bit is only valid when CSIHnCTL1.CSIHnEDLE = 1. While CSIHnCTL1.CSIHnEDLE = 0, the value of this bit is ignored even if 1 is read.
CSIHnTX0W	CSIHnCS[5:0]	In master mode, setting bits 23 to 16 = "FF _H " is prohibited. In slave mode, set these bits to "FE _H ".
CSIHnTX0W CSIHnTX0H		While CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0, writing to these bits are prohibited in direct access mode.
CSIHnRX0W CSIHnRX0H		These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1 → 0.
CSIHnBRSy y = 0 to 3		Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. The maximum baud rates are determined in combination with the setting of the CSIHnCTL2.PRS[2:0] bits and are as follows. Master mode: PCLK/8 Slave mode: PCLK/20

Table 17.36 Precautions on Extended Communication Function

Item	Description																																				
Restriction on the use of extended communication function	<p>When the extended communication function (by setting the CSIHnECTL0.CSIHnEEN bit to 1) is enabled, specify the following settings for CSIH:</p> <table border="1"> <thead> <tr> <th>Name of Register Bit</th> <th>Setting</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>CSIHnCTL0.CSIHnMBS</td> <td>1</td> <td>Set the access mode to the direct access mode.</td> </tr> <tr> <td>CSIHnCTL1.CSIHnEDLE</td> <td>* right column</td> <td>Communication data size 8 bits or 16 bits: 0 Communication data size 32 bits: 1</td> </tr> <tr> <td>CSIHnCTL1.CSIHnSLIT</td> <td>1</td> <td>Set the timing of INTCSIHTIC interrupt output during direct access to the timing of transmit data transfer to the shift register.</td> </tr> <tr> <td>CSIHnCTL1.CSIHnSIT</td> <td>0</td> <td>Cancel the delay of half a communication clock cycle in interrupt output.</td> </tr> <tr> <td>CSIHnCTL1.CSIHnJE</td> <td>0</td> <td>Disable the job mode.</td> </tr> <tr> <td>CSIHnCFGx.CSIHnDLSx[3:0] *x: 0 to 5</td> <td>* right column</td> <td>Communication data size 8bit: 8 Communication data size 16bit or 32bit: 0</td> </tr> <tr> <td>CSIHnCFGx.CSIHnDIRx *x: 0 to 5</td> <td>0</td> <td>Fix the serial data direction of CSIH to 0 (MSB). Also specify the serial data direction (MSB, LSB) in CSIHnECTL1.CSIHnEDIR.</td> </tr> </tbody> </table>	Name of Register Bit	Setting	Description	CSIHnCTL0.CSIHnMBS	1	Set the access mode to the direct access mode.	CSIHnCTL1.CSIHnEDLE	* right column	Communication data size 8 bits or 16 bits: 0 Communication data size 32 bits: 1	CSIHnCTL1.CSIHnSLIT	1	Set the timing of INTCSIHTIC interrupt output during direct access to the timing of transmit data transfer to the shift register.	CSIHnCTL1.CSIHnSIT	0	Cancel the delay of half a communication clock cycle in interrupt output.	CSIHnCTL1.CSIHnJE	0	Disable the job mode.	CSIHnCFGx.CSIHnDLSx[3:0] *x: 0 to 5	* right column	Communication data size 8bit: 8 Communication data size 16bit or 32bit: 0	CSIHnCFGx.CSIHnDIRx *x: 0 to 5	0	Fix the serial data direction of CSIH to 0 (MSB). Also specify the serial data direction (MSB, LSB) in CSIHnECTL1.CSIHnEDIR.												
Name of Register Bit	Setting	Description																																			
CSIHnCTL0.CSIHnMBS	1	Set the access mode to the direct access mode.																																			
CSIHnCTL1.CSIHnEDLE	* right column	Communication data size 8 bits or 16 bits: 0 Communication data size 32 bits: 1																																			
CSIHnCTL1.CSIHnSLIT	1	Set the timing of INTCSIHTIC interrupt output during direct access to the timing of transmit data transfer to the shift register.																																			
CSIHnCTL1.CSIHnSIT	0	Cancel the delay of half a communication clock cycle in interrupt output.																																			
CSIHnCTL1.CSIHnJE	0	Disable the job mode.																																			
CSIHnCFGx.CSIHnDLSx[3:0] *x: 0 to 5	* right column	Communication data size 8bit: 8 Communication data size 16bit or 32bit: 0																																			
CSIHnCFGx.CSIHnDIRx *x: 0 to 5	0	Fix the serial data direction of CSIH to 0 (MSB). Also specify the serial data direction (MSB, LSB) in CSIHnECTL1.CSIHnEDIR.																																			
Restriction on CSIH power-off during extended communication	<p>When turning off the CSIH power during extended communication, be sure to disable the extended communication function before CSIH power-off.</p> <ol style="list-style-type: none"> Write 0 to the CSIHnECTL0.CSIHnEEN bit. Write 0 to the CSIHnCTL0.CSIHnPWR bit. 																																				
Restriction of 32-bit slave receive-only mode with parity check	<p>Parity check is not available in 32-bit slave receive-only mode. Therefore, the following settings are prohibited.</p> <p>[Prohibited settings]</p> <table border="1"> <thead> <tr> <th>Name of Register Bit</th> <th>Value</th> <th>Setting</th> </tr> </thead> <tbody> <tr> <td>CSIHnCTL2.CSIHnPRS2-0</td> <td>11_B</td> <td>Slave setting</td> </tr> <tr> <td>CSIHnCTL0.CSIHnTXE</td> <td>0_B</td> <td>Transmission disabled</td> </tr> <tr> <td>CSIHnCTL0.CSIHnRXE</td> <td>1_B</td> <td>Reception enabled</td> </tr> <tr> <td>CSIHnCFGx.CSIHnPSx[1:0]</td> <td>Other than 00_B</td> <td>Adds a parity bit setting</td> </tr> <tr> <td>CSIHnETCL0.CSIHnEDS[1:0]</td> <td>10_B</td> <td>Communication data size 32 bits</td> </tr> </tbody> </table> <p>To enable parity check in transfer slave receive-only mode, set the transmit/receive mode and write the dummy transmission data.</p> <p>[Prohibited settings]</p> <table border="1"> <thead> <tr> <th>Name of Register Bit</th> <th>Value</th> <th>Setting</th> </tr> </thead> <tbody> <tr> <td>CSIHnCTL2.CSIHnPRS2-0</td> <td>11_B</td> <td>Slave setting</td> </tr> <tr> <td>CSIHnCTL0.CSIHnTXE</td> <td>1_B</td> <td>Transmission enabled</td> </tr> <tr> <td>CSIHnCTL0.CSIHnRXE</td> <td>1_B</td> <td>Reception enabled</td> </tr> <tr> <td>CSIHnCFGx.CSIHnPSx[1:0]</td> <td>Other than 00_B</td> <td>Adds a parity bit setting</td> </tr> <tr> <td>CSIHnETCL0.CSIHnEDS[1:0]</td> <td>10_B</td> <td>Communication data size 32 bits</td> </tr> </tbody> </table>	Name of Register Bit	Value	Setting	CSIHnCTL2.CSIHnPRS2-0	11 _B	Slave setting	CSIHnCTL0.CSIHnTXE	0 _B	Transmission disabled	CSIHnCTL0.CSIHnRXE	1 _B	Reception enabled	CSIHnCFGx.CSIHnPSx[1:0]	Other than 00 _B	Adds a parity bit setting	CSIHnETCL0.CSIHnEDS[1:0]	10 _B	Communication data size 32 bits	Name of Register Bit	Value	Setting	CSIHnCTL2.CSIHnPRS2-0	11 _B	Slave setting	CSIHnCTL0.CSIHnTXE	1 _B	Transmission enabled	CSIHnCTL0.CSIHnRXE	1 _B	Reception enabled	CSIHnCFGx.CSIHnPSx[1:0]	Other than 00 _B	Adds a parity bit setting	CSIHnETCL0.CSIHnEDS[1:0]	10 _B	Communication data size 32 bits
Name of Register Bit	Value	Setting																																			
CSIHnCTL2.CSIHnPRS2-0	11 _B	Slave setting																																			
CSIHnCTL0.CSIHnTXE	0 _B	Transmission disabled																																			
CSIHnCTL0.CSIHnRXE	1 _B	Reception enabled																																			
CSIHnCFGx.CSIHnPSx[1:0]	Other than 00 _B	Adds a parity bit setting																																			
CSIHnETCL0.CSIHnEDS[1:0]	10 _B	Communication data size 32 bits																																			
Name of Register Bit	Value	Setting																																			
CSIHnCTL2.CSIHnPRS2-0	11 _B	Slave setting																																			
CSIHnCTL0.CSIHnTXE	1 _B	Transmission enabled																																			
CSIHnCTL0.CSIHnRXE	1 _B	Reception enabled																																			
CSIHnCFGx.CSIHnPSx[1:0]	Other than 00 _B	Adds a parity bit setting																																			
CSIHnETCL0.CSIHnEDS[1:0]	10 _B	Communication data size 32 bits																																			
Caution of error occurrence	<p>When using the extended communication function, CSIH cannot determine the data bit of the communication data in which the parity check error and the data consistency check error occur.</p>																																				
Access to the CSIH register during communication using the extended communication function	<p>Continuous access to the CSIH registers is prohibited during CSI communication using the extended communication function.</p>																																				

17.4 Interrupt Sources

CSIH can generate the following interrupt requests:

- INTCSIHTIC (communication status interrupt)
- INTCSIHTIR (reception status interrupt)
- INTCSIHTIRE (communication error interrupt)
- INTCSIHTIJC (job completion interrupt)

17.4.1 Overview

The communication error interrupt INTCSIHTIRE is generated when an error is detected. The generation of the other interrupts depends on the memory mode, the job mode, and – in case of the job completion interrupt INTCSIHTIJC – also the operating mode.

The job completion interrupt INTCSIHTIJC is only generated when job mode is enabled (CSIHnCTL1.CSIHnJE = 1). It is not available in slave mode.

All of the interrupt requests are controlled by the CSIHnEIE register.

To enable each of the interrupt request must be set to 1 each bit of the CSIHnEIE register.

The following table gives an overview.

Table 17.37 Interrupt Generation

Memory Mode	Interrupt	Interrupt Source	
		Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
Direct access	INTCSIHTIC	One data transfer	One data transfer except the state of job abort*1
	INTCSIHTIR	Data received and CSIHnCTL0.CSIHnRXE = 1	Data received and CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC*2	Not applicable	Job abort*1

Note 1. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1

Note 2. INTCSIHTIJC is not available in slave mode.

17.4.2 Interrupt Delay

In master mode, all interrupts generated by the master can be delayed by half a cycle of the transmission clock, CSIHTSCK. This is not possible in slave mode.

The delay is specified by setting $\text{CSIHnCTL1.CSIHnSIT} = 1$. (The setting of the CSIHnSIT bit is invalid in slave mode.)

The following example illustrates the interrupt delay function, assuming a setting of $\text{CSIHnCTL1.CSIHnSIT} = 1$ (interrupt delay enabled), $\text{CSIHnCFGx.CSIHnCKPx} = 0$, $\text{CSIHnCFGx.CSIHnDAPx} = 0$ (clock phase and data phase), and $\text{CSIHnCFGx.CSIHnDLSx}[3:0] = 1000_{\text{B}}$ (data length 8 bits).

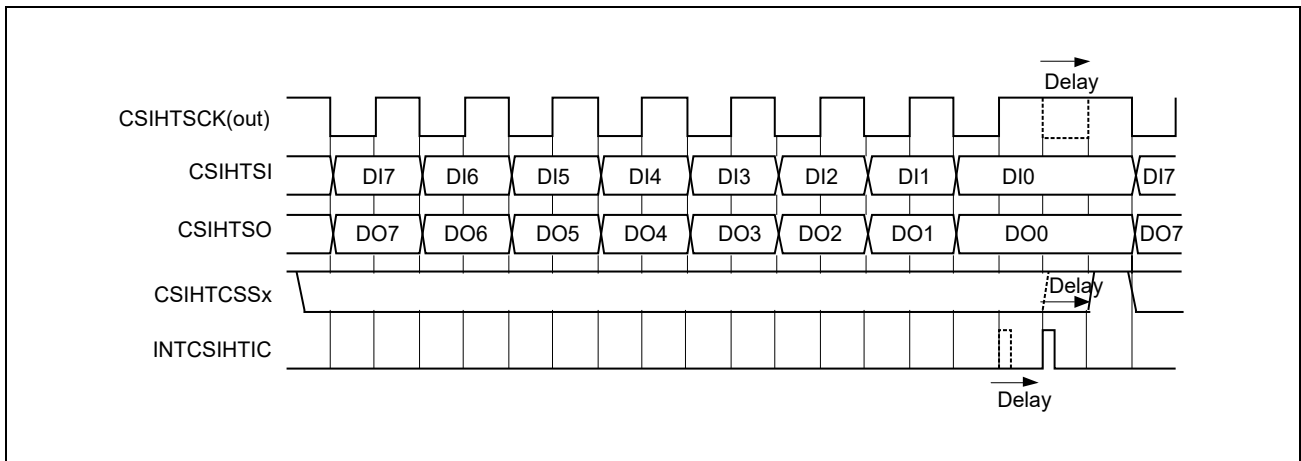


Figure 17.2 Interrupt Delay Function ($\text{CSIHnCTL1.CSIHnSIT} = 1$)

Setting $\text{CSIHnCTL1.CSIHnSIT} = 1$ adds a half cycle delay to the transmission clock. This also delays the end of the present chip select signal (CSIHTCSSx).

17.4.3 INTCSIHTIC (Communication Status Interrupt)

Depending on the job mode, this interrupt is generated according to the conditions shown in the following table.

Table 17.38 INTCSIHTIC Interrupt Generation

Memory Mode	Cause of Interrupt	
	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
Direct access	Generated after every data transfer.	Generated after every data transfer, except when the communication was aborted.

17.4.3.1 INTCSIHTIC in Direct Access Mode

The examples below show the INTCSIHTIC behavior in direct access mode.

The examples assume:

- Master mode
- Direct access mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase
(CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLsx[3:0] = 1000_B)
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Enable the INTCSIHTIC (CSIHnEIE.CSIHnEICE = 1)

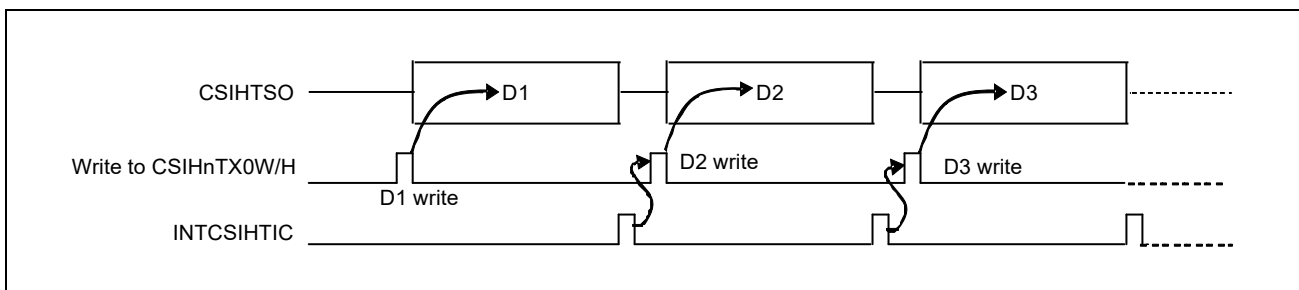


Figure 17.3 Generation of INTCSIHTIC after Transfer (CSIHnCTL1.CSIHnSLIT = 0)

If job mode is enabled (CSIHnCTL1.CSIHnJE = 1) and a job ends because data is sent with CSIHnTXOW.CSIHnEOJ = 1 and communication stop is requested (CSIHnCTL0.CSIHnJOBE = 1), then INTCSIHTIC is replaced by the job completion interrupt INTCSIHTIJC.

INTCSIHTIC can also be set up to occur as soon as the CSIHnTX0W/H register becomes empty and available for storing the next data. This is specified by setting CSIHnCTL1.CSIHnSLIT = 1.

The effect is illustrated in the figure below.

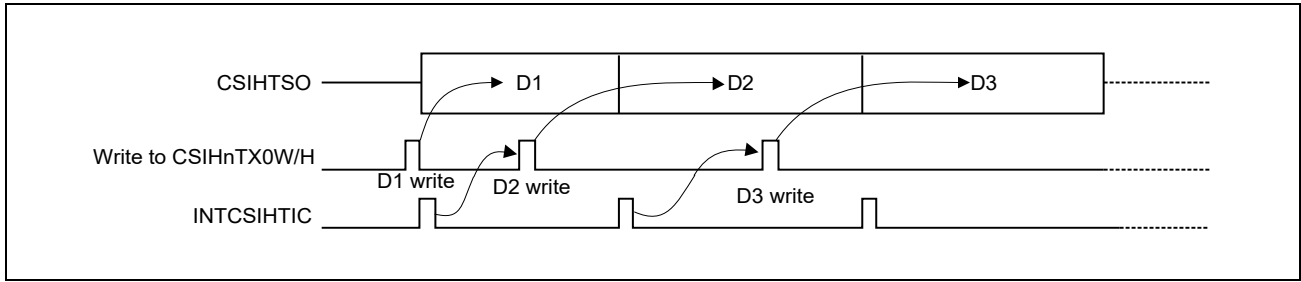


Figure 17.4 Immediate Generation of INTCSIHTIC (CSIHnCTL1.CSIHnSLIT = 1)

Thus, the new data can be written in advance.

NOTE

This mode allows faster data transfer but is only available in direct access mode.

17.4.3.2 INTCSIHTIC in Job Mode

The example below shows the INTCSIHTIC behavior in job mode.

The example assumes:

- Master mode
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase
(CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Enable the INTCSIHTIC (CSIHnEIE.CSIHnEICE = 1)

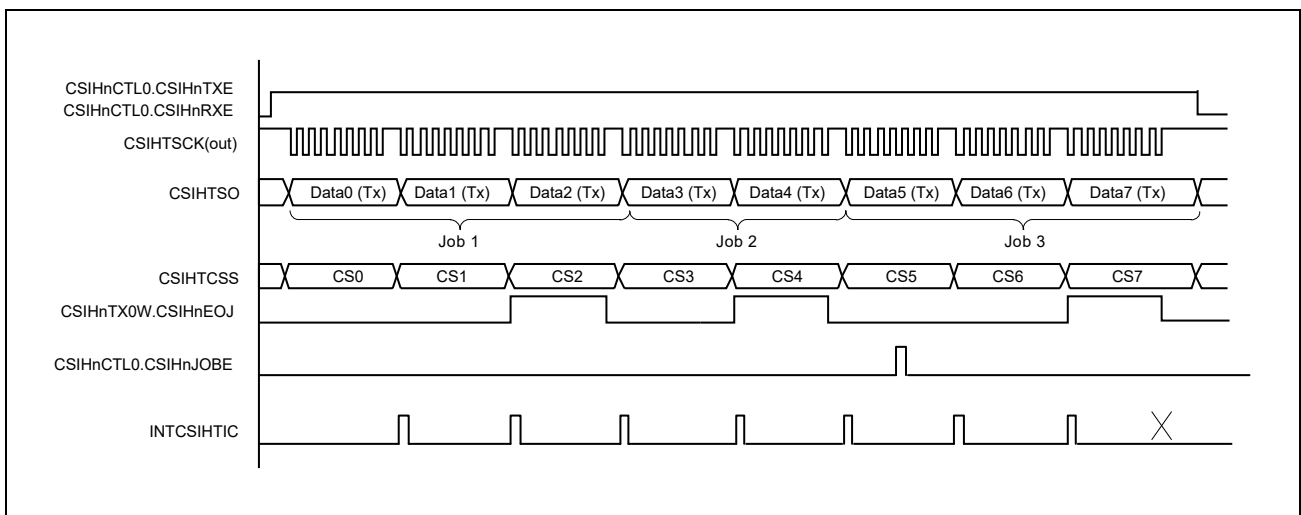


Figure 17.5 Generation of INTCSIHTIC in Job Mode

The rules for generating INTCSIHTIC in job mode are shown in the following table.

Table 17.39 Generation of INTCSIHTIC in Job Mode

CSIHnTX0W.CSIHnEOJ	INTCSIHTIC
0	Generated
1	CSIHnCTL0.CSIHnJOBE = 0: Generated
	CSIHnCTL0.CSIHnJOBE = 1: Not generated, replaced by interrupt INTCSIHTIJC

17.4.4 INTCSIHTIR (Reception Status Interrupt)

This interrupt is generated according to the conditions below.

Table 17.40 INTCSIHTIR Interrupt Generation

Memory Mode	Cause of Interrupt	
	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
Direct access	Generated after every data transfer.	

17.4.4.1 INTCSIHTIR in Direct Access Mode

The example below shows the INTCSIHTIR behavior in direct access mode.

The example below assumes:

- Master mode
- Direct access mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase
(CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Enable the INTCSIHTIR (CSIHnEIE.CSIHnEIRE = 1)

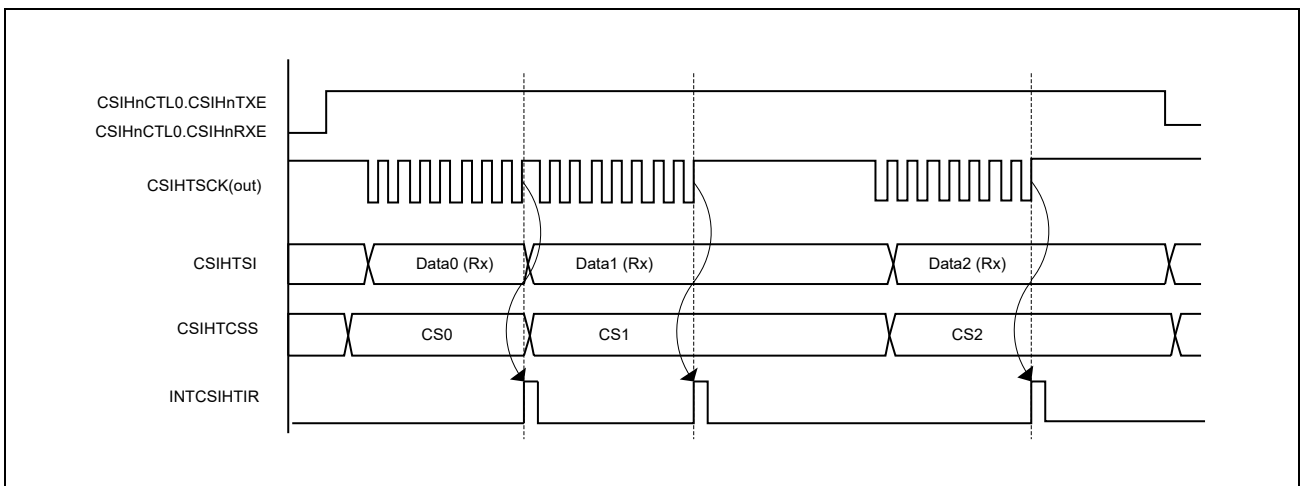


Figure 17.6 Generation of INTCSIHTIR in Direct Access Mode

17.4.5 INTCSIHTIRE (Communication Error Interrupt)

This interrupt is generated whenever an error is detected.

For details about interrupt generation timing, see **Section 17.5.11, Error Detection**.

Table 17.41 Data Error Types

Error Type	Communication Status after Error Interrupt	Note
Parity error	Communication continues even if an interrupt is generated.	—
Data consistency error	Communication continues even if an interrupt is generated.	—
Overrun error	Communication continues even if an interrupt is generated.	This error interrupt is only generated when CSIHnCTL1.CSIHnHSE = 0 (without handshake) in slave mode.

The type of error that caused the generation of INTCSIHTIRE is flagged in register CSIHnSTR0.

Additionally a parity error flag and a data consistency error flag are attached to the received data in CSIHnRX0W.

For details about the various error types, see **Section 17.5.11, Error Detection**.

17.4.6 INTCSIHTIJC (Job Completion Interrupt)

This interrupt supports the handling of jobs. For details, see **Section 17.5.3.3, Job Concept**. This interrupt is only available in master mode.

Job mode is enabled by setting CSIHnCTL1.CSIHnJE = 1. When CSIHnCTL1.CSIHnJE = 0, INTCSIHTIJC is not generated.

this interrupt is generated according to the condition shown in the following table.

Table 17.42 INTCSIHTIJC Interrupt Generation

Memory Mode	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
Direct access	Not applicable	Indicates that the communication stopped at the end of a job after a job abort* ¹ was triggered.

Note 1. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1

17.5 Operation

17.5.1 Operating Modes (Master/Slave)

Whether CSIH operates in, the master or slave mode determines the source of the serial clock.

17.5.1.1 Master Mode

In master mode, the serial transmission clock is generated by the internal baud rate generator (BRG) and supplied to the slave(s) via CSIHTSCK signal.

Master mode is enabled by setting $\text{CSIHnCTL2.CSIHnPRS}[2:0]$ to values other than 111_B . In master mode, the BRG frequency can be specified by setting the $\text{CSIHnCTL2.CSIHnPRS}[2:0]$ bits in combination with the $\text{CSIHnBRSy.CSIHnBRS}[11:0]$ bits.

(1) Chip select signals

In master mode, one or more chip select signals can be used. If several slaves are connected to the master, the chip select signals can be used to select one or more slaves. Only the selected slave is then enabled for communication.

The communication protocol as well as additional parameters are stored separately for each chip select signal. This makes it possible to adapt the data transfer individually to the requirements of each slave. For details, see **Section 17.5.3, Chip Selection (CS) Features**.

(2) Clock defaults

The default level of CSIHTSCK depends on the clock inversion function bit of the CSIHTSCK, and is high when $\text{CSIHnCTL1.CSIHnCKR} = 0$ and is low when $\text{CSIHnCTL1.CSIHnCKR} = 1$.

The example below shows the communication in master mode for 8-bit data, $\text{CSIHnCTL1.CSIHnCKR} = 0$, $\text{CSIHnCFGx.CSIHnCKPx} = 0$, $\text{CSIHnCFGx.CSIHnDAPx} = 0$, and MSB first.

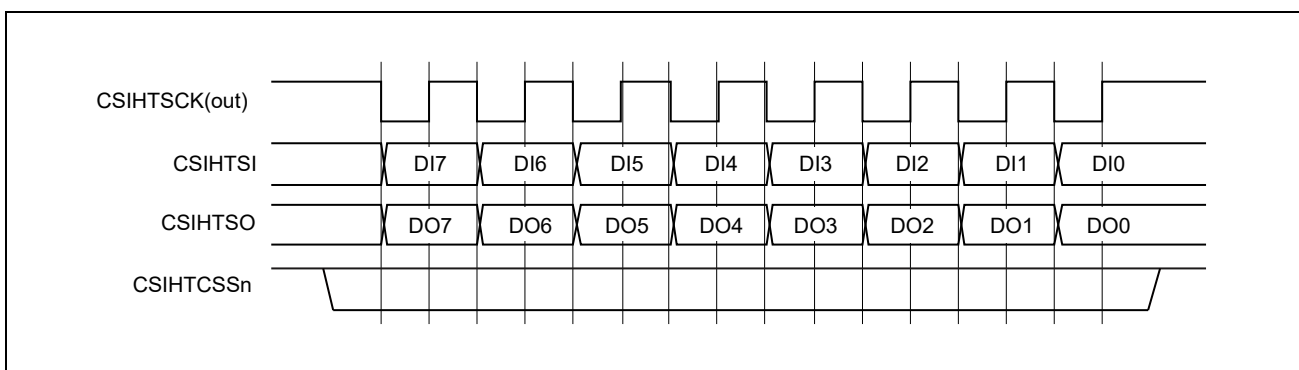


Figure 17.7 Transmission/Reception in Master Mode

17.5.1.2 Slave Mode

In slave mode, another device is the communication master and supplies the transmission clock. Normal transmit/receive operation is started as soon as a clock signal is detected.

Slave mode is selected by setting the CSIHnCTL2.CSIHnPRS[2:0] bits to 111_B.

In slave mode, the transmission protocol setting by the CSIHnCFG0 registers are enabled (setting of the CSIHnCFG1-CSIHnCFG7 register is disabled).

- CSIHnPSx[1:0]: parity usage
- CSIHnDLsx[3:0]: data length selection
- CSIHnDIRx: data direction
- CSIHnCKPx, CSIHnDAPx: clock phase and data phase

NOTE

When using slave mode, disable the baud rate generator (BRG) by setting the CSIHnBRSy.CSIHnBRS[11:0] bits to 000_H.

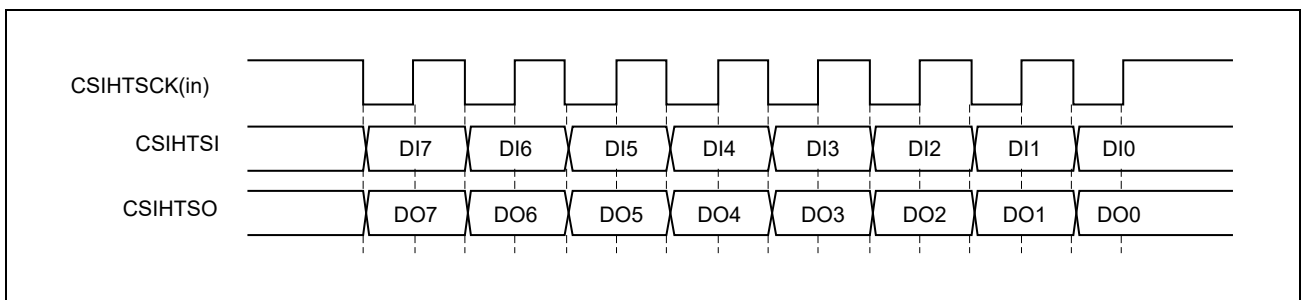


Figure 17.8 Transmission/Reception in Slave Mode

17.5.2 Master/Slave Connections

17.5.2.1 One Master and One Slave

The following figure illustrates the connections between one master and one slave.

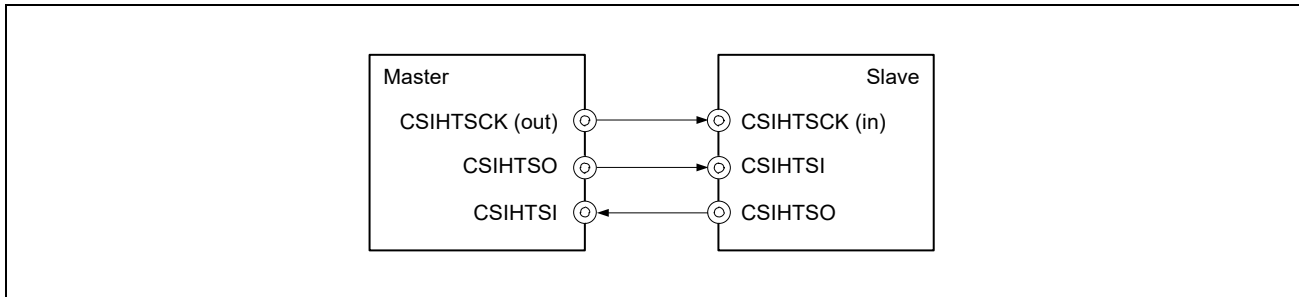


Figure 17.9 Direct Master/Slave Connection

17.5.2.2 One Master and Multiple Slaves

The following figure illustrates the connections between one master and multiple slaves. In this example, the master supplies one chip select (CS) signal to each of the slaves. This signal is connected to the slave select input $\overline{\text{CSIH TSSI}}$ of the slave.

The $\overline{\text{CSIH TSSI}}$ signal can be enabled/disabled by using the $\text{CSIHnCTL1.CSIHnSSE}$ bit.

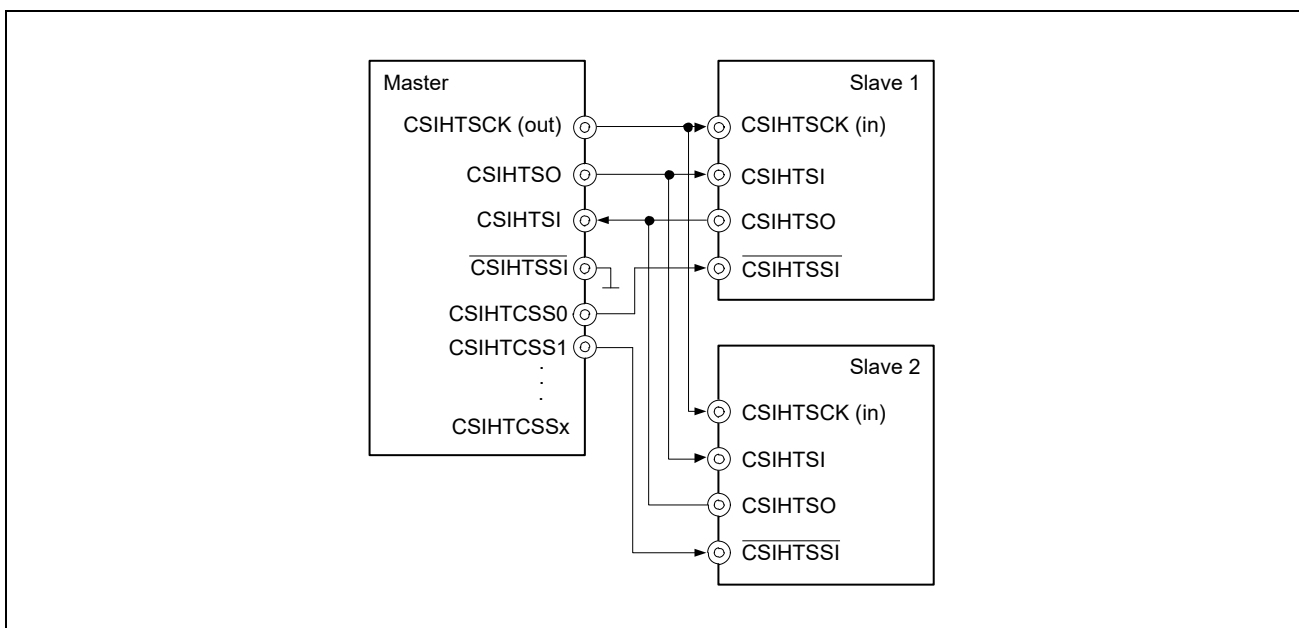


Figure 17.10 Connections between One Master and Multiple Slaves

By default, the chip select level is active low. That means, a slave is selected (enabled) as a CSIH slave when its $\overline{\text{CSIH TSSI}}$ signal is low level. However, to adapt the CS to other devices, the output level of each chip select signal can also be programmed to be active high.

If a slave is not selected, it will neither receive nor transmit data. In addition, when transmit-only mode or transmit/receive mode is set ($\text{CSIHnCTL0.CSIHnTXE} = 1$), the CSIH TSO output of the slaves that are not selected is disabled and set to input mode in order to avoid interference with the output of the selected slave.

17.5.3 Chip Selection (CS) Features

The chip select signal, CSIHnTCSx can be used by the master to select one or more slaves for communication.

17.5.3.1 Configuration Registers

The parameters for each chip select signal CSIHnTCSx are defined in the corresponding configuration register CSIHnCFGx. The parameters include the communication protocol and additional CS parameters.

The communication protocol specifies:

- Data length: The number of bits to be sent or received.
(CSIHnCFGx.CSIHnDLSx[3:0])
- Transfer direction: MSB or LSB first.
(CSIHnCFGx.CSIHnDIRx)
- Parity usage: Odd, even, 0 parity or none.
(CSIHnCFGx.CSIHnPSx[1:0])
- Clock phase and data phase.
(CSIHnCFGx.CSIHnCKPx, CSIHnCFGx.CSIHnDAPx)

Additional parameters for each chip select signal that are only available in master mode are:

- Individual selection of the baud rate generator prescaler for each chip select signal
(CSIHnCFGx.CSIHnBRSSx[1:0])
- Chip select priority: Categorizes chip select signals into “dominant” and “recessive”. The priority applies if two or more chip select signals with different configurations are simultaneously activated for message broadcasting. In this case, the configuration specified for dominant chip select signals is used. (CSIHnCFGx.CSIHnRCBx)

The principle is also called “Recessive Configuration for Broadcasting” (RCB).

CAUTION

When specifying multiple chip select signals as dominant, be sure to configure the same settings for all dominant signals.

- Chip select timing:
 - Setup time Tsetup: The time from when the CS signal becomes active to the start of data output.
(CSIHnCFGx.CSIHnSPx[3:0])
 - Inter-data time Tinter: The time between one data and the next data while the same CS signal is active.
(CSIHnCFGx.CSIHnINx[3:0])
 - Hold time Thold: The time during which the CS signal remains active until CS is switched.
(CSIHnCFGx.CSIHnHDx [3:0])
 - Idle time Tidle: Inactive time after terminating a CS signal or after every data transfer to the same CSx.
(CSIHnCFGx.CSIHnIDx[2:0])

The CS timings of the setup time, the inter-data time, the hold time, and the idle time are illustrated in the figure below. When CSIHnCFGx.CSIHnIDLx bit is set to 1, IDLE time is inserted for every transfer regardless of CS signal.

Figure 17.11 provides an example of when the default active low setting is specified for the CS1HTCSS1 and CS1HTCSS2 signals (CS1HnCTL1.CS1HnCSL1 bit = 0, CS1HnCTL1.CS1HnCSL2 bit = 0). The active level can be specified individually for each CS.

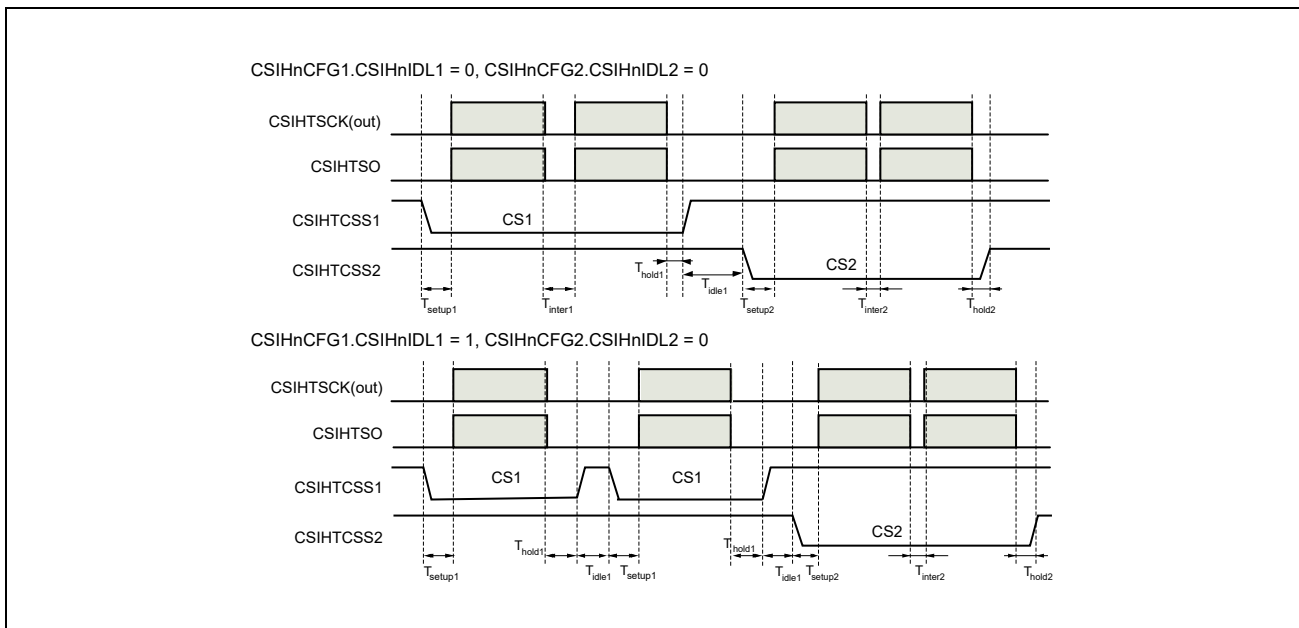


Figure 17.11 Chip Select Timings

Note that each CS signal can have a different value for the setup time, inter-data time, hold time, and idle time.

A particular chip select signal is activated by setting the appropriate bit in the transmission register CS1HnTX0W.CS1HnCSx.

CS1HnRX0W.CS1HnCSx in the reception register indicates the chip select signal associated with the received data.

17.5.3.2 CS Example

The following figure shows an example of two consecutive data transmissions.

The first communication uses CS0 to communicate with one slave. The second enables CS0 and CS1 to broadcast a message to two slaves. The priority of CS0 is set to “recessive: low priority” and the priority of CS1 to “dominant: high priority”. Consequently, the second communication is conducted using the CS1 settings, which are set as dominant.

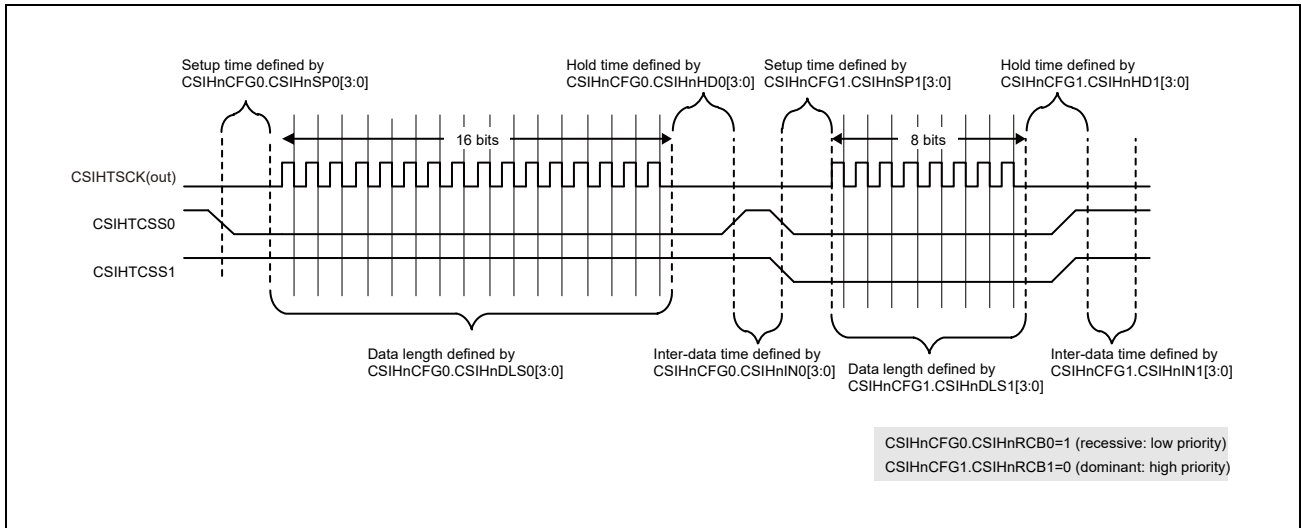


Figure 17.12 Chip Select and RCB Example

17.5.4 Details of Chip Select Timing

17.5.4.1 Changing the Clock Phase

The serial clock level specified by CSIHnCFGx.CSIHnCKPx can be changed while communication is stopped. The minimum value of an idle time is one transmission clock (CSIHTSCK(out)) cycle.

If the idle time is set to 0.5 transmission clock cycles (in CSIHnCFGx.CSIHnIDx[2:0]) and two consecutive data is sent with different CSIHnCFGx.CSIHnCKPx configuration, the idle time is automatically extended to one CSIHTSCK(out) cycle.

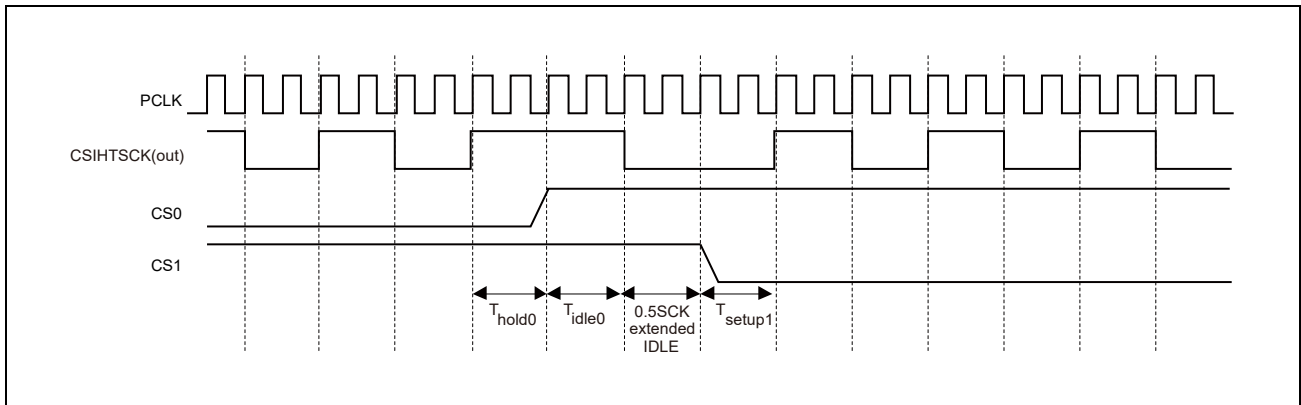


Figure 17.14 Clock Phase Timing with PCLK/8, $T_{hold0} = T_{setup1} = 0.5CSIHTSCK$, $T_{idle0} = 0.5CSIHTSCK$, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG1.CSIHnCKP1 = 1 (CSIHTCSS1)

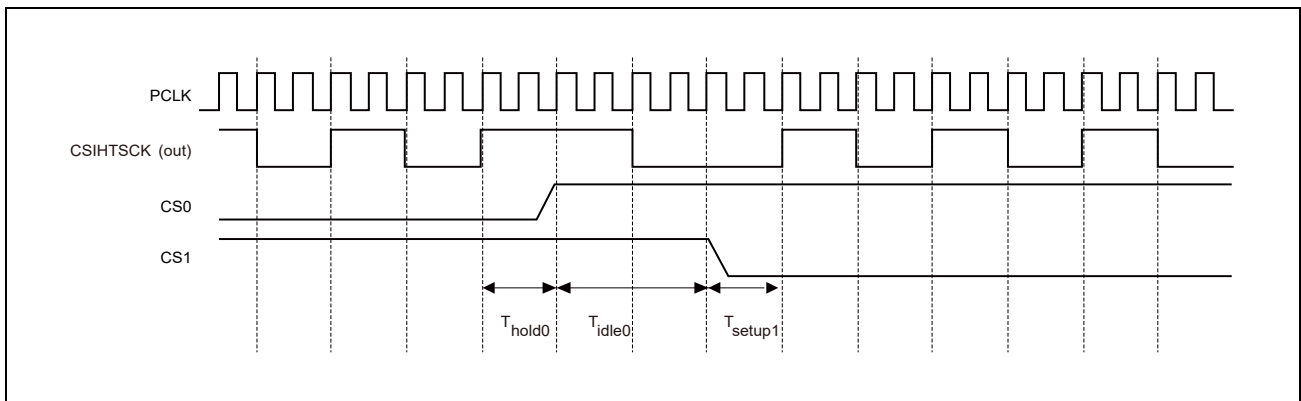


Figure 17.15 Clock Phase Timing with PCLK/8, $T_{hold0} = T_{setup1} = 0.5CSIHTSCK$, $T_{idle0} = 1CSIHTSCK$, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG1.CSIHnCKP1 = 1 (CSIHTCSS1)

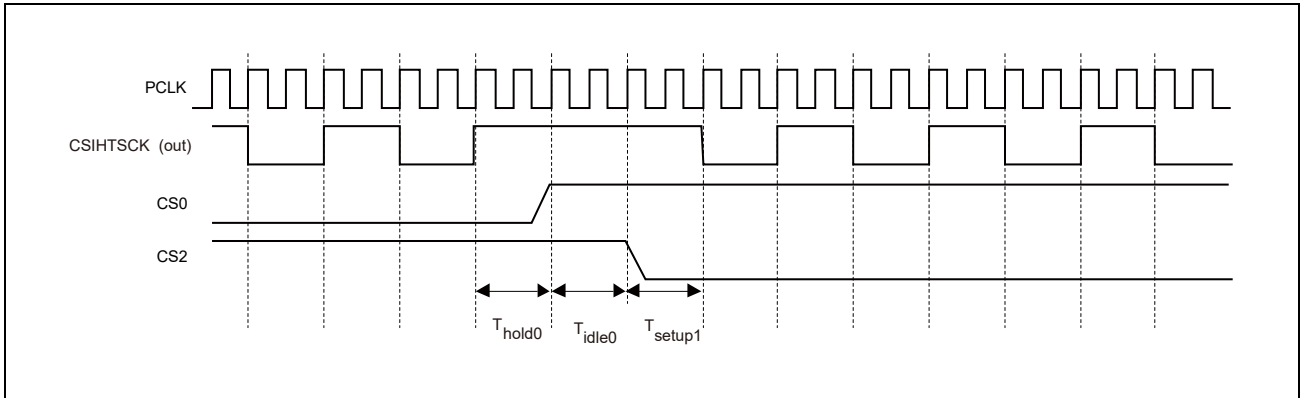


Figure 17.16 Clock Phase Timing with PCLK/8, T_{hold0} = T_{setup1} = 0.5CSIHTSCK, T_{idle0} = 0.5CSIHTSCK, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG2.CSIHnCKP2 = 0 (CSIHTCSS2)

17.5.4.2 Changing the Data Phase

The CSIHnCFGx.CSIHnDAPx bit defines the phase of the data bits relative to the clock.

The relation between the setting of the CSIHnCFGx.CSIHnDAPx bit and the hold and setup periods is as follows:

Hold time is the period from the last edge of the serial clock (CSIHTSCK) until the signals on CSIHTCSS[5:0] change to the inactive level.

Setup time is the period from when the signals on CSIHTCSS[5:0] change to the active level to when the transmission data (CSIHTSO) is output.

Therefore, there is a gap of 0.5 CSIHTSCK cycles until the edge of the serial clock signal (CSIHTSCK) is output according to the CSIHnCFGx.CSIHnDAPx setting.

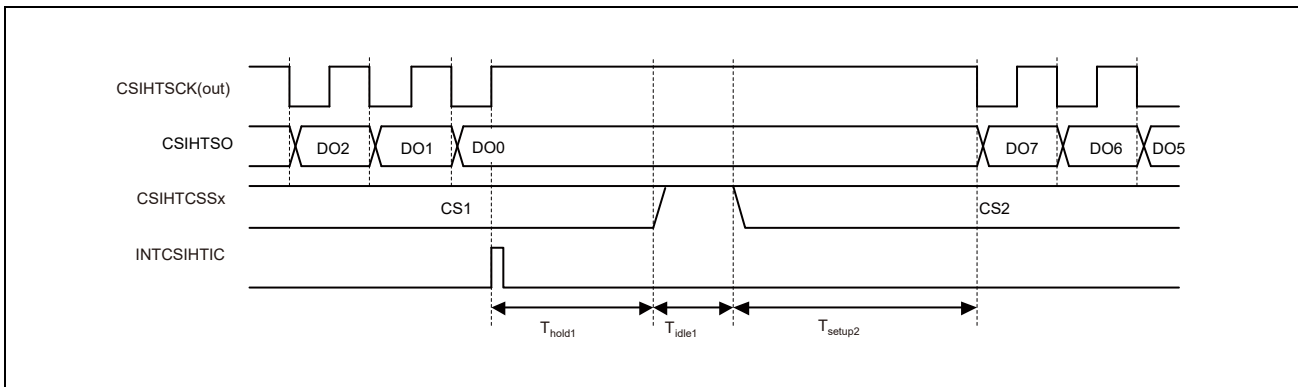


Figure 17.17 Data Phase Timing with CSIHnCFG1.CSIHnCKP1 = 0, CSIHnCFG1.CSIHnDAP1 = 0 and CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 0

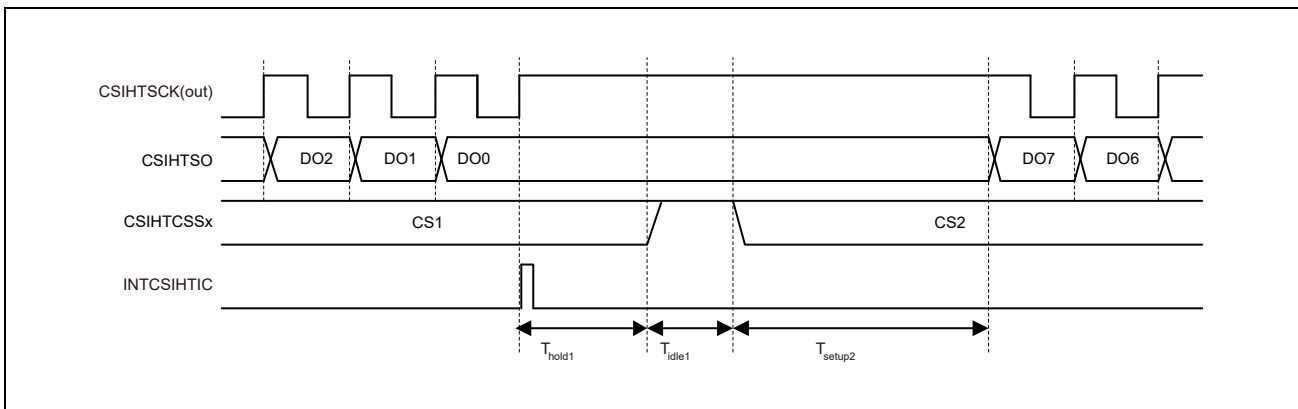


Figure 17.18 Data Phase Timing with CSIHnCFG1.CSIHnCKP1 = 1, CSIHnCFG1.CSIHnDAP1 = 0 and CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 1

17.5.5 Transmission Clock Selection

In master mode, the transfer clock frequency is selectable using the following bits:

- CSIHnCTL2.CSIHnPRS[2:0]
- CSIHnBRSy.CSIHnBRS[11:0]
- CSIHnCFGx.CSIHnBRSSx[1:0]

The transfer clock frequency of transmission clock CSIHTSCK is determined by the setting of the CSIHnCTL2.CSIHnPRS[2:0] bits and the setting of the CSIHnBRSy.CSIHnBRS[11:0] bits, but any one of CSIHnBRS3 to CSIHnBRS0 can be selected for each chip select signal by using the CSIHnCFGx.CSIHnBRSSx[1:0] bits.

The following figure shows a block diagram of the baud rate generator.

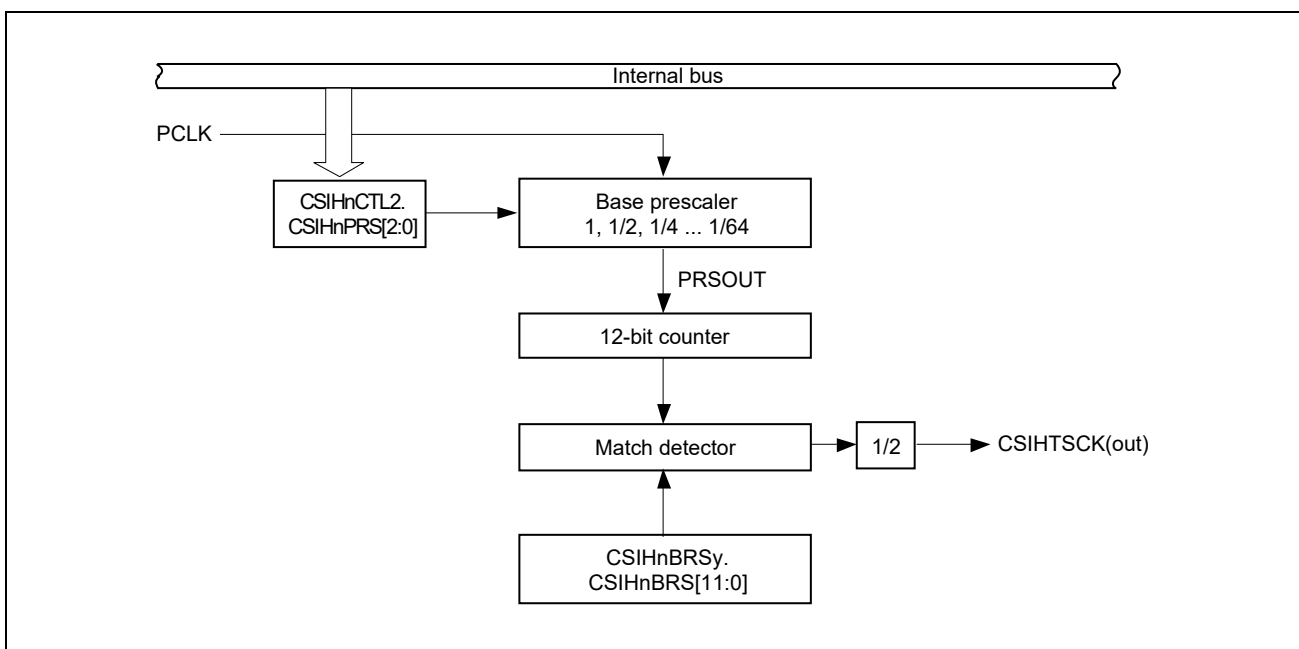


Figure 17.19 Baud Rate Generator Block Diagram

By setting CSIHnBRSy.CSIHnBRS[11:0] to 000_H, the baud rate generator is disabled and all CSIHTSCK are stopped.

17.5.5.1 Transfer Clock Frequency Calculation

The transfer clock frequency in master mode is calculated as:

$$\text{Transfer clock frequency (CSIHTSCK)} = \text{PCLK} / (\text{division ratio of PCLK}) = \text{PCLK} / (2^\alpha \times k \times 2),$$

where:

$$\alpha = \text{CSIHnCTL2.CSIHnPRS}[2:0] = 0 \text{ to } 6$$

$$k = \text{CSIHnBRS0.CSIHnBRS}[11:0] = 1 \text{ to } 4095 \text{ (when CSIHnCFGx.CSIHnBRSSx}[1:0] = 00_{\text{B}})$$

$$\text{CSIHnBRS1.CSIHnBRS}[11:0] = 1 \text{ to } 4095 \text{ (when CSIHnCFGx.CSIHnBRSSx}[1:0] = 01_{\text{B}})$$

$$\text{CSIHnBRS2.CSIHnBRS}[11:0] = 1 \text{ to } 4095 \text{ (when CSIHnCFGx.CSIHnBRSSx}[1:0] = 10_{\text{B}})$$

$$\text{CSIHnBRS3.CSIHnBRS}[11:0] = 1 \text{ to } 4095 \text{ (when CSIHnCFGx.CSIHnBRSSx}[1:0] = 11_{\text{B}})$$

17.5.5.2 Transfer Clock Frequency Upper and Lower Limits

When setting the transfer clock frequency, please note the following.

- The minimum transfer clock frequency in master mode and slave mode is PCLK/524160.
- The maximum transfer clock frequency is as follows:
 - In master mode: 10.0 MHz (however, it must be up to PCLK/8)
 - In slave mode: 4.0 MHz (however, it must be up to PCLK/20)

17.5.6 Data Transfer Modes

17.5.6.1 Transmit-Only Mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 0 puts the CSIH in transmit-only mode.

- In case of direct access mode, transmission starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.

17.5.6.2 Receive-Only Mode

Setting CSIHnCTL0.CSIHnTXE = 0 and CSIHnCTL0.CSIHnRXE = 1 puts the CSIH in receive-only mode.

- In case of direct access mode, reception starts when dummy data is written to the CSIHnTX0W or CSIHnTX0H register.

In slave mode, reception starts as soon as the CSIHTSCK transmission clock is received from the master. It is not necessary to write data to the CSIHnTX0W or CSIHnTX0H register of the slave.

17.5.6.3 Transmit/Receive Mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 1 puts the CSIH in transmit/receive mode.

- In case of direct access mode, communication starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.

17.5.6.4 Summary

The following table summarizes this section. It shows how data transfer is started in the various memory, operating, and transfer modes.

Table 17.43 Start of Data Transfer

Operating Mode		Transfer Mode	
		Transmit-Only Transmit/Receive	Receive-Only
Direct access	Master	Writing to the CSIHnTX0W register or the CSIHnTX0H register	Writing to the CSIHnTX0W register or the CSIHnTX0H register
	Slave	Clock reception from master	Incoming clock from master

17.5.7 Data Length Selection

17.5.7.1 Data Length from 2 to 16 Bits

The length of a data packet is selectable for each chip select signal from 2 to 16 bits using $\text{CSIHnCFGx.CSIHnDLSx}[3:0]$. The examples below show the communication with MSB first ($\text{CSIHnCFGx.CSIHnDIRx} = 0$).

Data length = 16 bits ($\text{CSIHnCFGx.CSIHnDLSx}[3:0] = 0000_{\text{B}}$):

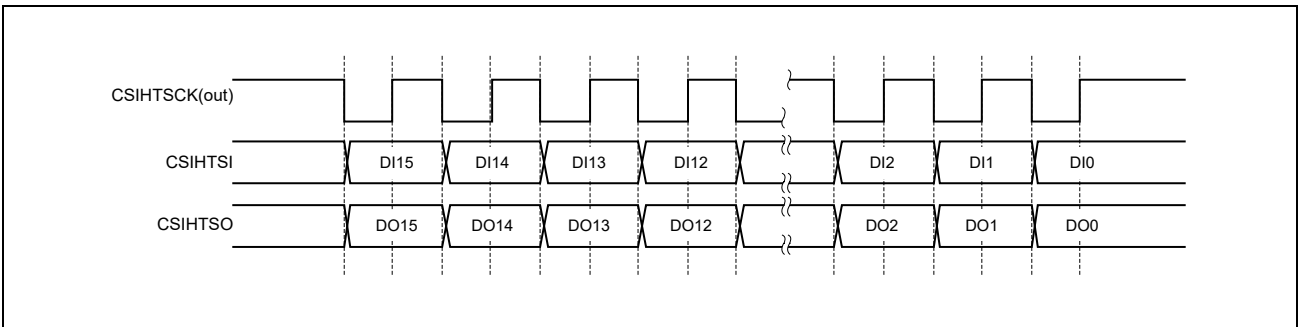


Figure 17.20 16 Bit Data Length, MSB First

Data length = 14 bits ($\text{CSIHnCFGx.CSIHnDLSx}[3:0] = 1110_{\text{B}}$):

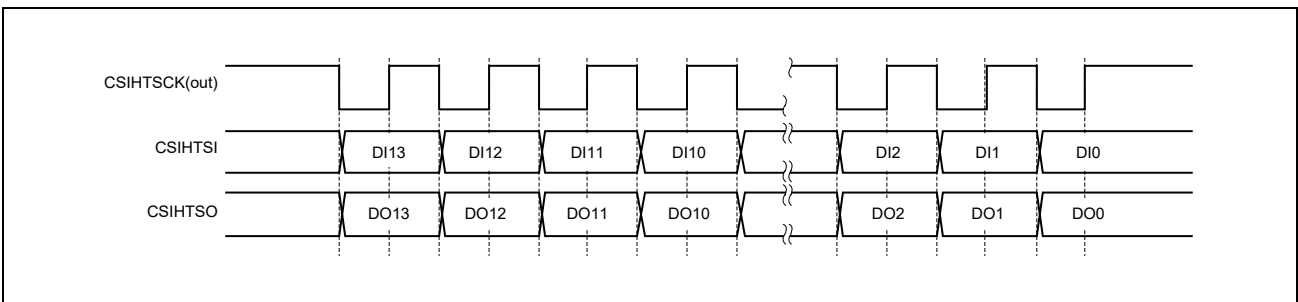


Figure 17.21 14 Bit Data Length, MSB First

17.5.7.2 Data Length Greater than 16 Bits

If the data to be sent/received exceeds 16 bits, the extended data length (EDL) function can be used.

EDL function is enabled by setting the CSIHnCTL1.CSIHnEDLE bit to 1.

EDL function works as follows:

- The data has to be broken into 16-bit blocks plus remainder. For example, data of 42 bits would be broken into two 16-bit blocks plus 10 bits.
- The bit length of the remainder is set as “data length” in CSIHnCFGx.CSIHnDLSx[3:0].
- For transmitting the 16-bit blocks, CSIHnTX0W.CSIHnEDL must be set to 1. In this case, the data written to CSIHnTX0W is sent as a 16-bit data length regardless of the CSIHnCFGx.CSIHnDLSx[3:0] bit setting.
- The transfer is complete after a block with the specified data length (the remainder of data specified with CSIHnTX0W.CSIHnEDL = 0) has been sent.

Example

Example of transmitting 40-bit data (123456789A_H) to CS0:

40 bits are split into two blocks of 16-bit blocks and 8 bits.

- Initialize to CSIHnCFG0.CSIHnDLS0[3:0] = 8.
- To transmit 123456789A_H with MSB first, write the following sequence to CSIHnTX0W:
 - 20FE 1234_H (CSIHnTX0W.CSIHnEDL = 1)
 - 20FE 5678_H (CSIHnTX0W.CSIHnEDL = 1)
 - 00FE 009A_H (CSIHnTX0W.CSIHnEDL = 0)

The following figure illustrates the timing.

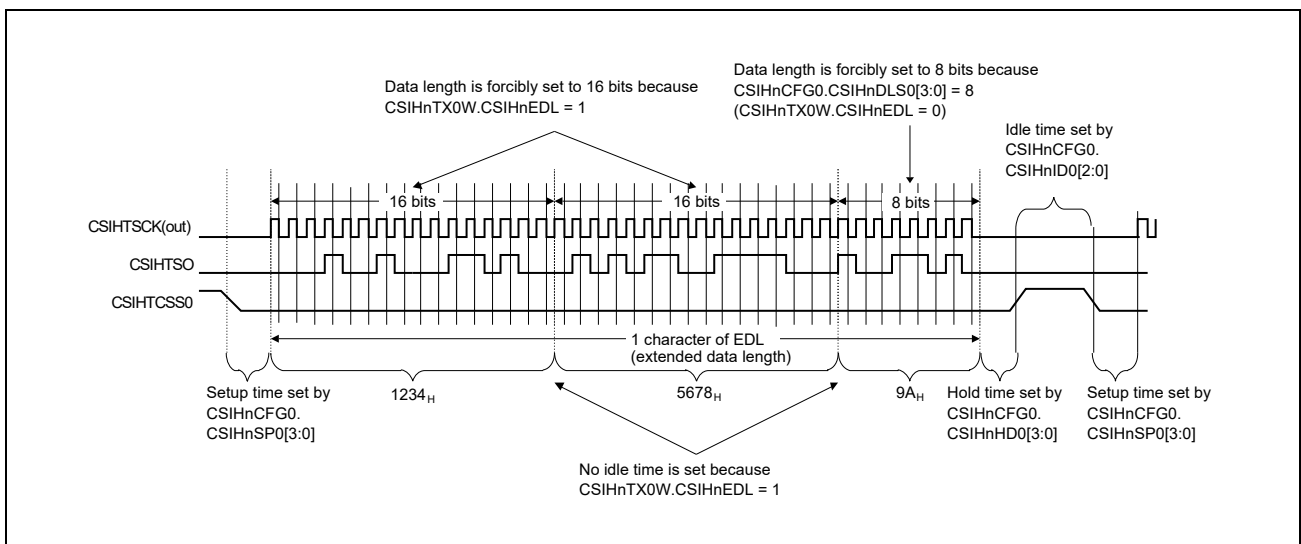


Figure 17.22 EDL Timing Diagram

NOTES

1. 1-bit data length is allowed only when using the EDL mode.
 2. If parity is enabled, the parity bit is added after the last bit.
 3. When data is sent using extended data length (EDL) function, use the same chip select signal.
 4. Example for configuring the data direction:
 - Data to be sent: 123456_H
 - MSB first:
 - Set CSIHnCFGx.CSIHnDIRx = 0
 - Write CSIHnTX0W = 20FE 1234_H (EDL bit = 1)
 - Write CSIHnTX0W = 00FE 0056_H (EDL bit = 0)
 - LSB first:
 - Set CSIHnCFGx.CSIHnDIRx = 1
 - Write CSIHnTX0W = 20FE 3456_H (EDL bit = 1)
 - Write CSIHnTX0W = 00FE 0012_H (EDL bit = 0)
 5. Operation is not guaranteed if CSIHnTX0W.CSIHnEOJ and CSIHnTX0W.CSIHnEDL are simultaneously set to "1" while CSIHnCTL1.CSIHnJE = 1 and CSIHnCTL1.CSIHnEDLE = 1.
 6. EDL mode cannot be used in receive-only mode of slave mode.
(CSIHnCTL2.CSIHnPRS[2:0] = 111_B, CSIHnCTL0.CSIHnTXE = 0, CSIHnCTL0.CSIHnRXE = 1)
-

17.5.8 Serial Data Direction Selection

The serial data direction is selectable for each chip select signal using the CSIHnDIRx bit in the CSIHnCFGx register.

The examples below show communication for a data length of 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).

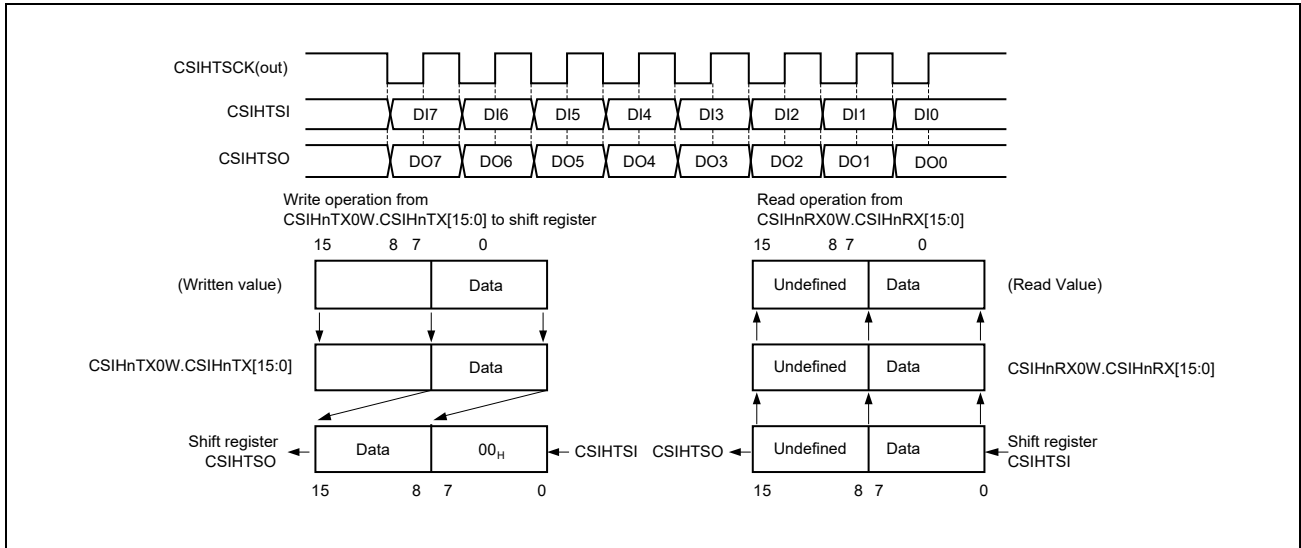


Figure 17.23 Serial Data Direction Select Function - MSB First (CSIHnDIRx = 0)

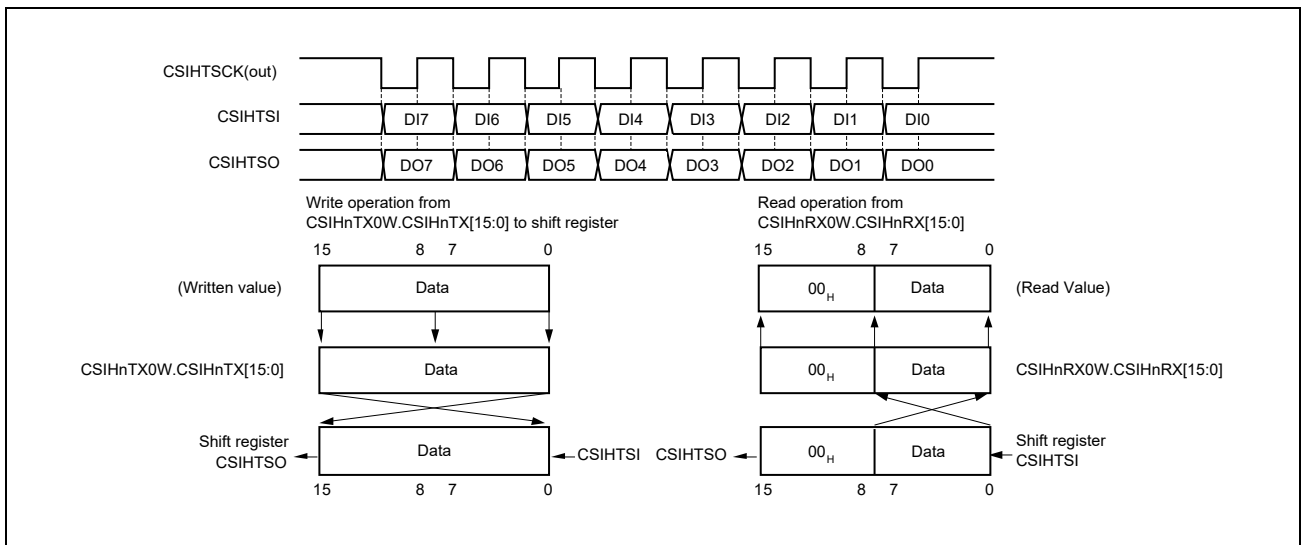


Figure 17.24 Serial Data Direction Select Function - LSB First (CSIHnDIRx = 1)

17.5.9 Slave Select (SS) Function

The Slave Select (SS) function enables communication between one master and multiple slaves.

In master mode, the master device outputs the slave select signal ($\overline{\text{CSIHTCSSx}}$) to a slave. Communication by a device in slave mode is enabled when the slave input select signal ($\overline{\text{CSIHTSSI}}$) is at the low level.

See the **Section 17.5.2, Master/Slave Connections**, for examples of connections using the SS function.

17.5.9.1 Communication Timing Using SS Function

The following figure illustrates the communication signal and timings using the SS function.

In slave mode, the data transfer configuration is determined by the CSIHnCFG0 register.

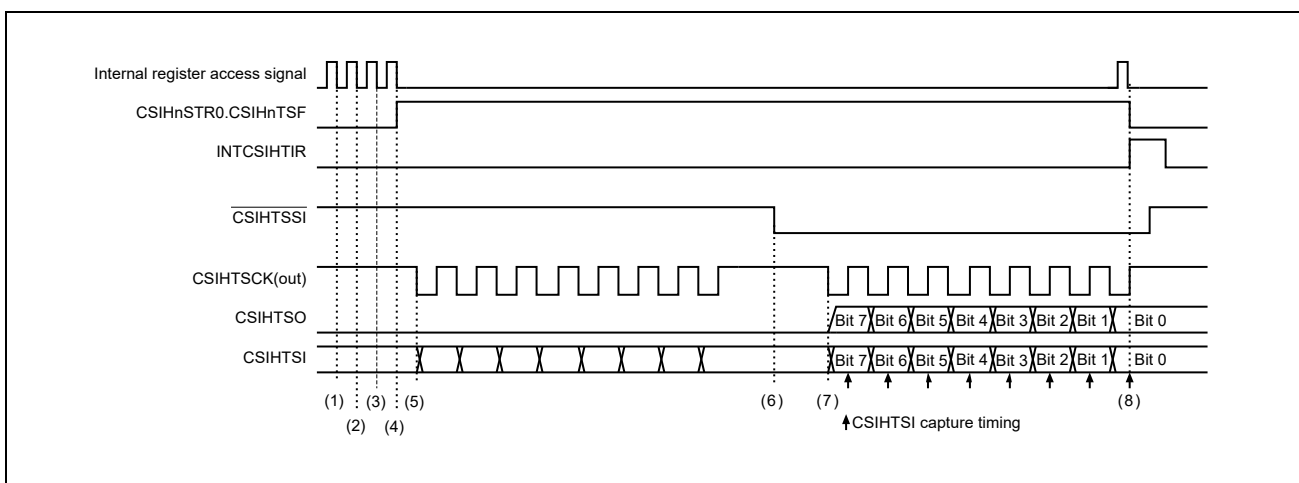


Figure 17.25 Transmission/Reception Timing of Communication Using SS Function

- (1) CSIH enters slave mode by setting $\text{CSIHnCTL2.CSIHnPRS}[2:0] = 111_{\text{B}}$. $\text{CSIHnCFG0.CSIHnCKP0}$ and $\text{CSIHnCFG0.CSIHnDAP0}$ are 0.
- (2) The data length is 8 bits ($\text{CSIHnCFG0.CSIHnDLS0}[3:0] = 1000_{\text{B}}$). The data direction is MSB first ($\text{CSIHnCFG0.CSIHnDIR0} = 0$).
- (3) The transmit/receive mode is set ($\text{CSIHnCTL0.CSIHnTXE} = 1$, $\text{CSIHnCTL0.CSIHnRXE} = 1$, and $\text{CSIHnCTL0.CSIHnPWR} = 1$). Communication start is permitted.
- (4) The transfer status flag $\text{CSIHnSTR0.CSIHnTSF}$ is automatically set when transfer data is written to the CSIHnTX0W or CSIHnTX0H transmission register during direct access mode.
- (5) As long as signal $\overline{\text{CSIHTSSI}}$ is at the high level, transmission/reception is not started, even if an external transmission clock CSIHTSCK is input. Input to CSIHTSI is ignored.
- (6) $\overline{\text{CSIHTSSI}}$ falling to low level indicates that CSIHTSO is enabled and ready for transmission.
- (7) As soon as the external clock signal CSIHTSCK is detected, the slave transmits data to CSIHTSO and simultaneously captures data from CSIHTSI .
- (8) Interrupt INTCSIHTIR indicates that the reception is complete. The CSIHnRX0W/H register can be read.

17.5.9.2 CSIHTSSO Operation

CSIHnPWR	CSIHnTXE	CSIHnRXE	CSIHnSSE	CSIHTSSO
0	—	—	—	H
1	—	—	0	H
	0		1	H
	1		1	Reversed value of $\overline{\text{CSIHTSSI}}$ level

The CSIHTSSO pin is a signal to control the I/O function of the chip's SO pin when using the SS function.

The CSIHTSO pin is enabled when the CSIHTSSO pin is "High" (the chip's SO pin is being driven).

The CSIHTSO pin is disabled when the CSIHTSSO pin is "Low" (the chip's SO pin is not being driven).

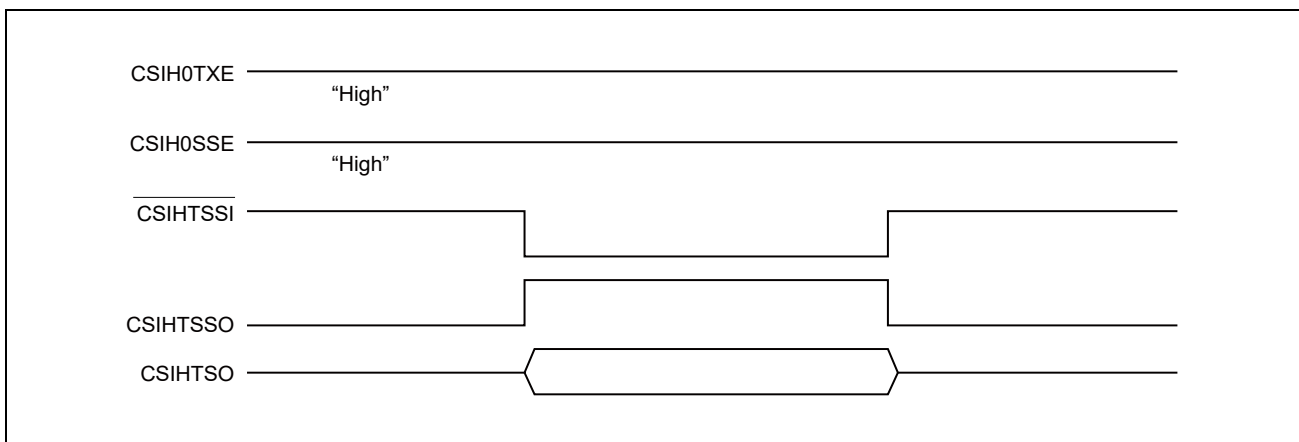


Figure 17.26 Operation of CSIHTSSO

CAUTION

If $\overline{\text{CSIHTSSI}}$ pin is changed during communication (CSIHnSTR0.CSIHnTSF = 1), current communication is not guaranteed.

17.5.10 Handshake Function

CSIH features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by the CSIHnCTL1.CSIHnHSE bit. For handshake, the signals CSIHTRYI and CSIHTRYO are used. The busy timing depends on the data phase selection bit CSIHnCFGx.CSIHnDAPx.

17.5.10.1 Slave Mode

If CSIHnCTL1.CSIHnHSE = 1, a low-level CSIHTRYO signal is output when the slave becomes busy. This can happen in two cases:

1. When the next data to be sent is not ready:

When the slave is in transmit-only mode or transmit/receive mode (CSIHnCTL0.CSIHnTXE = 1) and is in the states listed below, the CSIHTRYO outputs the busy state (low level).

Table 17.44 Memory Mode and Slave Transfer State

Memory Mode	Slave Transfer State
Direct access mode	When there is no more data to be sent

2. When receive register is full:

When slave is set in receive-only mode or transmit/receive mode (CSIHnCTL0.CSIHnRXE = 1), and new data cannot be copied from a shift register to CSIHnRX0W/H because the previously received data is still in the CSIHnRX0W/H register (CSIHnRX0W/H is full).

When CSIHnCTL0.CSIHnRXE is 1 and is in any of the following states, CSIHTRYO outputs busy state (low level).

Table 17.45 Memory Mode and Slave Reception State

Memory Mode	Slave Reception State
Direct access mode	When CSIHnRX0W or CSIHnRX0H is full

The following examples assume an eight-bit data length.

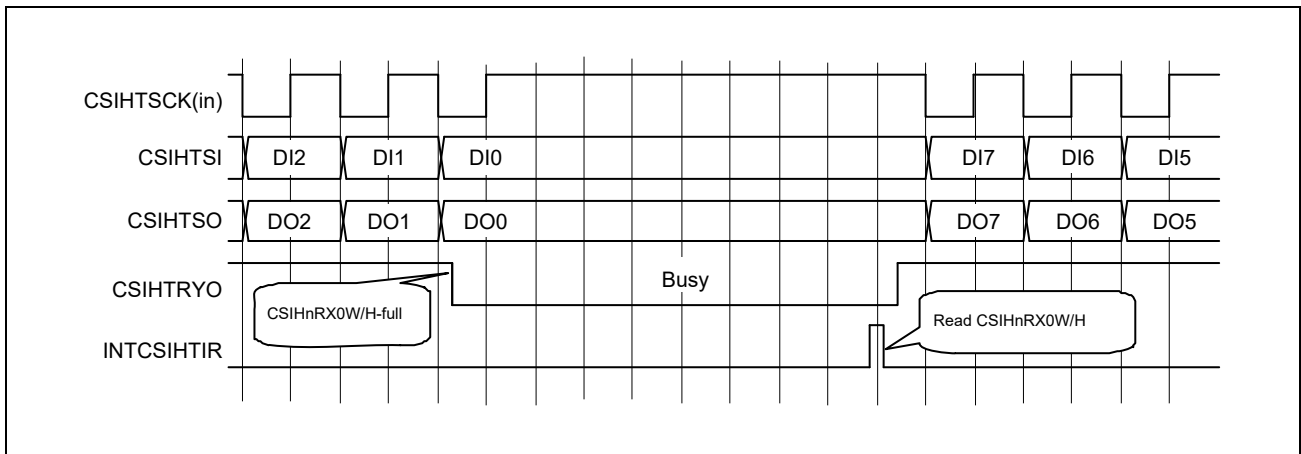


Figure 17.27 Busy Signal from the Slave (Direct Access Mode; CSIHnCFGx.CSIHnDAPx = 0)

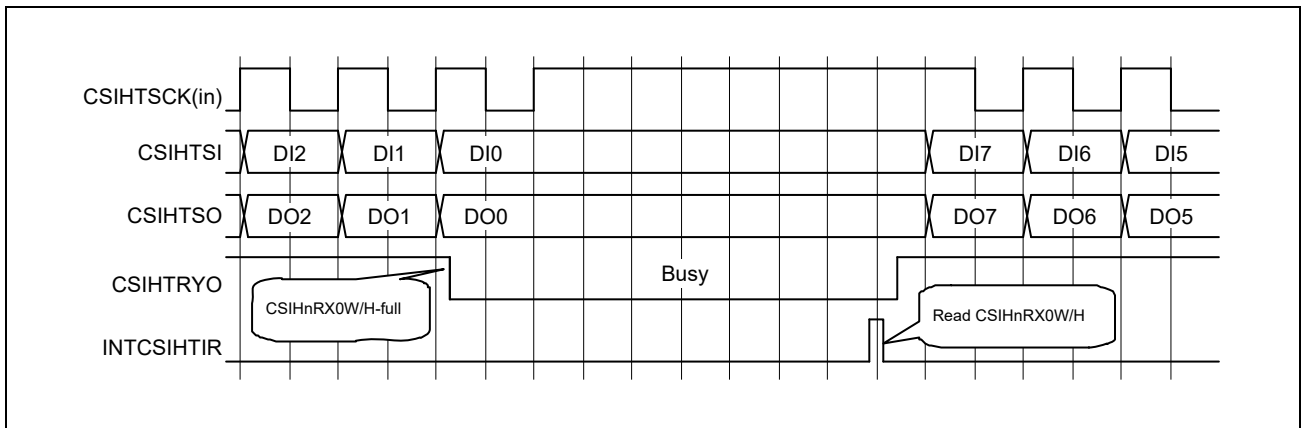


Figure 17.28 Busy Signal from the Slave (Direct Access Mode; CSIHnCFGx.CSIHnDAPx = 1)

17.5.10.2 Master Mode

When the master detects $CSIHTRYI = 0$ while $CSIHnCTL1.CSIHnHSE = 1$, the subsequent transfers are put on hold, and the master goes into wait status. It suspends the $CSIHTSCK$ clock.

The $CSIHTRYI$ level is checked at each half clock cycle of $CSIHTSCK$.

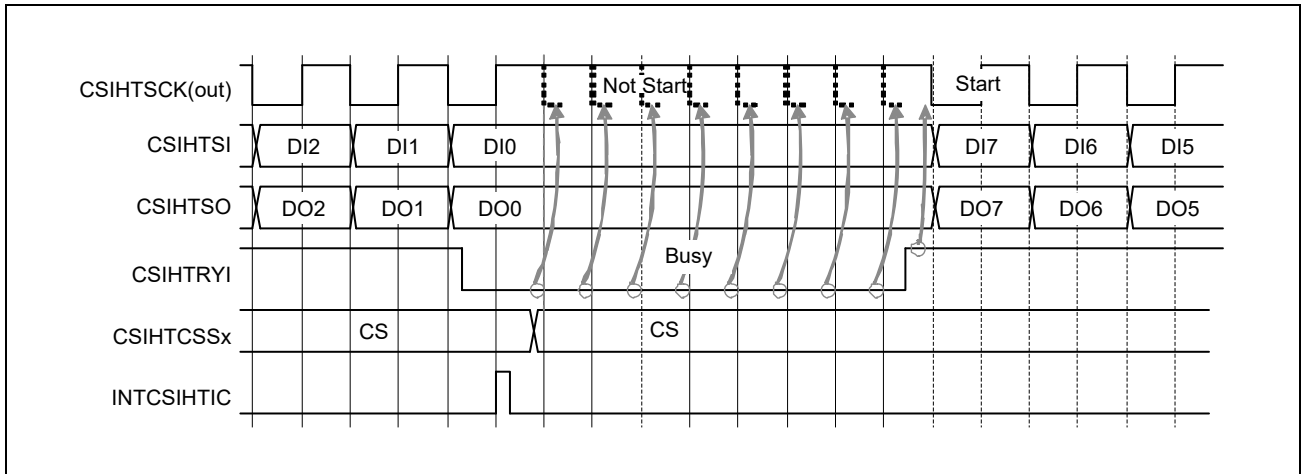


Figure 17.29 Master's Reaction to $CSIHTRYI$ ($CSIHnCFGx.CSIHnDAPx = 0$)

The $CSIHTRYI$ signal must be pulled down by the slave before the next transfer starts. If this is done while data transfer is in progress, the serial clock from the master is suspended after the transfer is complete.

The master resumes the communication as soon as $CSIHTRYI$ becomes high (the slave is "ready").

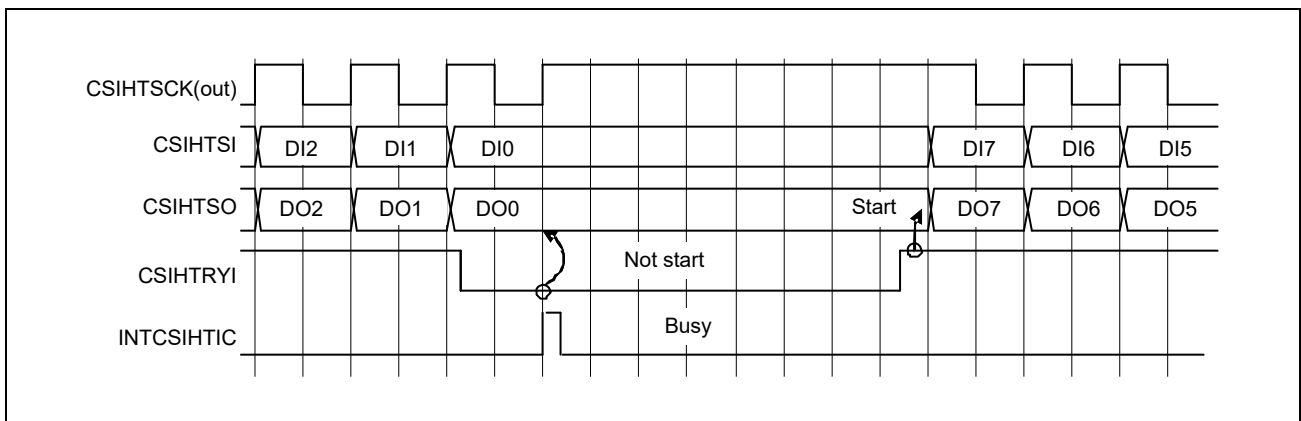


Figure 17.30 Master's Reaction on $CSIHTRYI$ ($CSIHnCFGx.CSIHnDAPx = 1$)

CAUTIONS

1. If multiple slaves are connected, the master must only detect the $CSIHTRYI$ signal of the slave it has selected for communication.
2. Even when the $CSIHTRYI$ pin of the master detects a $CSIHTRYO$ signal from the slave during data transfer, the communication is not put on hold but continues until the data transfer is completed.

17.5.11 Error Detection

CSIH can detect five error types:

- Data consistency error (transmission data)
- Parity error (received data)
- Overrun error (received data)

Check for parity, data consistency can be enabled/disabled individually.

If any of these errors is detected, the interrupt request INTCSIHTIRE is generated and the corresponding flags are set.

17.5.11.1 Data Consistency Check

The purpose of the data consistency check is to ensure that the data physically sent as output signal is identical to the original data that was copied to the shift register.

The data consistency check can be enabled/disabled by the CSIHnCTL1.CSIHnDCS bit (when checking data consistency, make sure that PIPn.PIPCn_m = 1 and PIBn.PIBCn_m = 0). It will not be enabled if data transmission is disabled (CSIHnCTL0.CSIHnTXE = 0).

When the data consistency check is active, the data transferred from CSIHnTX0W or CSIHnTX0H to the shift register is copied to a separate register. In addition, the physical levels of CSIHTSO are read back via the CSIHTDCS signal into the corresponding shift register.

After completion of the transmission, the sent data is compared with the original transmission data.

Mismatch is considered as a data consistency error and:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnDCE bit is set.

Additionally, CSIHnRX0W.CSIHnTDCE of data that contains the error is set.

The data consistency check function is illustrated in the following block diagram.

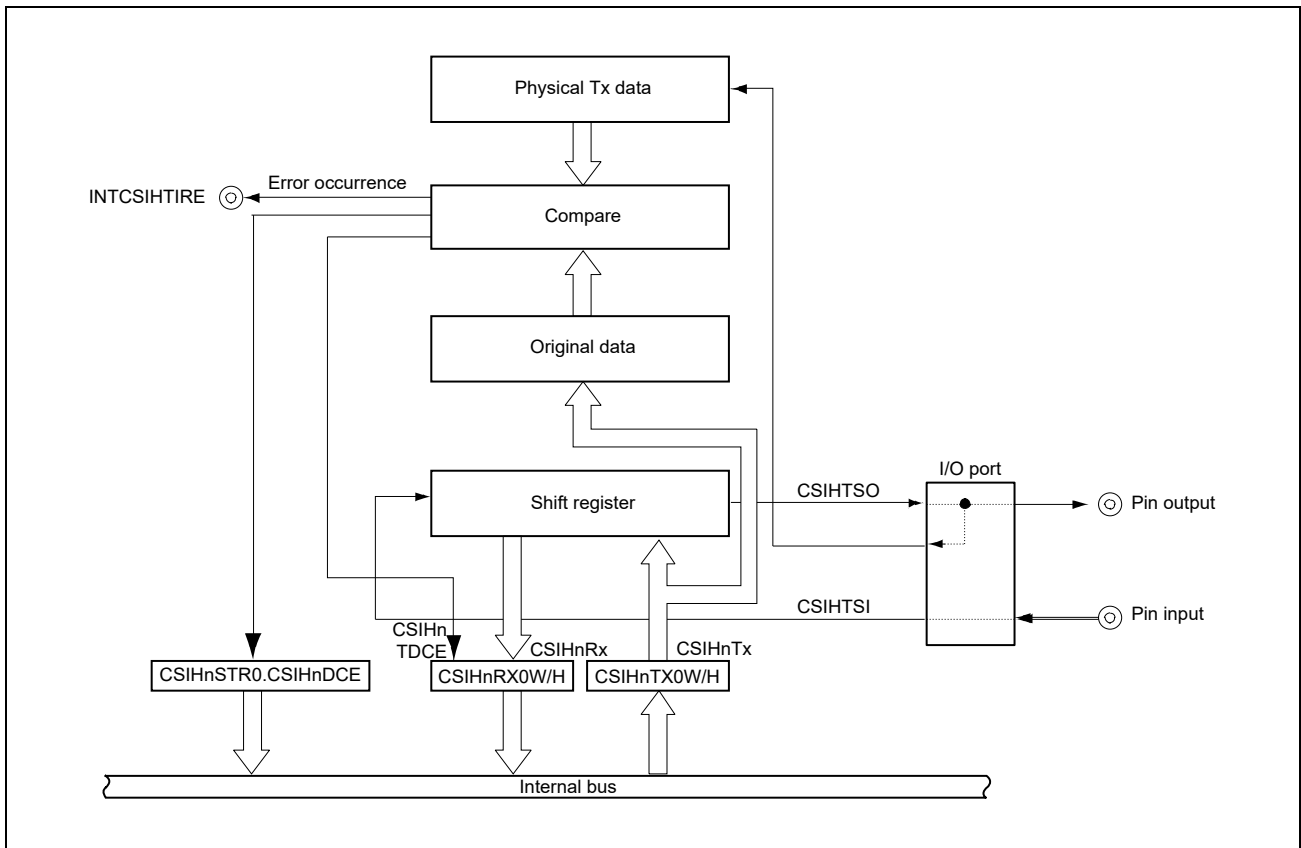


Figure 17.31 Block Diagram of Data Consistency Check Function

17.5.11.2 Parity Check

CSIH can append a parity bit to the last data bit (even if extended data length is used).

The use and type of parity is specified in `CSIHnCFGx.CSIHnPSx[1:0]`.

Parity check is enabled if `CSIHnCFGx.CSIHnPSx[1] = 1`.

The parity bit is checked after a reception is complete. If a parity error occurs, the following happen::

- Interrupt `INTCSIHTIRE` is generated.
- The `CSIHnSTR0.CSIHnPE` bit is set.

Additionally, `CSIHnRX0W.CSIHnRPE` of data that contains the error is set.

The figure below shows an example.

- Data length is 8 bits.
- The data to be transmitted is `05H` and `35H`.
- Data direction is LSB first.
- Parity type is odd.

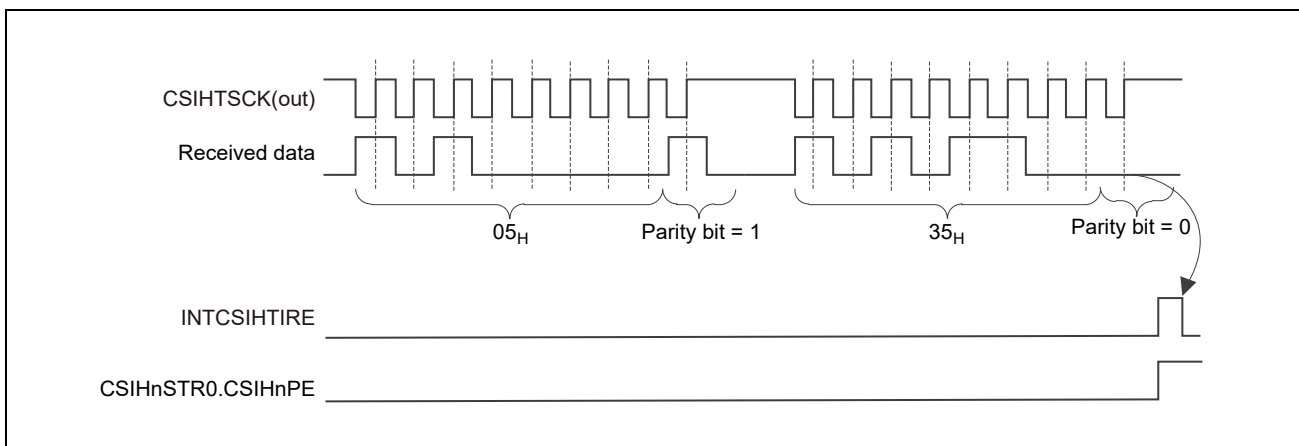


Figure 17.32 Parity Check Example

The parity bit of the first data is 1. There is no parity error, because the total number of ones (including the parity bit) is odd.

The parity bit of the second data is 0. This is detected as a parity error, because the total number of ones (including the parity bit) is even.

If the EDL (extended data length) function is used, the parity bit is added after the last bit of the data.

17.5.11.3 Overrun Error

An overrun error may occur in direct access mode. The overrun error does not occur if data reception is disabled (CSIHnCTL0.CSIHnRXE = 0).

There are one conditions for overrun errors.

Condition for Errors 1

- In slave mode, when CSHnCTL1.CSIHnHSE = 0 (handshake function disabled):
 - In direct access mode, when reception is completed while the previously received data is remains in the CSHnRX0W/H register.

(1) Direct Access

In direct access modes, this error occurs when newly received data cannot be transferred from the shift register to the reception register CSHnRX0W/H. This happens when CSHnRX0W/H was not read and therefore contains previously received data.

The following figure illustrates the overrun error detection function.

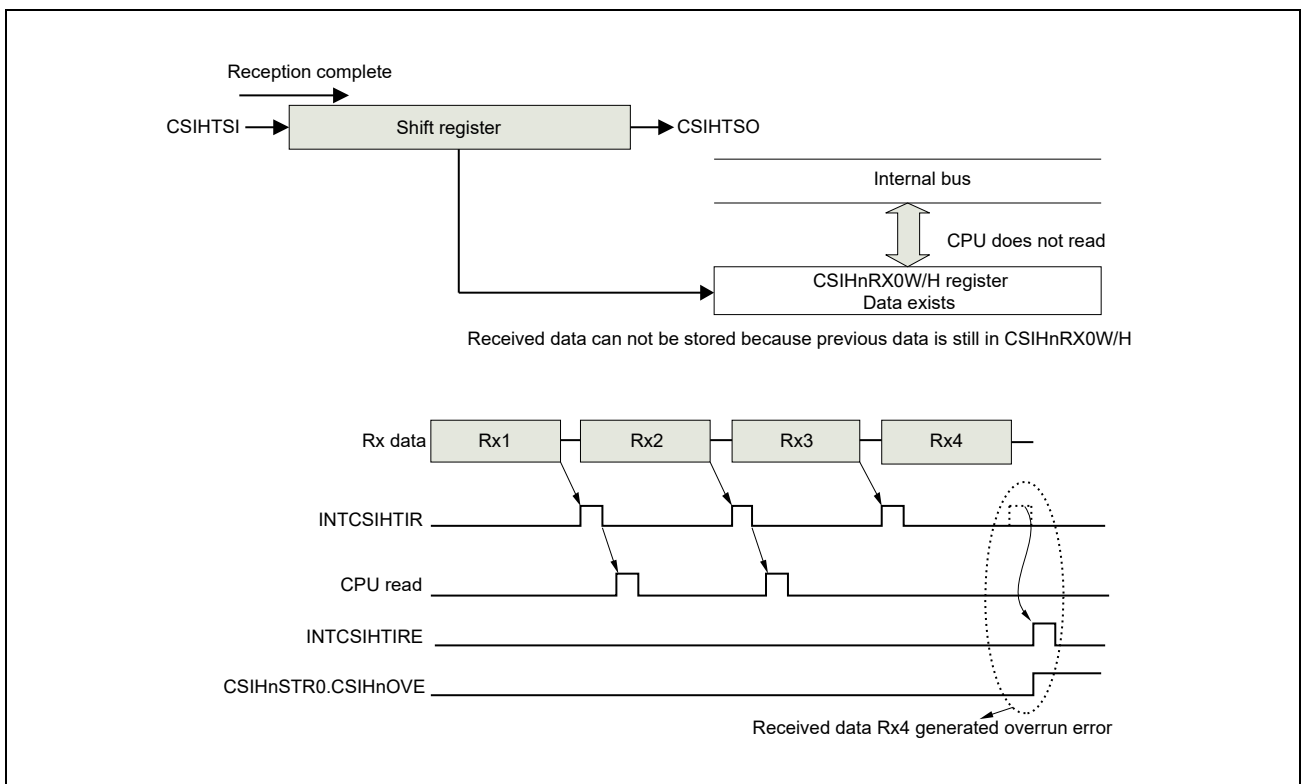


Figure 17.33 Overrun Error Detection in Direct Access Modes

NOTE

An overrun error can be avoided in slave mode by using handshake function. When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver reads its reception register and becomes ready again.

17.5.12 Loop-Back Mode

Loop-back mode is a special mode for self-test. This feature is only available in master mode.

When this mode is active (CSIHnCTL1.CSIHnLBM = 1), CSIHnCSSx is fixed to the inactive level (the active level is defined by the CSIHnCTL1.CSIH0CSLx value). The transmit and receive signals are internally connected, as shown in the figures below. The signals CSIHnTSCK, CSIHnTSO, CSIHnTSI, and CSIHnCSSx are disconnected from the ports. In addition, the CSIHnTSO output level is fixed to low, and CSIHnTSCK is set to reset level (High) regardless of the value of the CSIHnCFGx.CSIHnCKPx. The rest of CSIH works as in normal operation.

In order to test CSIH, put it in loop-back mode and carry out normal transfer operations. Then check that the received data is the same as the transmitted data. Any connected device remains unaffected by the loop-back test.

Table 17.46 Pin Output Level in Loop-Back Mode

Pin Name	Output level
CSIHnTSCK(out)	High level
CSIHnCSS[5:0]	Inactive level
CSIHnTSO	Low level (not dependent on the previous value)
Interrupt	Normal function
CSIHnTRYO	Normal function (Low level)

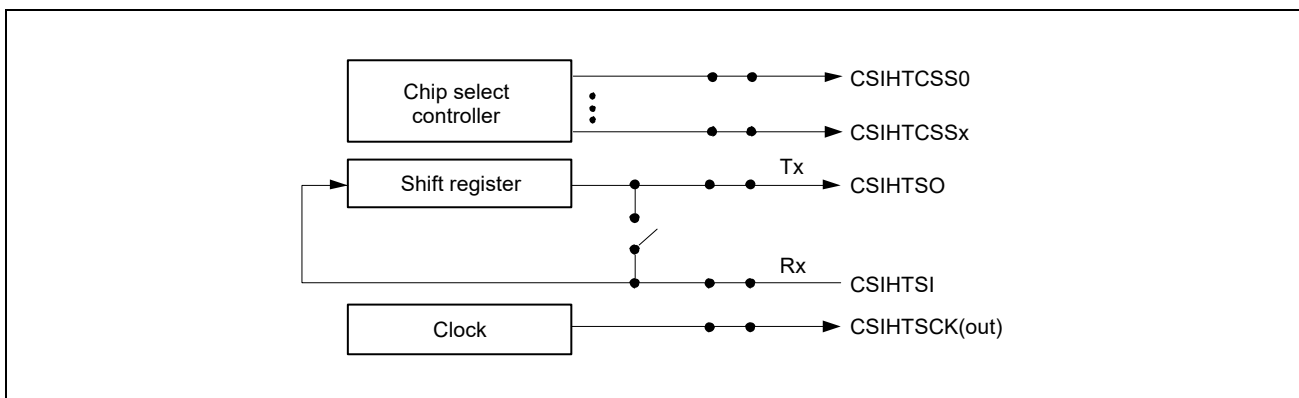


Figure 17.34 Normal Operation

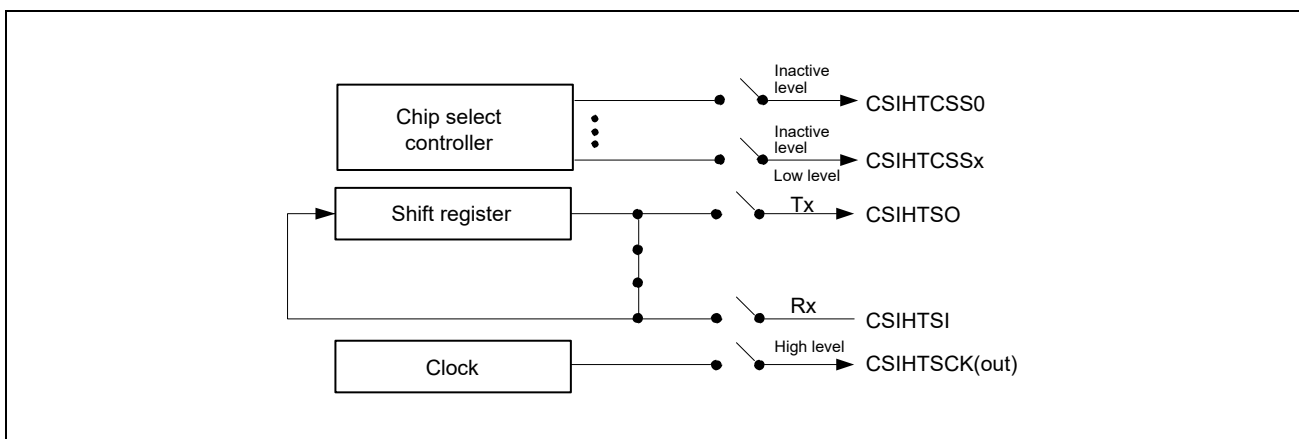


Figure 17.35 Loop-Back Mode Operation

17.5.13 Enforced Chip Select Idle Setting

It is possible to insert an idle state between two consecutive data transfers by setting $\text{CSIHnCFGx.CSIHnIDLx}$.

1. When $\text{CSIHnCFGx.CSIHnIDLx} = 0$

If the next CSIHTCSSx is the same as the previous one, an idle state is not inserted and an inter-data time is inserted.

If the next CSIHTCSSx is different from the previous one, an idle state is inserted.

2. When $\text{CSIHnCFGx.CSIHnIDLx} = 1$

An idle state is always inserted even if a next CSIHTCSSx is not different from the previous one.

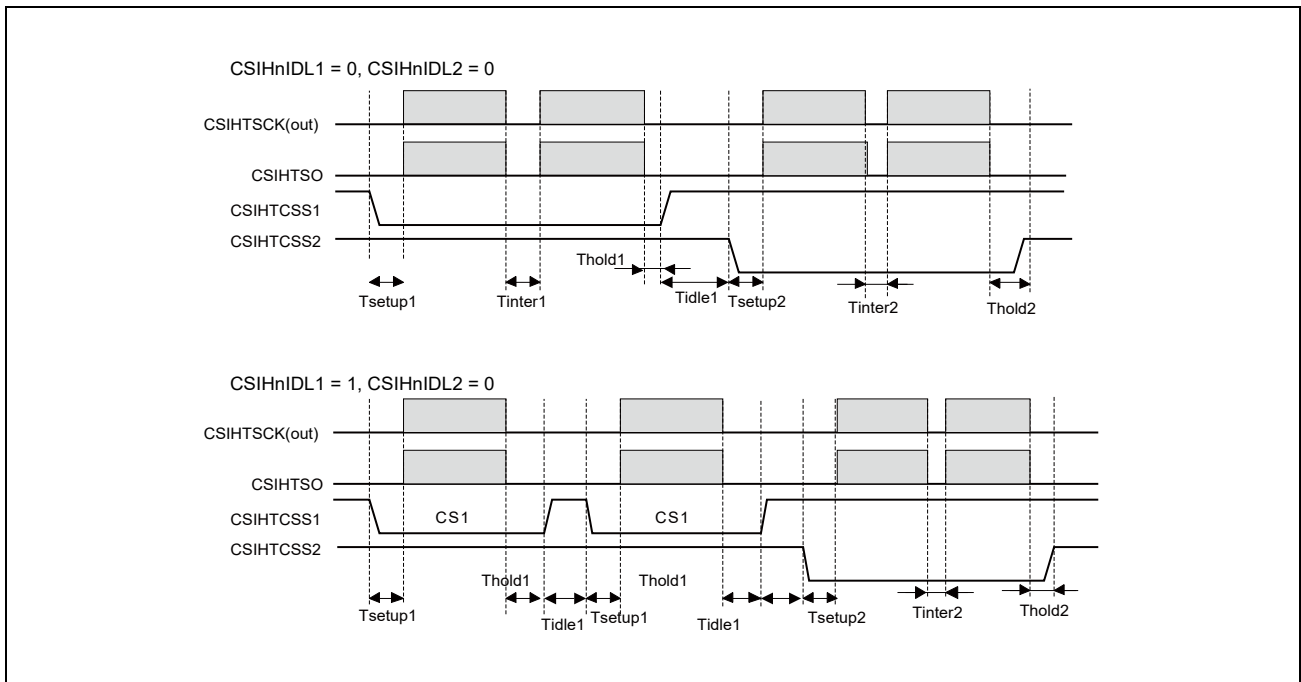


Figure 17.36 Enforced Chip Select Setting Idle Example

17.5.14 Extended Communication Function

CSIH has an extended communication function that can perform 8-bit, 16-bit, or 32-bit CSI (SPI) communication by using up to 16 stages of extended communication data registers.

When the CSIHnECTL0.CSIHnEEN bit is set to 1, the extended communication function is enabled and can perform extended communication based on a specified data size. A data transfer direction can be specified (LSB-first or MSB-first), and the extended communication can be stopped.

When the CSIHnECTL0.CSIHnEEN bit is 0, the extended communication function is disabled and CSIH operates in the same way as the original function.

17.5.14.1 Data Transfer Direction Selection Function

CSIH enables the selection of the data transfer direction by using the CSIHnECTL1.CSIHnEDIR bit when the extended communication function is enabled.

The figures listed in the following table show the data transfer operations with MSB-first transfer or LSB-first transfer selected as the data transfer direction.

Table 17.47 Selection of Data Transfer Direction

Figure No.	Content	Register Setting	
		CSIHnECTL1.CSIHnEDS[1:0]	CSIHnECTRL1.CSIHnEDIR
Figure 17.37	Data size of 32 bits, MSB-first transfer	10 _B	0
Figure 17.38	Data size of 32 bits, LSB-first transfer	10 _B	1
Figure 17.39	Data size of 8 or 16 bits, MSB-first transfer	00 _B or 01 _B	0
Figure 17.40	Data size of 8 or 16 bits, LSB-first transfer	00 _B or 01 _B	1

NOTE

To select the transfer direction when the extended communication function is enabled, use the CSIHnECTL1.CSIHnEDIR bit for the extended communication function, and set the CSIHnCFGx.CSIHnDIRx bit (which is used to select the transfer direction for communication by CSIH) to 0 (MSB-first). If a transfer direction is selected for both the extended communication function and CSIH, the actual transfer direction is not guaranteed.

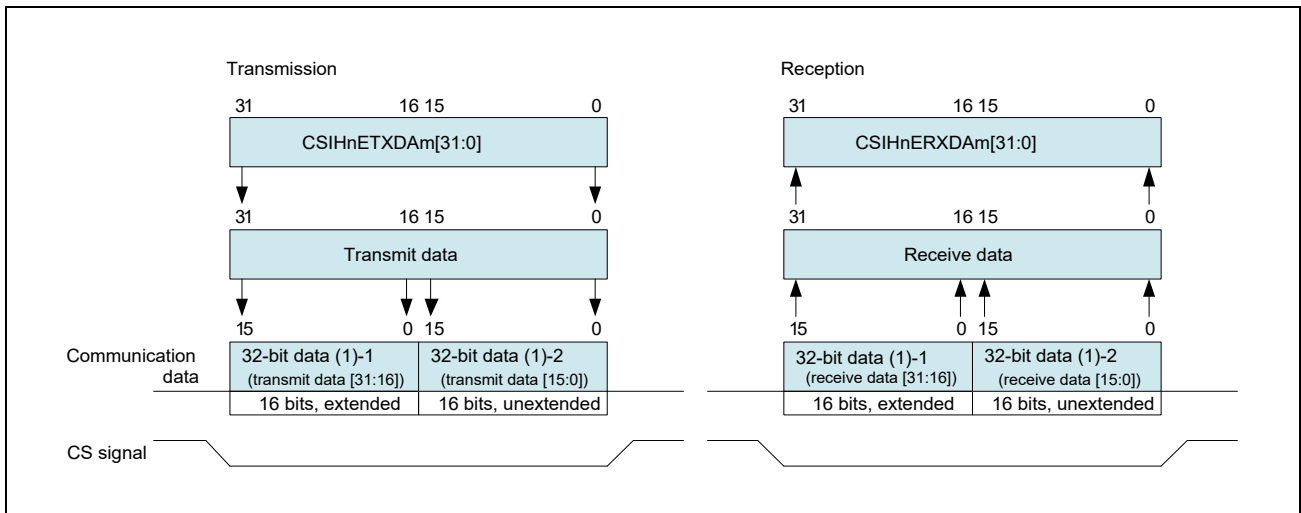


Figure 17.37 Operations with Data Size Set to 32 Bits and Transfer Direction Set to MSB-First

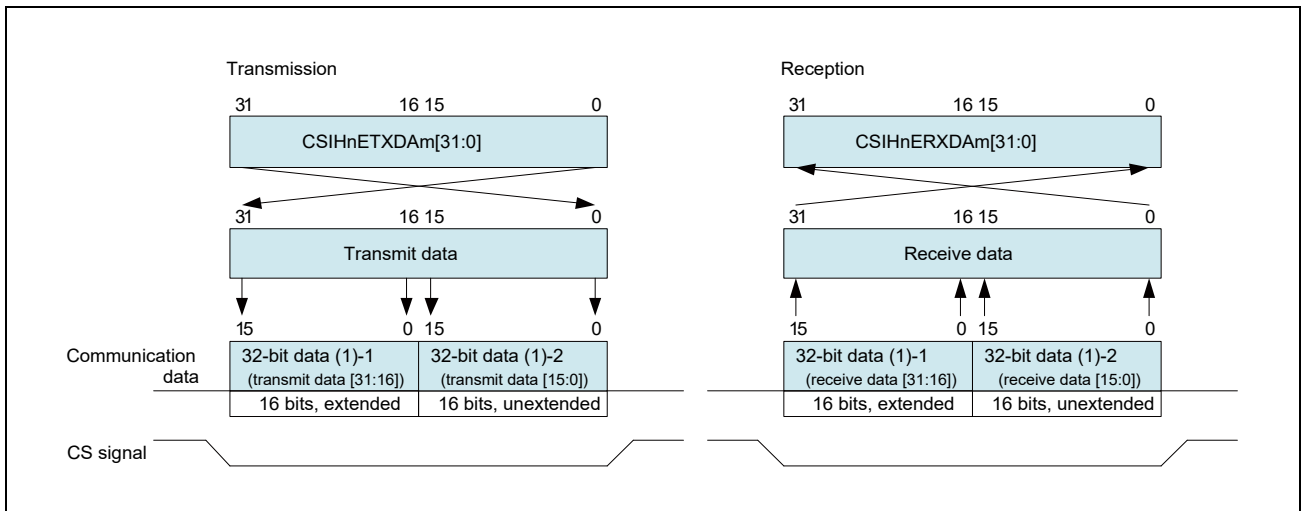


Figure 17.38 Operations with Data Size Set to 32 Bits and Transfer Direction Set to LSB-First

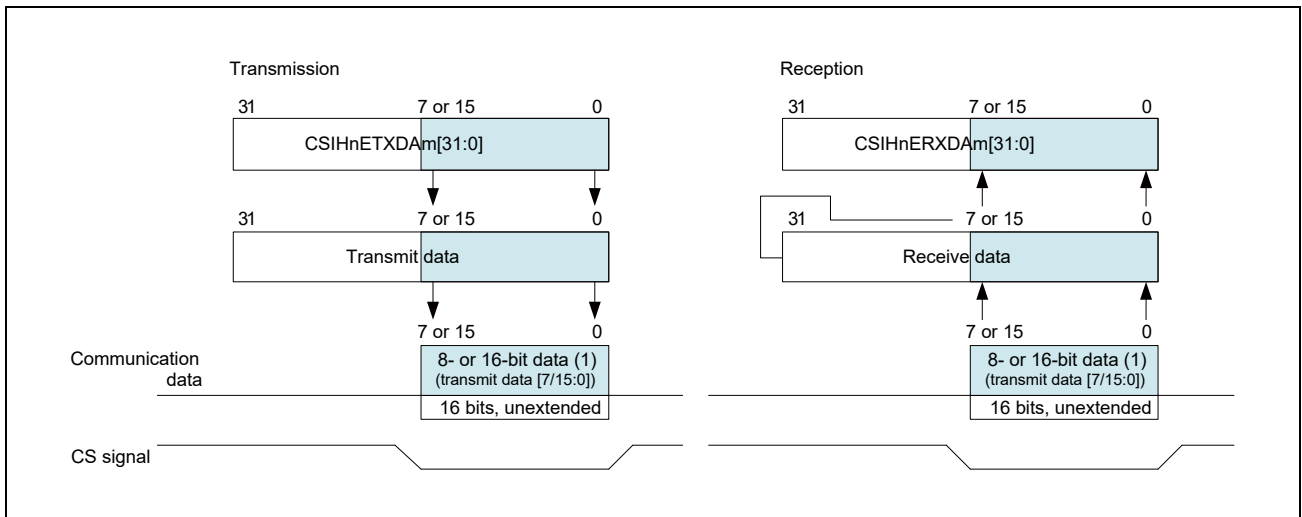


Figure 17.39 Operations with Data Size Set to 8 or 16 Bits and Transfer Direction Set to MSB-First

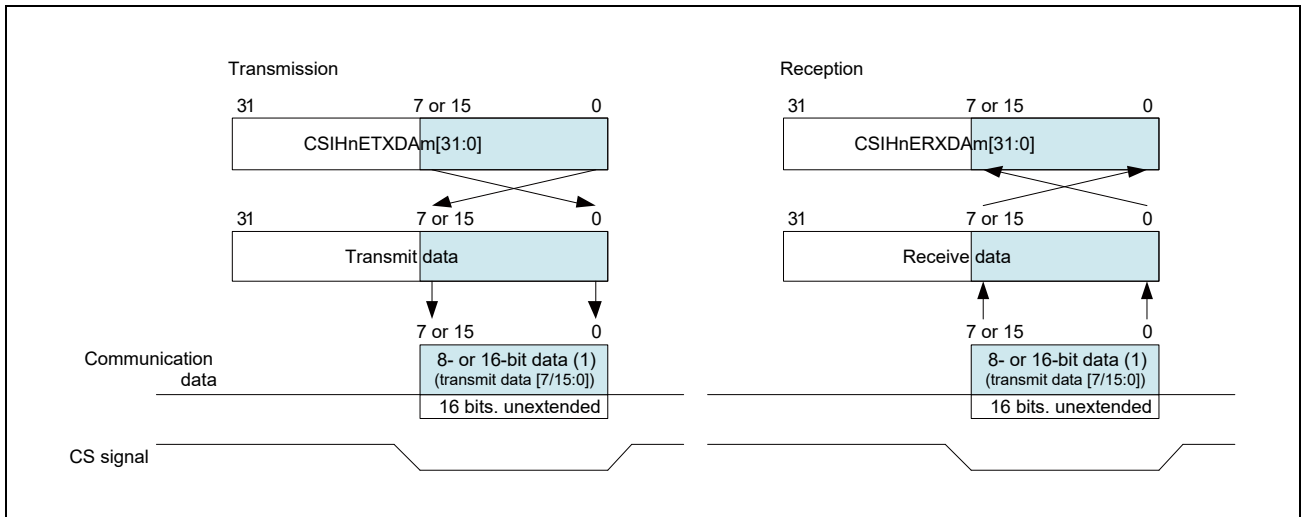


Figure 17.40 Operations with Data Size Set to 8 or 16 Bits and Transfer Direction Set to LSB-First

17.5.14.2 Extended Communication Stop Function

CSIH enables the stopping of extended communication by using the stop trigger bit (CSIHnESTT.CSIHnESTRG).

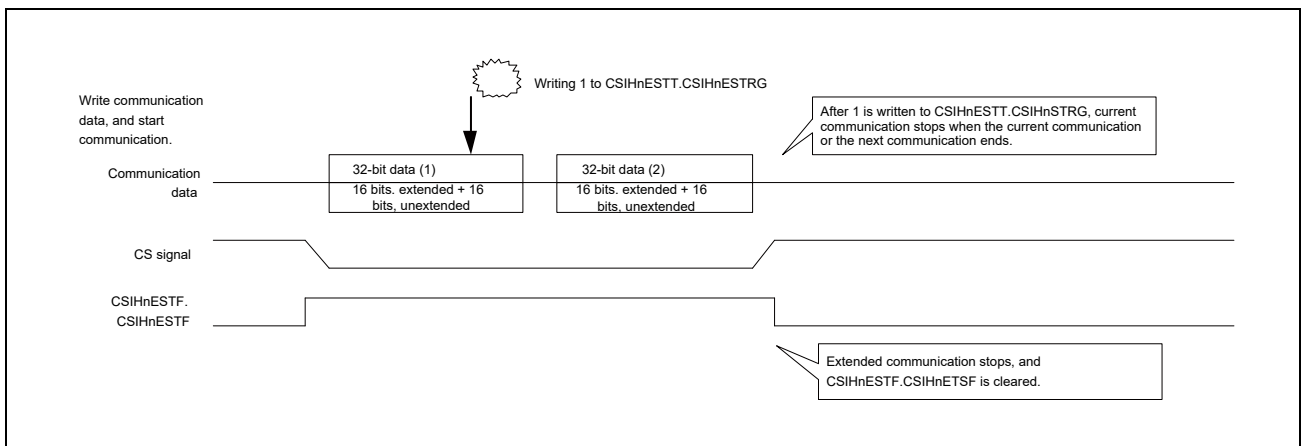


Figure 17.41 Stopping Extended Communication

- When communication is stopped, the CSIHnESTF.CSIHnETCF[4:0] bits indicate the remaining number of data transfers.
- Stopped communication cannot be resumed. Therefore, to complete the communication, perform the operating procedure to start the communication.

17.5.14.3 Extended Communication Status

CSIH enables checking of the status of extended communication. The CSIHnESTF.CSIHnETSF bit indicates whether communication is in progress or stopped, and the CSIHnESTF.CSIHnETCF[4:0] bits indicate the remaining number of communications.

The figures below show examples of communication status according to the condition of communication.

Table 17.48 Examples of Communication Status

Figure No.	Content	Register Transmit/Receive Mode Setting		Register Setting	
		CSIHnTXE	CSIHnRXE	CSIHnECTL1. CSIHnEDS[1:0]	CSIHnECTL1. CSIHnETC[3:0]
Figure 17.42	Receive-only or Transmit/Receive mode. Data size of 32 bits, transfer count of 3	0 or 1	1	10 _B	0010 _B
Figure 17.43	Receive-only or Transmit/Receive mode. Data size of 8 or 16 bits, transfer count of 4	0 or 1	1	00 _B or 01 _B	0111 _B
Figure 17.44	Transmit-only mode. Data size 32 bits, transfer count of 3	1	0	10 _B	0010 _B
Figure 17.45	Transmit-only mode. Data size of 8 or 16 bits, transfer count of 4	1	0	00 _B or 01 _B	0111 _B

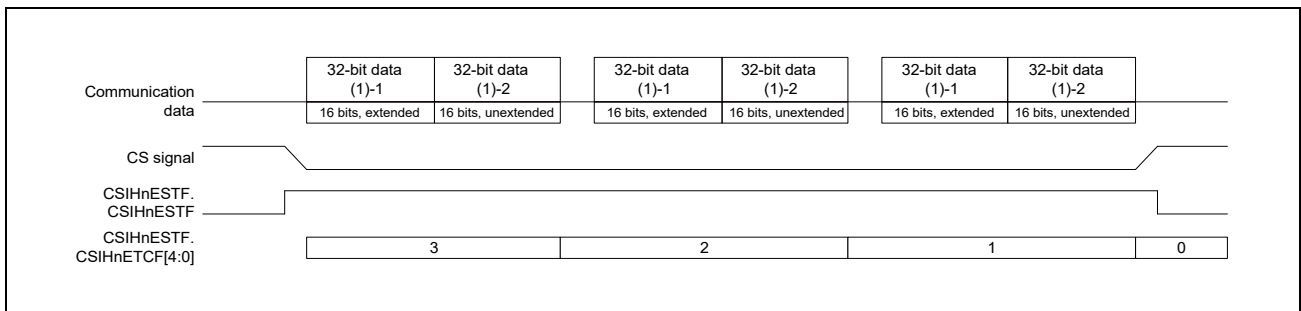


Figure 17.42 Receive-only or Transmit/Receive Mode Status of Communication with Data Size Set to 32 Bits and Transfer Count Set to 3

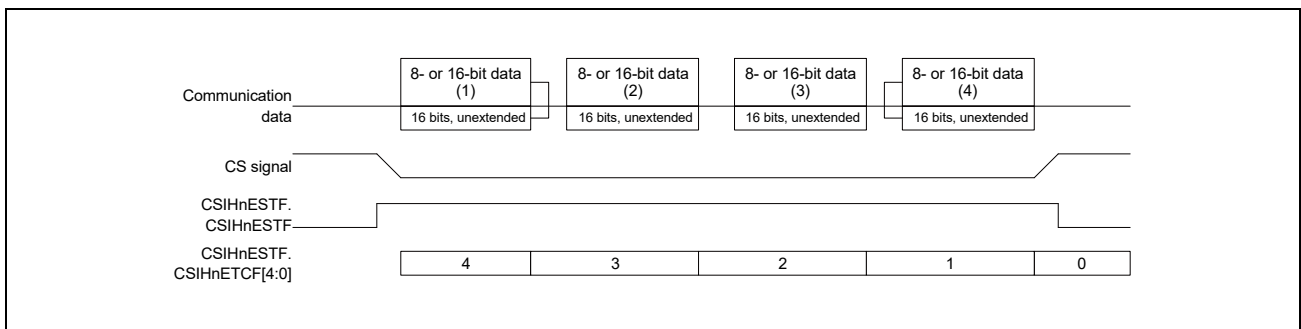


Figure 17.43 Receive-only or Transmit/Receive Mode Status of Communication with Data Size Set to 8 or 16 Bits and Transfer Count Set to 4

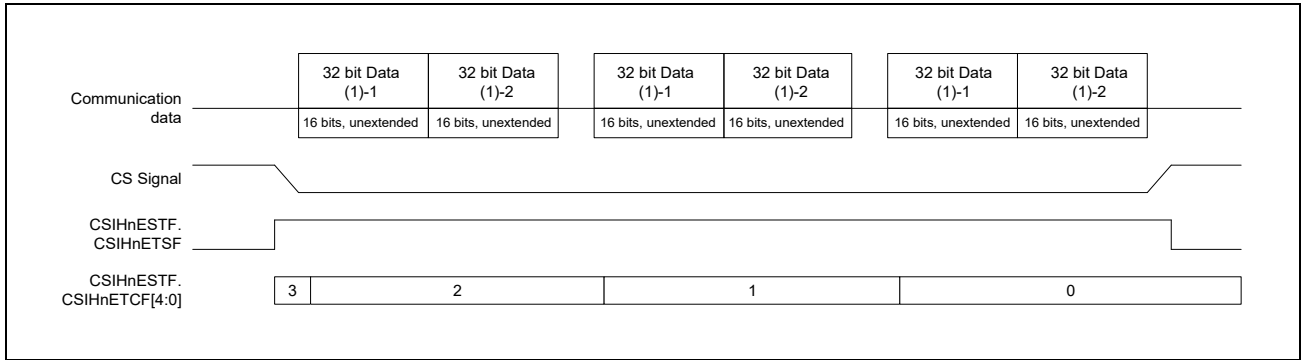


Figure 17.44 Transmit-only Mode Status of Communication with Data Size Set to 32 Bits and Transfer Count Set to 3

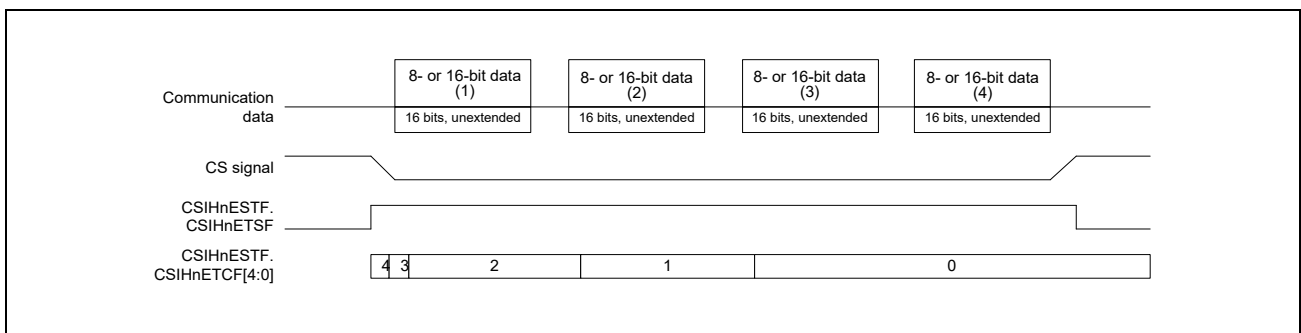


Figure 17.45 Transmit-only Mode Status of Communication with Data Size Set to 8 or 16 Bits and Transfer Count Set to 4

17.5.14.4 Extended Interrupt Control Function

When the extended communication function is enabled, the INTCSIHTIRE interrupt signal from CSIH is output without change. The INTCSIHTIC interrupt and INTCSIHTIR interrupt signals are output after they are fixed to the 0 level by the extended communication function. The INTCSIHTIJC interrupt signal is output only once when the specified number of communications ends.

The following figure shows the interrupt output for the extended communication with transfer count set to 2 (CSIHnECTL1.CSIHnETC[3:0] = 0001_B).

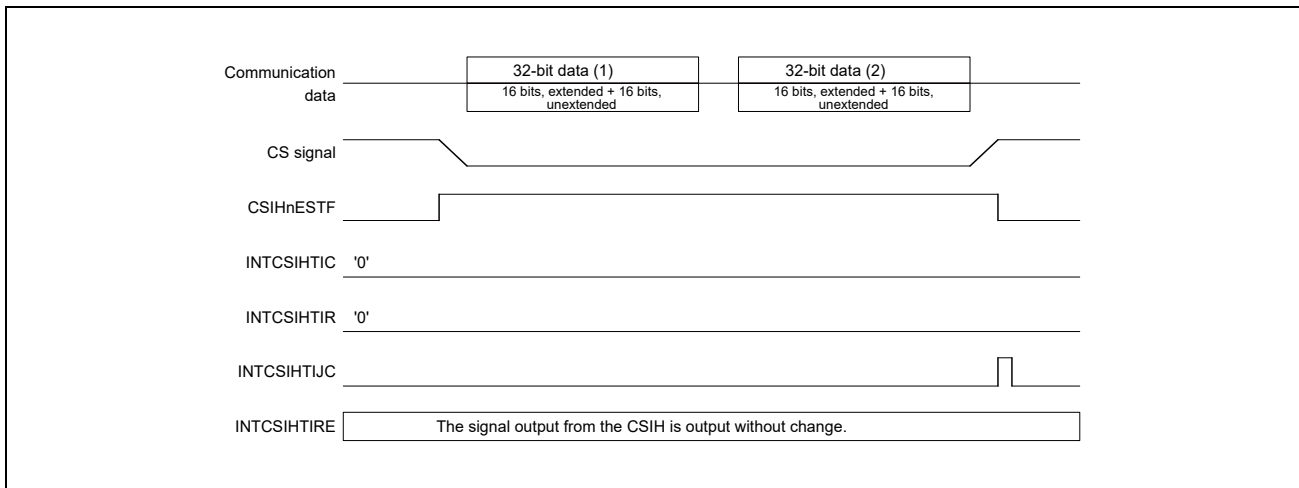


Figure 17.46 Interrupts Output with the Extended Communication Function Enabled

NOTE

When extended communication is stopped halfway by using the stop trigger bit (CSIHnESTT.CSIHnESTRG), the INTCSIHTIJC interrupt is not output. To determine whether the communication is in progress or stopped, check the extended communication status flag (CSIHnESTF.CSIHnETSF).

Note that it can be specified whether to enable individual interrupt outputs by using the CSIHnECTL0.CSIHnEIJE, CSIHnECTL0.CSIHnEIEE, CSIHnECTL0.CSIHnEIRE, and CSIHnECTL0.CSIH0EICE bits. When the output of an interrupt is disabled, the corresponding interrupt request is set to the Low level.

17.6 Operating Procedures

17.6.1 Procedures in Direct Access Mode

Two examples for a master are provided, one with job mode disabled, and the other with job mode enabled.

17.6.1.1 Transmit/Receive in Master Mode when Job Mode Is Disabled

The procedures below are based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- Transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)

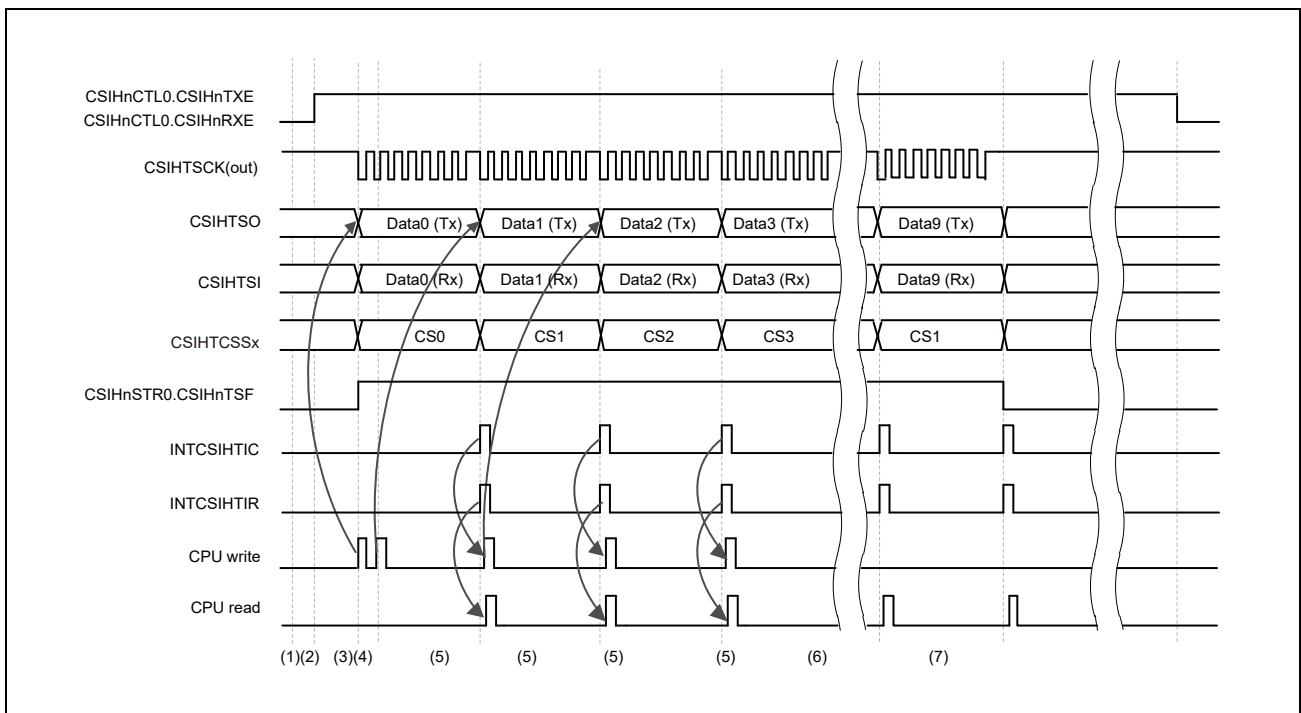


Figure 17.47 Master in Direct Access Mode, CSIHnCTL1.CSIHnJE = 0

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHnCSS0 to CSIHnCSS3.
Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), CSIHnRXE = 1 (permits reception), and CSIHnMBS = 1 (selects direct access mode).
3. Write the first data to be sent to the transmission register CSIHnTX0W. Within the same write operation, activate CS0. Transmission starts automatically when the first data becomes available.
4. Write the second data to CSIHnTX0W. If required, change the CS to address a different device. Writing the second data immediately after the first one avoids unnecessary delays between the data.
5. After data transmission/reception, the interrupts INTCSIHTIC and INTCSIHTIR are generated:
 - INTCSIHTIC indicates that the next data can be written to CSIHnTX0W.
 - INTCSIHTIR indicates that the reception register CSIHnRX0W must be read.
6. No more write action is required after completion of data 8. Data 9 (the last data) has been written in advance. However, reception register CSIHnRX0W must be read after completion of writing data 8 and 9.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of CSIHn.

17.6.1.2 Transmit/Receive in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- Transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- Normal INTCSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)
- Two jobs, each transmitting three data.

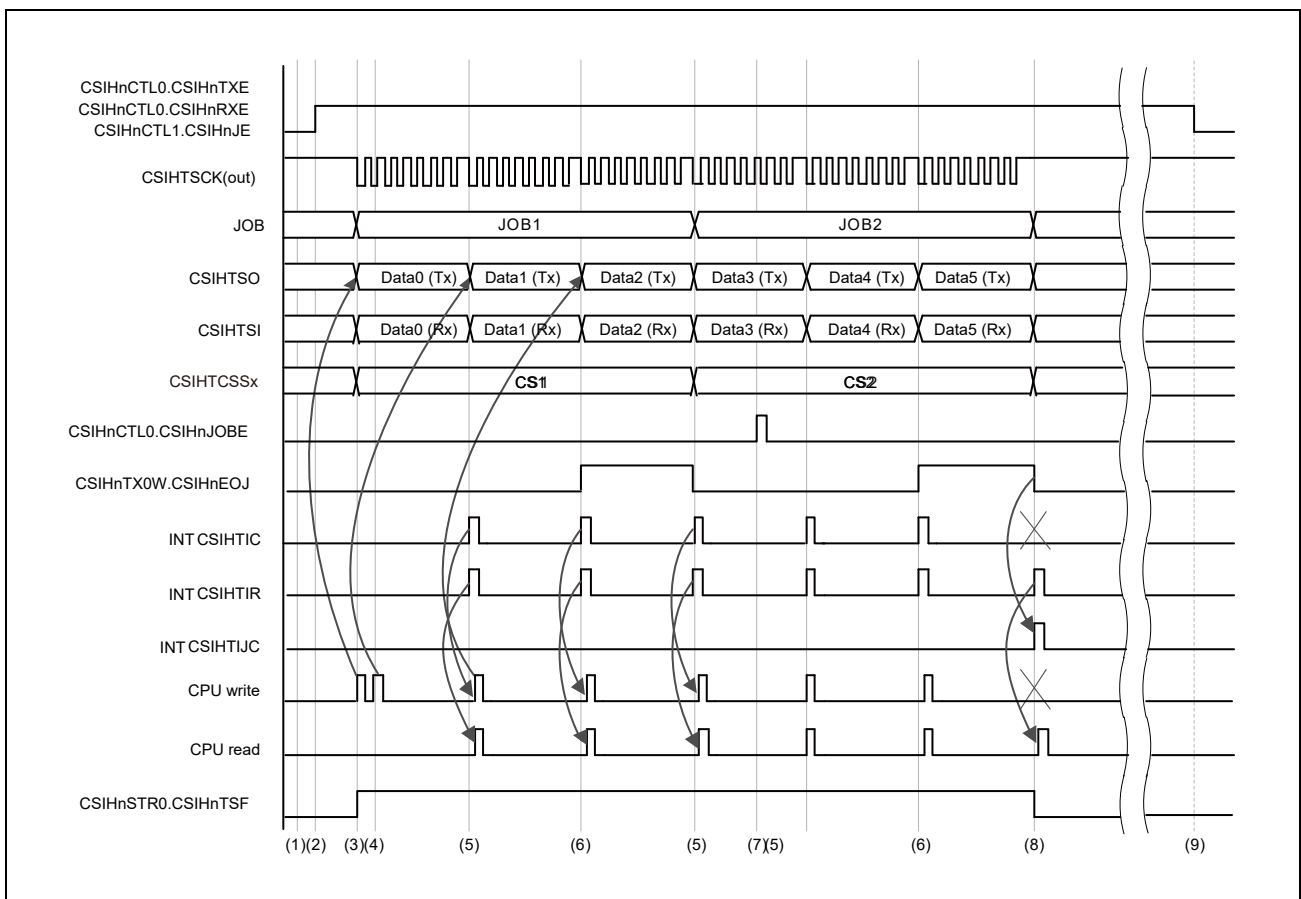


Figure 17.48 Master in Direct Access Mode, CSIHnCTL1.CSIHnJE = 1

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CS1 and CS2.
Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), CSIHnRXE = 1 (permits the reception), and CSIHnMBS = 1 (selects direct access mode).
3. Write the first data to be sent to the transmission register CSIHnTX0W. Transmission starts automatically when the first data becomes available.
The CSIHnSTR0.CSIHnTSF flag indicates that communication is in progress.
4. Write the second data to CSIHnTX0W. Writing the second data immediately after the first one avoids unnecessary delays between the data.
5. After every data transmission/reception, the interrupt requests INTCSIHTIC and INTCSIHTIR are generated.
 - INTCSIHTIC indicates that the next data can be written to CSIHnTX0W.
 - INTCSIHTIR indicates that the reception register CSIHnRX0W must be read.
6. Setting CSIHnTX0W.CSIHnEOJ = 1 indicates that the last data of the current job is sent. After that, the next job may begin.
7. By setting CSIHnCTL0.CSIHnJOB2 = 1, communication is forced to stop at the end of the current job (JOB2).
8. After the forced stop of communication, the interrupt request INTCSIHTIC is replaced by INTCSIHTIJC.
INTCSIHTIR is generated as usual.
The interrupt request INTCSIHTIJC indicates that the communication was forcibly stopped at the end of the current job.
The interrupt request INTCSIHTIC is not generated. Additionally, the transmission data available in the CSIHnTX0W register is not sent.
9. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation.
When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.
To start another transmission without stopping communication, perform steps 3 and later.

17.6.2 Procedures in Extended Communication Function Mode

The extended communication function of CSIH performs CSI (SPI) communication by using the registers dedicated to extended communication. It is possible to select a communication data size from 8, 16, and 32 bits by using the CSIHnECTL1.CSIH0EDS[1:0] bits. Also, it is possible to specify the number of extended communications in the range of 1 to 16 by using the CSIHnECTL1.CSIH0ETC[3:0] bits.

The following describes the operating procedure to use the extended communication function.

NOTE

When the extended communication function is enabled, do not write data to the transmit data registers (CSIHnTX0W[31:0] and CSIHnTX0H[15:0]) of CSIH. If data is written to any of these registers, the content of the communication using the extended communication function is not guaranteed.

Table 17.49 Procedure to Perform Communication Using the Extended Communication Function

Figure No.	Content	CSIH Register Setting		
		CSIHnCTL2. CSIHnPRS[2:0]	CSIHnCTL0. CSIHnTXE	CSIHnCTL0. CSIHnRXE
Figure 17.49	Transmission or transmission/reception in master or slave mode	— (Optional)	1	0 or 1
Figure 17.50	Reception in master mode	Other than 111 _B	0	1
Figure 17.51	Reception in slave mode	111 _B	0	1
Figure 17.52	Transmission or transmission/reception in master mode (more than 17 times)	Other than 111 _B	0 or 1	0 or 1
Figure 17.53	Transmission or transmission/reception in slave mode (more than 17 times)	111 _B	0 or 1	0 or 1

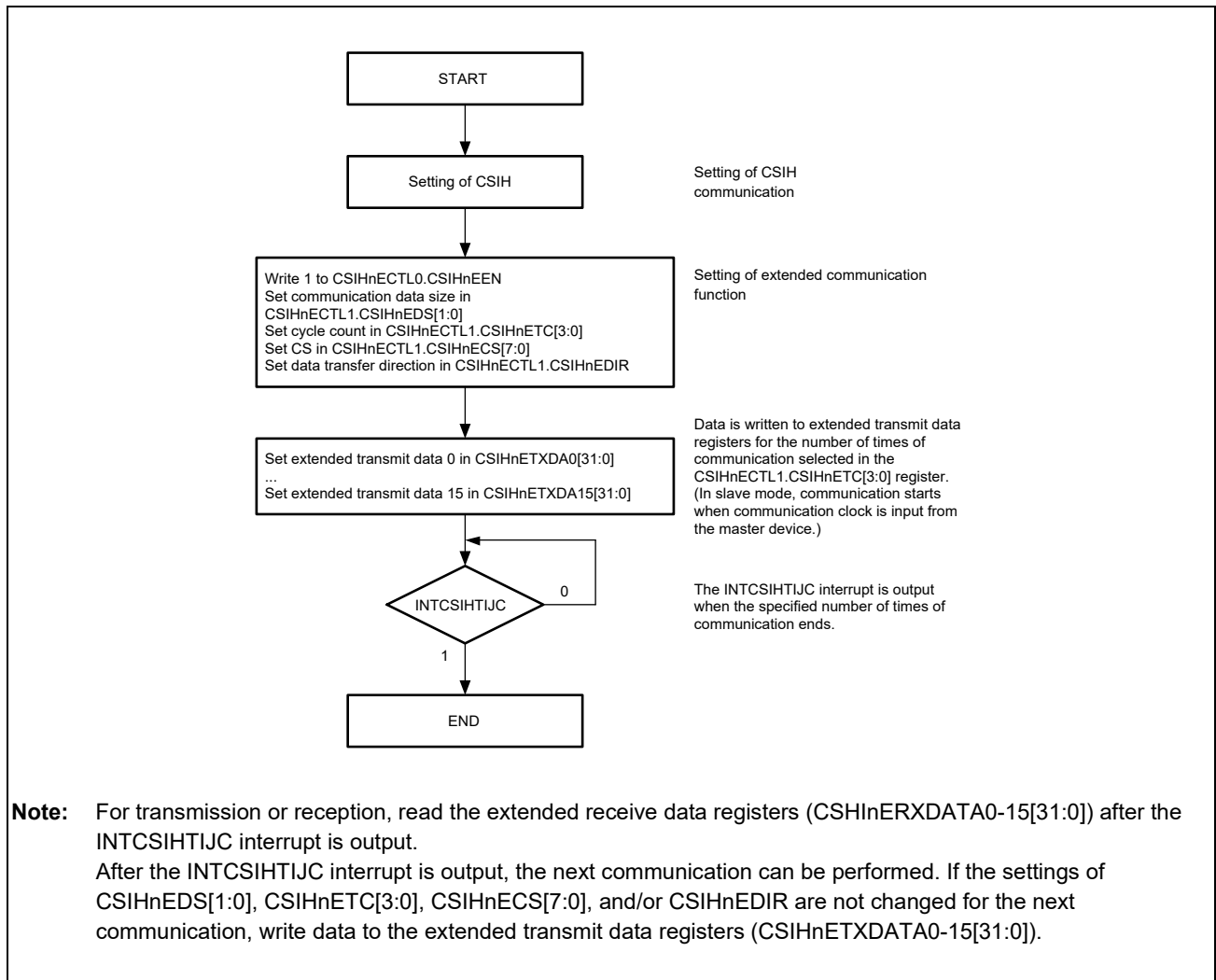


Figure 17.49 Operating Procedure for Transmission or Transmission/Reception

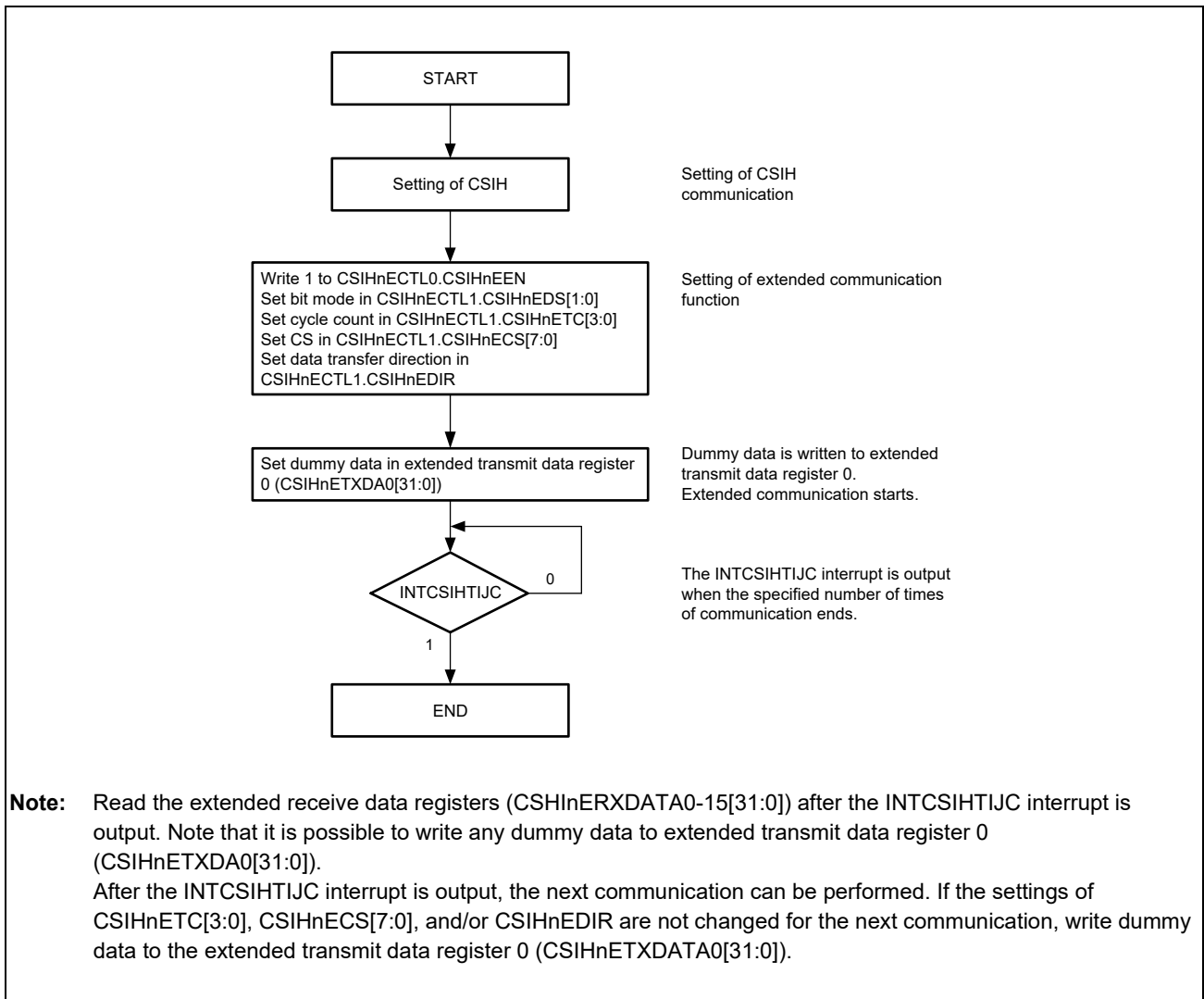


Figure 17.50 Operating Procedure for Reception in Master Mode

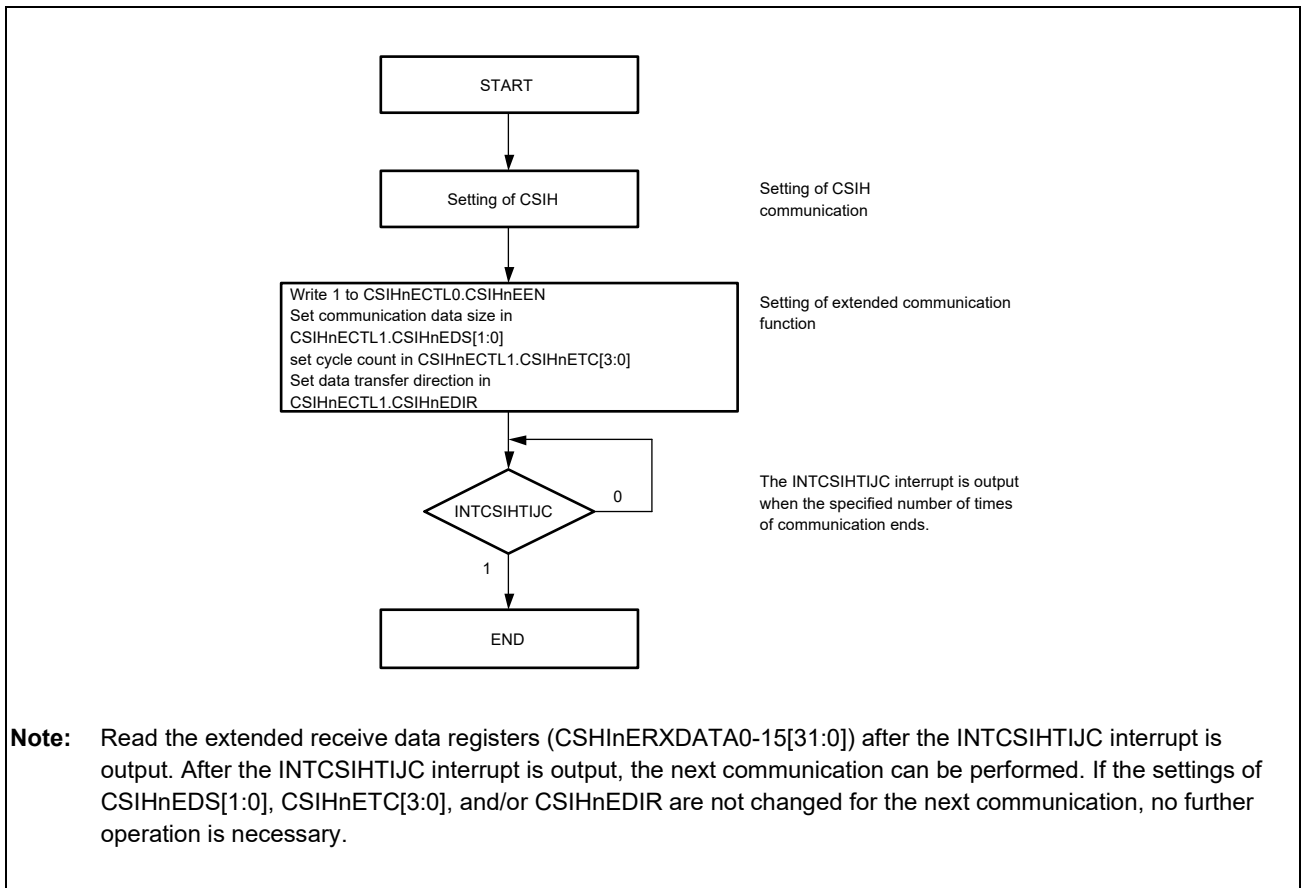


Figure 17.51 Operating Procedure for Reception in Slave Mode

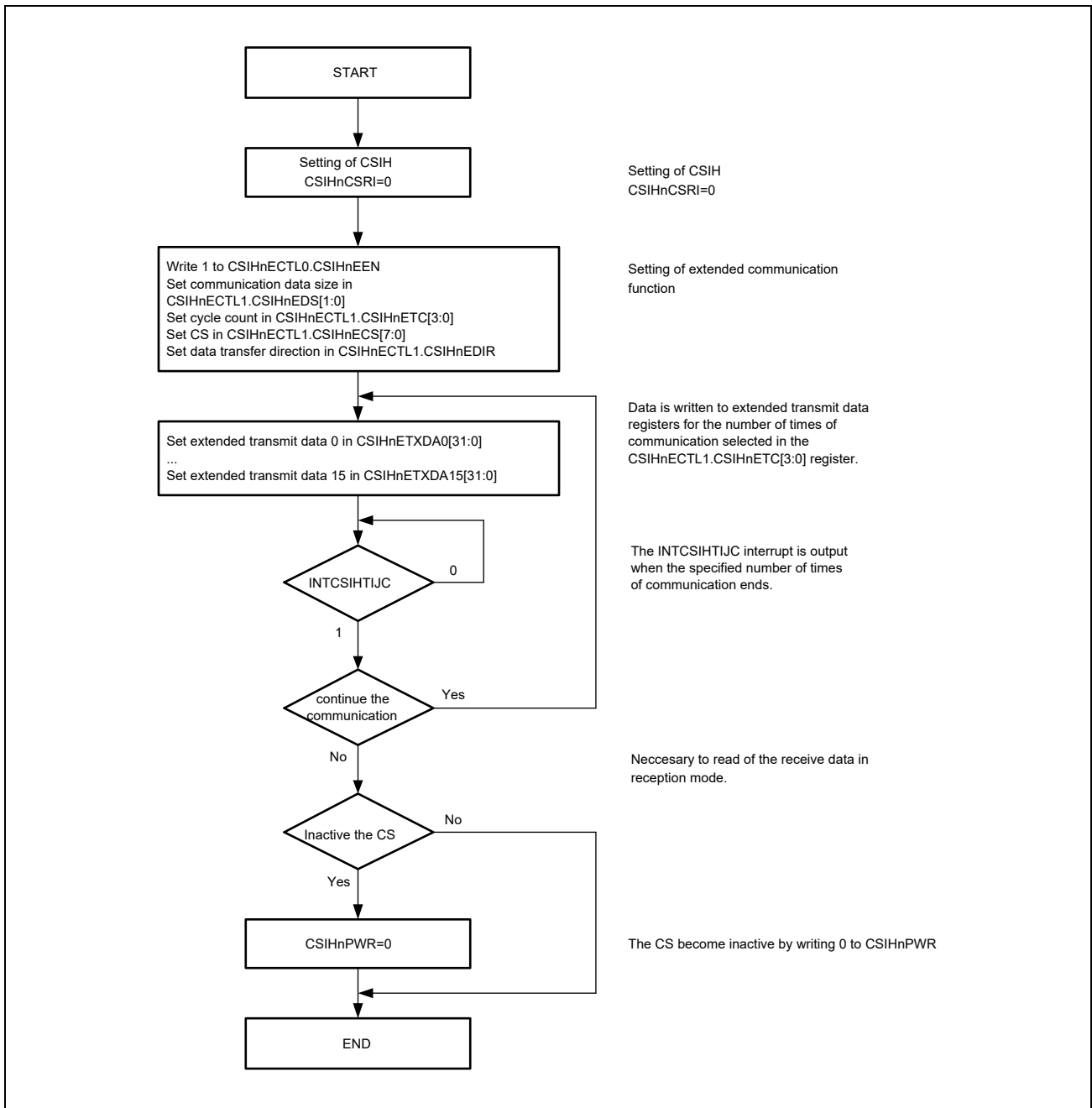


Figure 17.52 Transmission or Transmission/Reception in Master Mode (more than 17 times)

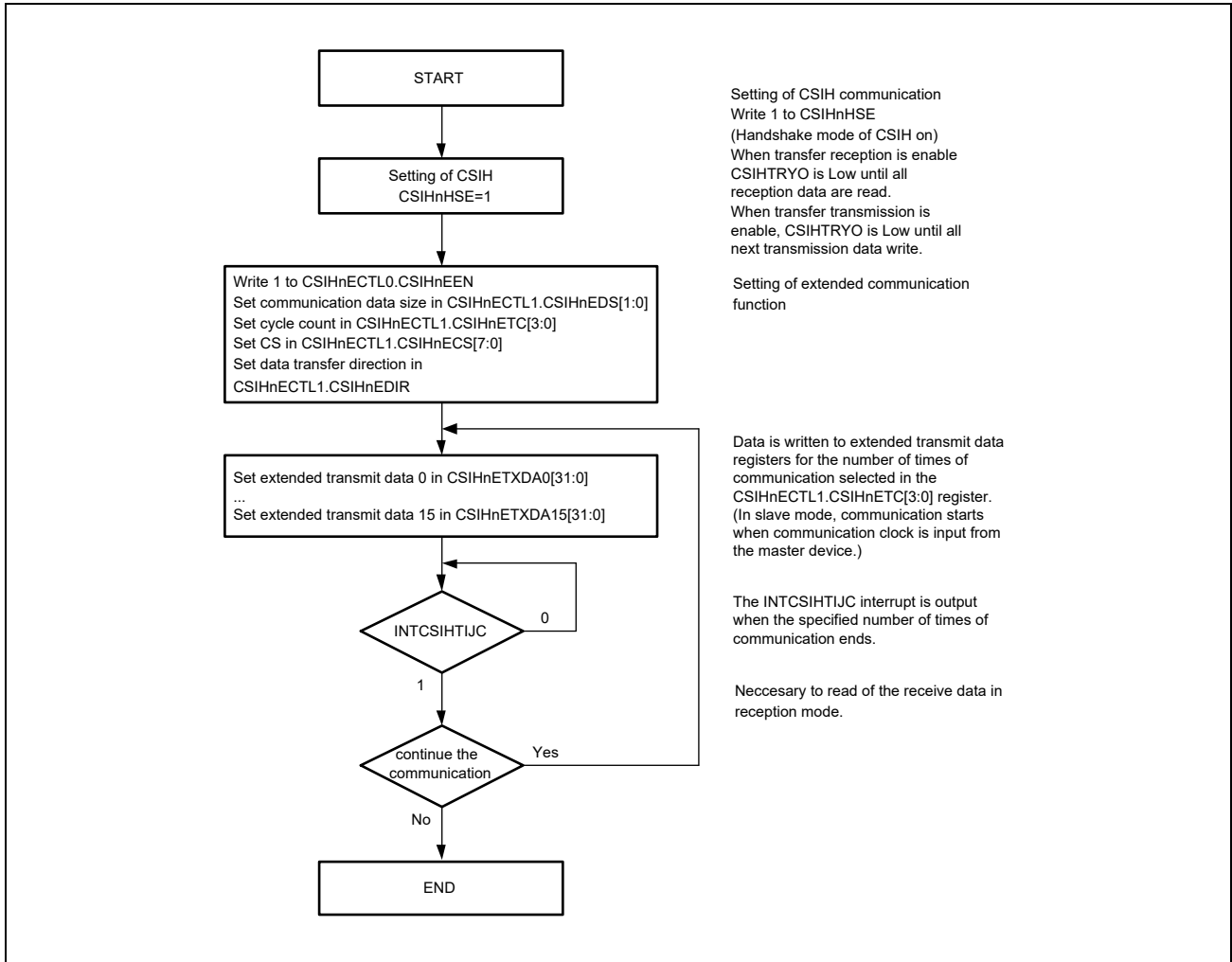


Figure 17.53 Transmission or Transmission/Reception in Slave Mode (more than 17 times)

17.6.2.1 Trigger to Start Extended Communication

For transmission or transmission/reception, CSIH starts communication when the last communication data to be transmitted is written. (In slave mode, CSIH starts communication when the communication clock is input from the master mode after the last data is written).

As examples, **Table 17.50** lists the transfer start timings for the 8-bit, 16-bit, or 32-bit communication with the data transfer count set to 16, and **Table 17.51** lists the transfer start timings for the 8-bit, 16-bit, or 32-bit communication with the data transfer count set to 4.

Table 17.50 Timings When CSIH Starts Transfer (Transfer Count: 16)

Address	Transmit Data Array	Register Name	
<CSIHn_base> + 110C _H	8-bit transmit data 0_0	CSIHnETXDA0	
<CSIHn_base> + 110D _H	8-bit transmit data 0_1		
<CSIHn_base> + 110E _H	8-bit transmit data 0_2		
<CSIHn_base> + 110F _H	8-bit transmit data 0_3		
<CSIHn_base> + 1110 _H	8-bit transmit data 1_0	CSIHnETXDA1	
<CSIHn_base> + 1111 _H	8-bit transmit data 1_1		
<CSIHn_base> + 1112 _H	8-bit transmit data 1_2		
<CSIHn_base> + 1113 _H	8-bit transmit data 1_3		
<CSIHn_base> + 1114 _H	8-bit transmit data 2_0	CSIHnETXDA2	
<CSIHn_base> + 1115 _H	8-bit transmit data 2_1		
<CSIHn_base> + 1116 _H	8-bit transmit data 2_2		
<CSIHn_base> + 1117 _H	8-bit transmit data 2_3		
<CSIHn_base> + 1118 _H	8-bit transmit data 3_0	CSIHnETXDA3	
<CSIHn_base> + 1119 _H	8-bit transmit data 3_1		
<CSIHn_base> + 111A _H	8-bit transmit data 3_2		
<CSIHn_base> + 111B _H	8-bit transmit data 3_3		-> For the communication of "8 bits x 16" data, transfer starts when the data (<CSIHn_base> + 111B _H) shown on the left is written.
.	.	.	
.	.	.	
.	.	.	
<CSIHn_base> + 1128 _H	8-bit transmit data 7_0	CSIHnETXDA7	
<CSIHn_base> + 1129 _H	8-bit transmit data 7_1		
<CSIHn_base> + 112A _H	8-bit transmit data 7_2		
<CSIHn_base> + 112B _H	8-bit transmit data 7_3		-> For the communication of "16 bits x 16" data, transfer starts when the data (<CSIHn_base> + 112B _H) shown on the left is written.
.	.	.	
.	.	.	
.	.	.	
<CSIHn_base> + 1148 _H	8-bit transmit data 15_0	CSIHnETXDA15	
<CSIHn_base> + 1149 _H	8-bit transmit data 15_1		
<CSIHn_base> + 114A _H	8-bit transmit data 15_2		
<CSIHn_base> + 114B _H	8-bit transmit data 15_3		-> For the communication of "32 bits x 16" data, transfer starts when the data (<CSIHn_base> + 114B _H) shown on the left is written.

Table 17.51 Timings When CSIH Starts Transfer (Communication Data Count: 4)

Address	Transmit Data Array	Register Name	
<CSIHn_base> + 110C _H	8-bit transmit data 0_0	CSIHnETXDA0	
<CSIHn_base> + 110D _H	8-bit transmit data 0_1		
<CSIHn_base> + 110E _H	8-bit transmit data 0_2		
<CSIHn_base> + 110F _H	8-bit transmit data 0_3		-> For the communication of "8 bits x 4" data, transfer starts when the data (<CSIHn_base> + 110F _H) shown on the left is written.
<CSIHn_base> + 1110 _H	8-bit transmit data 1_0	CSIHnETXDA1	
<CSIHn_base> + 1111 _H	8-bit transmit data 1_1		
<CSIHn_base> + 1112 _H	8-bit transmit data 1_2		
<CSIHn_base> + 1113 _H	8-bit transmit data 1_3		-> For the communication of "16 bits x 4" data, transfer starts when the data (<CSIHn_base> + 1113 _H) shown on the left is written.
<CSIHn_base> + 1114 _H	8-bit transmit data 2_0	CSIHnETXDA2	
<CSIHn_base> + 1115 _H	8-bit transmit data 2_1		
<CSIHn_base> + 1116 _H	8-bit transmit data 2_2		
<CSIHn_base> + 1117 _H	8-bit transmit data 2_3		
<CSIHn_base> + 1118 _H	8-bit transmit data 3_0	CSIHnETXDA3	
<CSIHn_base> + 1119 _H	8-bit transmit data 3_1		
<CSIHn_base> + 111A _H	8-bit transmit data 3_2		
<CSIHn_base> + 111B _H	8-bit transmit data 3_3		-> For the communication of "32 bits x 4" data, transfer starts when the data (<CSIHn_base> + 111B _H) shown on the left is written.
.	.	.	
.	.	.	
.	.	.	
<CSIHn_base> + 1128 _H	8-bit transmit data 7_0	CSIHnETXDA7	
<CSIHn_base> + 1129 _H	8-bit transmit data 7_1		
<CSIHn_base> + 112A _H	8-bit transmit data 7_2		
<CSIHn_base> + 112B _H	8-bit transmit data 7_3		
.	.	.	
.	.	.	
.	.	.	
<CSIHn_base> + 1148 _H	8-bit transmit data 15_0	CSIHnETXDA15	
<CSIHn_base> + 1149 _H	8-bit transmit data 15_1		
<CSIHn_base> + 114A _H	8-bit transmit data 15_2		
<CSIHn_base> + 114B _H	8-bit transmit data 15_3		

The CSIHnETXDA0-15[31:0] registers can be accessed in units of 8, 16, or 32 bits.

For the communication of “8 bits x 4” data described above, transfer is started by an 8-bit access to the address “<CSIHn_base> + 110F_H,” 16-bit access to the address “<CSIHn_base> + 110E_H,” or 32-bit access to the address “<CSIHn_base> + 110C_H.”

NOTE

For transmission or transmission/reception, CSIH starts communication when the last transmit data is written (to the extended transmit data register that has the highest address [among the “CSIHnETXDA0-15[31:0] registers] based on the specified transfer count). Therefore, the last transmit data must always be written at the end of writing. If the last transmit data is not written at the end of writing, values previously set in the CSIHnETXDA0-15[31:0] registers might be transmitted. Because transmission by CSI communication conflicts with writing of transmit data, it is not guaranteed which of previously set data and newly written data will be transmitted.

17.6.2.2 Storage Address of Received Data in Extended Communication

The storage address of the received data is determined by the communication data size.

Table 17.52 lists the storage address for the 8-bit, 16-bit, or 32-bit transfer reception data.

Table 17.52 Storage Address of Reception Data

Address	Register Name	Data Size 8 bits	Data Size 16 bits	Data Size 32 bits
<CSIHn_base> + 1150 _H	CSIHnERXDA0	Reception data1	Reception data 1	Reception data 1
<CSIHn_base> + 1151 _H		Reception data2		
<CSIHn_base> + 1152 _H		Reception data3	Reception data 2	
<CSIHn_base> + 1153 _H		Reception data4		
<CSIHn_base> + 1154 _H	CSIHnERXDA1	Reception data5	Reception data 3	Reception data 2
<CSIHn_base> + 1155 _H		Reception data6		
<CSIHn_base> + 1156 _H		Reception data7	Reception data 4	
<CSIHn_base> + 1157 _H		Reception data8		
<CSIHn_base> + 1158 _H	CSIHnERXDA2	Reception data9	Reception data 5	Reception data 3
<CSIHn_base> + 1159 _H		Reception data10		
<CSIHn_base> + 115A _H		Reception data11	Reception data 6	
<CSIHn_base> + 115B _H		Reception data12		
<CSIHn_base> + 115C _H	CSIHnERXDA3	Reception data13	Reception data 7	Reception data 4
<CSIHn_base> + 115D _H		Reception data14		
<CSIHn_base> + 115E _H		Reception data15	Reception data 8	
<CSIHn_base> + 115F _H		Reception data16		
.	.	Unused	.	.
.	.		.	.
.	.		.	.
<CSIHn_base> + 116C _H	CSIHnERXDA7	Unused	Reception data 15	Reception data 8
<CSIHn_base> + 116D _H		Unused		
<CSIHn_base> + 116E _H		Unused	Reception data 16	
<CSIHn_base> + 116F _H		Unused		
.	.	Unused	Unused	.
.	.			.
.	.			.
<CSIHn_base> + 118C _H	CSIHnERXDA15	Unused	Unused	Reception data 16
<CSIHn_base> + 118D _H		Unused		
<CSIHn_base> + 118E _H		Unused	Unused	
<CSIHn_base> + 118F _H		Unused		

Section 18 Serial Communication Interface 3 (SCI3)

18.1 Features of SCI3

18.1.1 Units and Channels

This product contains the serial communication interface for the number of channels shown below.

Table 18.1 Number of Channels (For E2x-FCC1)

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Channels	4	4
Name	SCI3n (n = 0 to 3)	

Table 18.2 Number of Channels (For E2M)

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Channels	4	4
Name	SCI3n (n = 0 to 3)	

Meaning of “n”

In this section, each channel of the serial communication interface 3 is identified by “n” (“n” is channel numbers.). For example, a serial mode register is written as SCI3nSMR.

18.1.2 Register Base Address

The base addresses <SCI3n_base> of each SCI3 are listed in **Table 18.3**.

Table 18.3 Register Base Address

SCI3n Channel	SCI3n_base Address	Bus Group
<SCI30_base>	FFD9 0000 _H	Peripheral Group 2L
<SCI31_base>	FFD9 1000 _H	Peripheral Group 2L
<SCI32_base>	FFD9 2000 _H	Peripheral Group 2L
<SCI33_base>	FFD9 3000 _H	Peripheral Group 2L

18.1.3 Clock Supply

Clock supply by and to SCI3 is listed in the following table.

Table 18.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
SCI3n	PCLK	CLK_LSB

18.1.4 Interrupt Requests

Table 18.5 Interrupt Requests

Interrupt symbol name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
SCI30					
INTSCI30ERITEI	ERI / TEI	Receive error / Transmit end	367	—	—
INTSCI30RXI	RXI	Receive data full	368	group0-116	group0-116
INTSCI30TXI	TXI	Transmit data empty	369	group0-117	group0-117
SCI31					
INTSCI31ERITEI	ERI / TEI	Receive error / Transmit end	370	—	—
INTSCI31RXI	RXI	Receive data full	371	group0-118	group0-118
INTSCI31TXI	TXI	Transmit data empty	372	group0-119	group0-119
SCI32					
INTSCI32ERITEI	ERI / TEI	Receive error / Transmit end	373	—	—
INTSCI32RXI	RXI	Receive data full	374	group0-120	group0-120
INTSCI32TXI	TXI	Transmit data empty	375	group0-121	group0-121
SCI33					
INTSCI33ERITEI	ERI / TEI	Receive error / Transmit end	376	—	—
INTSCI33RXI	RXI	Receive data full	377	group0-122	group0-122
INTSCI33TXI	TXI	Transmit data empty	378	group0-123	group0-123

18.1.5 Reset Sources

Table 18.6 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
SCI3n	All registers	✓	✓	✓	✓	—	—

18.1.6 External Input / Output Signals

The SCI3 has the input / output pins listed in **Table 18.7**.

Table 18.7 Pin Configuration

Unit Signal Name	Outline	Alternative Port Pin Signal
SCI30		
SCI0SCK	SCI30 serial clock input / output	SCI0SCK
SCI0RxD	SCI30 receive data input	SCI0RxD
SCI0TxD	SCI30 transmit data output	SCI0TxD
SCI31		
SCI1SCK	SCI31 serial clock input / output	SCI1SCK
SCI1RxD	SCI31 receive data input	SCI1RxD
SCI1TxD	SCI31 transmit data output	SCI1TxD
SCI32		
SCI2SCK	SCI32 serial clock input / output	SCI2SCK
SCI2RxD	SCI32 receive data input	SCI2RxD
SCI2TxD	SCI32 transmit data output	SCI2TxD
SCI33		
SCI3SCK	SCI33 serial clock input / output	SCI3SCK
SCI3RxD	SCI33 receive data input	SCI3RxD
SCI3TxD	SCI33 transmit data output	SCI3TxD

CAUTION

This AC specification is only applied to the specific pin groups. For details, refer to Appendix file "Limited_conditions_for_AC_specification.xlsx".

18.2 Overview

18.2.1 Functional Overview

The serial communication interface 3 (SCI3: Serial Communication Interface 3) can handle two methods of serial communications: asynchronous and clock synchronous serial communications. Asynchronous serial data communications can be carried out with standard asynchronous communication LSIs such as Universal Asynchronous Receiver / Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA). Asynchronous mode is equipped with a serial communications function between multiple processors (multi-processor communications function).

SCI3 has following features:

- The serial data communication mode can be configured for asynchronous communications or clock synchronous communications.
- Full-duplex communications are available. The independent transmitter unit and receiver unit allow simultaneous transmission and reception. Both the transmitter and the receiver have a double-buffered structure, enabling continuous data transmission and reception.
- An arbitrary bit rate is selectable with the on-chip baud rate generator. An external clock is also selectable as a transmission/reception clock source.
- LSB-first or MSB-first transfer is selectable (except for asynchronous 7-bit data).
- Interrupt sources: 4 types consisting of transmit-end, transmit-data-empty, receive-data-full, and receive-error. Transmit-data-empty and receive-data-full interrupt sources can activate the sDMA or DTS.
- Module standby can be set for each channel (for details of module standby, see **Section 16, Standby Controller**).
- The bit rate modulation function can reduce errors averagely (even in a high bit rate) by correcting the output of the on-chip baud rate generator (except for the maximum speed in clock synchronous mode).
- The pin level of serial input data can be checked.

18.2.2 Serial Communication Modes

Asynchronous mode

- Data length: 7 bits or 8 bits selectable
- Stop bit length: 1 bit or 2 bits selectable
- Parity: Even parity, odd parity, or none selectable
- Receive error detection: Parity error, overrun error, and framing error
- Break detection: A break can be detected by reading a register when a framing error occurs.

Clock synchronous mode

- Data length: 8 bits
- Receive error detection: Overrun error

18.2.3 Block Diagram

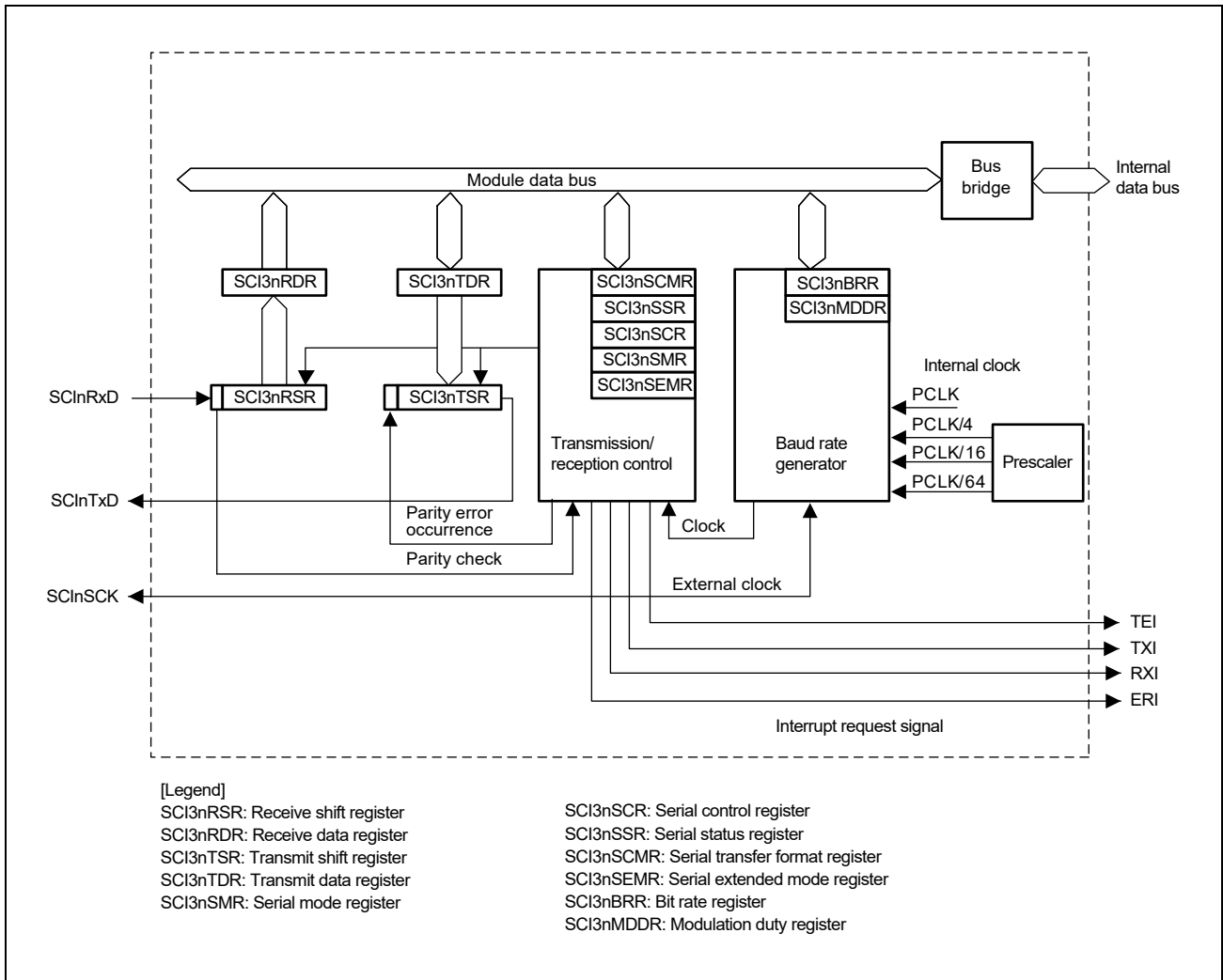


Figure 18.1 SCI3 Block Diagram

18.3 Registers

18.3.1 List of Registers

SCI3n register addresses are given as offsets from the base address <SCI3n_base>.

The SCI3 has the registers listed in **Table 18.8**. Some registers have limitations in read / write by the CPU.

Table 18.8 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection
SCI3n	Receive Shift Register	SCI3nRSR	—	—	—
	Serial Mode Register	SCI3nSMR	<SCI3n_base> + 0000 _H	8	—
	Bit Rate Register / Modulation Duty Register	SCI3nBRR / SCI3nMDDR	<SCI3n_base> + 0004 _H	8	—
	Serial Control Register	SCI3nSCR	<SCI3n_base> + 0008 _H	8	—
	Transmit Data Register	SCI3nTDR	<SCI3n_base> + 000C _H	8	—
	Transmit Shift Register	SCI3nTSR	—	—	—
	Serial Status Register	SCI3nSSR	<SCI3n_base> + 0010 _H	8	—
	Receive Data Register	SCI3nRDR	<SCI3n_base> + 0014 _H	8	—
	Serial Transfer Format Register	SCI3nSCMR	<SCI3n_base> + 0018 _H	8	—
	Serial Extended Mode Register	SCI3nSEMR	<SCI3n_base> + 001C _H	8	—

CAUTION

- SCI3nBRR and SCI3nMDDR are allocated to the same address (relative address 4). The SCI3nMDDRS bit in SCI3nSEMR is used to switch these registers.
- Relative addresses $4k + 1$, $4k + 2$, and $4k + 3$ ($k = 0$ to 7) are reserved areas. These areas are always read as 0. Writing is invalid.

18.3.2 SCI3nRSR — Receive Shift Register

SCI3nRSR is a shift register which is used to receive serial data input from the SCIInRxD pin and convert it to parallel data. When one frame of data has been received, it is automatically transferred to SCI3nRDR. SCI3nRSR cannot be directly accessed by the CPU.

18.3.3 SCI3nRDR — Receive Data Register

SCI3nRDR is an 8-bit register to store receive data. The value of SCI3nRDR after a reset is 00_H. When one frame of data has been received, it is transferred from SCI3nRSR to this register allowing SCI3nRSR to receive the next data. SCI3nRSR and SCI3nRDR function as a double-buffer in this way, allowing continuous receive operations. Be sure to check that the RDRF flag in SCI3nSSR is set to 1 before reading SCI3nRDR. SCI3nRDR cannot be written from the CPU.

When the data length is 7 bits, received data is stored in bits 0 to 6. Bit 7 is fixed to 0.

At this time, the fixed value of bit 7 is not affected by the SINV bit in the SCI3nSCMR register.

18.3.4 SCI3nTDR — Transmit Data Register

SCI3nTDR is an 8-bit register to store transmit data. The value of SCI3nTDR after a reset is FF_H. When SCI3nTSR empty is detected, the transmit data written to SCI3nTDR is transferred to SCI3nTSR and transmission starts. The double-buffered structure of SCI3nTDR and SCI3nTSR allows continuous serial transmission. If the next transmit data has been written to SCI3nTDR when one frame of data is sent, the transmit data is transferred to SCI3nTSR to continue transmission. SCI3nTDR can always be read and written by the CPU. Be sure to check that the TDRE flag in SCI3nSSR is set to 1 before writing transmit data to SCI3nTDR.

18.3.5 SCI3nTSR — Transmit Shift Register

SCI3nTSR is a shift register to transmit serial data. To perform serial data transmission, transmit data written to SCI3nTDR is automatically transferred SCI3nTSR, and is then sent to the SCIInTxD pin. SCI3nTSR cannot be directly accessed by the CPU.

18.3.6 SCI3nSMR — Serial Mode Register

SCI3nSMR is a register used to select the transfer format and the clock source for the on-chip baud rate generator.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Note 1. Writable only when TE = RE = 0.

Table 18.9 SCI3nSMR Register Contents

Bit Position	Bit Name	Function
7	CM	Communication Mode 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	Character Length (Valid only in asynchronous mode) 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. It is fixed to LSB-first and the MSB (bit 7) in SCI3nTDR is not sent in transmission. In clock synchronous mode, the data length is fixed to 8 bits.
5	PE	Parity Enable (Valid only in asynchronous mode) When this bit is set to 1, a parity bit is added to transmit data and the parity bit is checked in reception. Regardless of the setting of this bit, no parity bit is added or checked in the multi-processor format.
4	PM	Parity Mode (Valid only when PE = 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity. When even parity is set, parity bit is added so that the total number of 1 bits in the transmit/receive character plus the parity bit is even. Similarly, when odd parity is set, parity bit is added so that the total number of 1 bits in the transmit / receive character plus the parity bit is odd.
3	STOP	Stop Bit Length (Valid only in asynchronous mode) 0: 1 stop bit for transmission 1: 2 stop bits for transmission In reception, only the first stop bit is checked regardless of the setting of this bit. If the second stop bit is 0, it is treated as the start bit of the next transmission frame.
2	MP	Multi-Processor Mode (Valid only in asynchronous mode) When this bit is set to 1, the multi-processor communication function is enabled. In multi-processor mode, settings of the PE and PM bits are invalid.
1, 0	CKS[1:0]	Clock Select 1, 0 These bits select the clock source for the on-chip baud rate generator. 00: PCLK clock (m = 0) 01: PCLK / 4 clock (m = 1) 10: PCLK / 16 clock (m = 2) 11: PCLK / 64 clock (m = 3) For the relation between the setting of these bits and the baud rate, see Section 18.3.11, SCI3nBRR — Bit Rate Register . The character m is the decimal notation of the value of m in Section 18.3.11, SCI3nBRR — Bit Rate Register .

18.3.7 SCI3nSCR — Serial Control Register

SCI3nSCR is a register used for the transmission / reception control, interrupt control, and transmission/reception clock source selection listed below. For interrupt requests, see **Section 18.8, Interrupt Sources**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W*1	R/W*1	R/W	R/W	R/W*2	R/W*2

Note 1. While the CM bit in SCI3nSMR is 1, a value of 1 can be written only when TE = 0 and RE = 0. After the TE or RE bit is set to 1, only 0 can be written to TE and RE. While the CM bit in SCI3nSMR is 0, writing is enabled at any timing.

Note 2. Writable only when TE = 0 and RE = 0. Also, writable at the same time when TE = 0 and RE = 0 are written.

Table 18.10 SCI3nSCR Register Contents (1/2)

Bit Position	Bit Name	Function
7	TIE	Transmit Interrupt Enable When this bit is set to 1, TXI interrupt request is enabled. TXI interrupt request can be cancelled by reading 1 from the TDRE flag and then clearing the flag to 0, or clearing the TIE bit.
6	RIE	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled. RXI and ERI interrupt requests can be cancelled by reading 1 from the RDRF, FER, PER, or ORER flag and then clearing the flag to 0, or clearing the RIE bit.
5	TE	Transmit Enable When this bit is set to 1, transmission is enabled. In this state, serial transmission is started by writing transmit data to SCI3nTDR and clearing the TDRE flag in SCI3nSSR to 0. Be sure to configure SCI3nSMR before setting the TE bit to 1 to determine the transmission format. When this bit is set to 0 to disable transmission, the TDRE flag in SCI3nSSR is fixed to 1.
4	RE	Receive Enable When this bit is set to 1, reception is enabled. In this state, serial reception is started by detecting a start bit in asynchronous mode or the input of synchronous clock in clock synchronous mode. Be sure to configure SCI3nSMR before setting the RE bit to 1 to determine the reception format. Even if reception is disabled by clearing this bit, the RDRF, FER, PER, and ORER flags are not affected and the previous value is retained.
3	MPIE	Multiprocessor Interrupt Enable (Valid only when MP in SCI3nSMR = 1 in asynchronous mode) When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the RDRF, FER, and ORER flags in SCI3nSSR to 1 is disabled. When the data with the multi-processor bit set to 1 is received, this bit is automatically cleared to 0, and normal reception is resumed. For details, see Section 18.5, Multi-Processor Communication Function . When the receive data includes MPB = 0 in SCI3nSSR, the receive data is not transferred from SCI3nRSR to SCI3nRDR, a receive error is not detected, and setting the RDRF, FER, and ORER flags in SCI3nSSR to 1 is disabled. When the receive data includes MPB = 1 in SCI3nSSR, the MPB bit in SCI3nSSR is set to 1, the MPIE bit is automatically cleared to 0, the RXI and ERI interrupt requests are enabled (if the RIE bit in SCI3nSCR is set to 1), and setting the FER and ORER flags to 1 is enabled.

Table 18.10 SCI3nSCR Register Contents (2/2)

Bit Position	Bit Name	Function
2	TEIE	Transmit End Interrupt Enable When this bit is set to 1, TEI interrupt request is enabled. TEI interrupt request can be cancelled by reading 1 from the TDRE flag and then clearing the flag to 0 to clear the TEND flag to 0, or clearing the TEIE bit.
1, 0	CKE[1:0]	Clock Enable 1, 0 These bits select the clock source and the SCInSCK pin function. For asynchronous mode 00: On-chip baud rate generator (The SCInSCK pin functions as an input / output port.) 01: On-chip baud rate generator (The clock with the same frequency as the bit rate is output from the SCInSCK pin.) 1X: Setting prohibited For clock synchronous mode 0X: Internal clock (The SCInSCK pin functions as a clock output pin.) 1X: External clock (The SCInSCK pin functions as a clock input pin.)

Note: X: Don't care

NOTE

When writing to any bit other than the MPIE bit of this register, use a store instruction such that the value of the MPIE bit becomes 0.

Note that when using a bit-manipulation instruction for writing to this register, the read-modify-write operation may inadvertently set the MPIE bit to 1.

18.3.8 SCI3nSSR — Serial Status Register

SCI3nSSR consists of SCI3 status flags and multi-processor transmit/receive bits.

The TDRE, RDRF, ORER, PER, and FER flags can be cleared only.

Value after reset: 84_H

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Value after reset	1	0	0	0	0	1	0	0
R/W	R/W* ¹	R/W* ¹	R/W* ¹	R/W* ¹	R/W* ¹	R	R	R/W

Note 1. Only 0 can be written to clear the flag.

Table 18.11 SCI3nSSR Register Contents (1/2)

Bit Position	Bit Name	Function
7	TDRE	Transmit Data Register Empty Indicates whether or not transmit data exists in SCI3nTDR. [Setting conditions] <ul style="list-style-type: none"> When the TE bit in SCI3nSCR is 0 When the data in SCI3nTDR has been transferred to SCI3nTSR so that new data can be written to SCI3nTDR [Clearing condition] <ul style="list-style-type: none"> Writing 0 to TDRE after reading TDRE = 1 When transmit data is written to TDR while TE = 1
6	RDRF	Receive Data Register Full Indicates whether or not receive data exists in SCI3nRDR [Setting condition] <ul style="list-style-type: none"> When reception finishes successfully and the receive data is transferred from SCI3nRSR to SCI3nRDR [Clearing conditions] <ul style="list-style-type: none"> Writing 0 to RDRF after reading RDRF = 1 When data is read from SCI3nRDR Even when the RE bit in SCI3nSCR is cleared, the RDRF flag is not affected and the previous value is retained. Note that completing the reception of the next data with the RDRF flag set to 1 will cause an overrun error, resulting in the loss of the receive data.
5	ORER	Overrun Error Indicates that an overrun error has occurred during reception and the reception ended abnormally. [Setting condition] <ul style="list-style-type: none"> When the next data is received while RDRF = 1 In SCI3nRDR, data received prior to the overrun error is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be continued. In clock synchronous mode, serial transmission also cannot be continued. [Clearing condition] <ul style="list-style-type: none"> Writing 0 to ORER after reading ORER = 1 Even when the RE bit in SCI3nSCR is cleared, the ORER flag is not affected and the previous value is retained.

Table 18.11 SCI3nSSR Register Contents (2/2)

Bit Position	Bit Name	Function
4	FER	<p>Framing Error</p> <p>Indicates that a framing error has occurred during reception in asynchronous mode and the reception ended abnormally.</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> When the stop bit is 0 In 2-stop-bit mode, only whether the first stop bit is 1 is checked but the second stop bit is not checked. Although the receive data that existed when a framing error has occurred is transferred to SCI3nRDR, the RDRF flag is not set. In addition, while the FER flag is set to 1, the subsequent receive data is not transferred to SCI3nRDR. <p>[Clearing condition]</p> <ul style="list-style-type: none"> Writing 0 to FER after reading FER = 1 Even when the RE bit in SCI3nSCR is cleared to 0, the FER flag is not affected and the previous value is retained.
3	PER	<p>Parity Error</p> <p>Indicates that a parity error has occurred during reception in asynchronous mode and the reception ended abnormally.</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> When a parity error is detected during reception Although the receive data that existed when a parity error has occurred is transferred to SCI3nRDR, the RDRF flag is not set. In addition, while the PER flag is set to 1, the subsequent receive data is not transferred to SCI3nRDR. <p>[Clearing condition]</p> <ul style="list-style-type: none"> Writing 0 to PER after reading PER = 1 Even when the RE bit in SCI3nSCR is cleared to 0, the PER flag is not affected and the previous value is retained.
2	TEND	<p>Transmit End</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> When the TE bit in SCI3nSCR is 0 When the TDRE flag is 1 while the last bit of a transmit character is being transmitted <p>[Clearing condition]</p> <ul style="list-style-type: none"> Writing 0 to the TDRE flag after reading TDRE = 1 When transmit data is written to SCI3nTDR while TE = 1
1	MPB	<p>Multi-Processor Bit</p> <p>Holds the value of the multi-processor bit in the receive frame.</p>
0	MPBT	<p>Multi-Processor Bit Transfer</p> <p>Sets the value of the multi-processor bit to be added to the transmit frame.</p>

18.3.9 SCI3nSCMR — Serial Transfer Format Register

SCI3nSCMR is a register to select transfer format which can be commonly configured for both asynchronous mode and clock synchronous mode.

Value after reset: F2_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SDIR	SINV	—	—
Value after reset	1	1	1	1	0	0	1	0
R/W	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. Writable only when TE = 0 and RE = 0.

Table 18.12 SCI3nSCMR Register Contents

Bit Position	Bit Name	Function
7 to 4	—	Reserved These bits are always read as 1. When writing, write the value after reset.
3	SDIR	Serial Data Transfer Direction (Valid in asynchronous mode and clock synchronous mode) This bit is used to select the direction of serial / parallel conversion. 0: Transfer with LSB-first 1: Transfer with MSB-first This bit is valid only when the transfer format is 8-bit data. For 7-bit data, LSB-first transfer is used.
2	SINV	Serial Data Invert (Valid in asynchronous mode and clock synchronous mode) This bit is used to invert the transfer data logic level. The SINV bit does not affect the logic level of the parity bit. To invert the parity bit, invert the PM bit in SCI3nSMR. 0: SCI3nTDR data is transmitted as it is, and receive data is stored in SCI3nRDR as it is. 1: SCI3nTDR data is inverted and transmitted, and receive data is inverted and stored in SCI3nRDR.
1	—	Reserved These bits are always read as 1. When writing, write the value after reset.
0	—	Reserved These bits are always read as 0. When writing, write the value after reset.

18.3.10 SCI3nSEMR — Serial Extended Mode Register

SCI3nSEMR is a register to select a 1-bit period.

Value after reset: 04_H

Bit	7	6	5	4	3	2	1	0
	BRME	MDDRS	—	—	ABCS	RXDMON	—	—
Value after reset	0	0	0	0	0	1	0	0
R/W	R/W*1	R/W*1	R	R	R/W*1	R	R	R

Note 1. Writable only when TE = 0 and RE = 0.

Table 18.13 SCI3nSEMR Register Contents

Bit Position	Bit Name	Function
7	BRME	Bit Rate Modulation Enable When this bit is set to 1, the bit rate modulation function is enabled.
6	MDDRS	Modulation Duty Register Select This bit is used to select an accessible register. 0: SCI3nBRR is accessible. 1: SCI3nMDDR is accessible.
5, 4	—	Reserved These bits are always read as 0. When writing, write the value after reset.
3	ABCS	Asynchronous Reference Clock Select (Valid only in asynchronous mode) This bit is used to select the reference clock for 1-bit period. 0: Operates on the reference clock with a frequency of 16 times the transfer rate. 1: Operates on the reference clock with a frequency of 8 times the transfer rate (double-speed operation).
2	RXDMON	Serial Input Data Monitor This bit indicates the SCInRxD pin state. 0: SCInRxD pin state is low level. 1: SCInRxD pin state is high level.
1, 0	—	Reserved These bits are always read as 0. When writing, write the value after reset.

18.3.11 SCI3nBRR — Bit Rate Register

SCI3nBRR is an 8-bit register to adjust the bit rate. Since each SCI3 channel has an independent baud rate generator, different bit rates can be set for each channel. **Table 18.15** shows the relationship between the setting (N) in SCI3nBRR and the bit rate (B) in normal asynchronous mode, clock synchronous mode. SCI3nBRR is allocated at the same address as SCI3nMDDR and is selected when MDDRS in SCI3nSEMR is 0. This register is writable only when TE = 0 and RE = 0.

Value after reset: FF_H

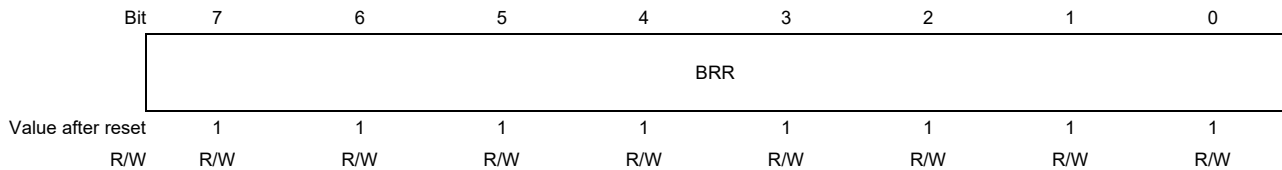


Table 18.14 SCI3nBRR Register Contents

Bit Position	Bit Name	Function
7 to 0	BRR	Baud rate generator setting (0 ≤ N ≤ 255)

Table 18.15 Relationship between Setting N in SCI3nBRR and Bit Rate B

Mode	ABCS Setting	Bit Rate	Mean Error
Asynchronous	0	$B = \frac{\phi \times 10^6}{64 \times 2^{2m-1} \times (N + 1)}$	$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{B \times 64 \times 2^{2m-1} \times (N + 1)} - 1 \right\} \times 100$
	1	$B = \frac{\phi \times 10^6}{32 \times 2^{2m-1} \times (N + 1)}$	$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{B \times 32 \times 2^{2m-1} \times (N + 1)} - 1 \right\} \times 100$
Clock synchronous	—	$B = \frac{\phi \times 10^6}{8 \times 2^{2m-1} \times (N + 1)}$	—

Note: B: Bit rate (bit/s)

N: SCI3nBRR setting for baud rate generator (0 ≤ N ≤ 255)

φ: Operating frequency PCLK (MHz)

m: Determined by the SCI3nSMR setting as shown in the following table.

SCI3nSMR Setting		m
CKS1	CKS0	
0	0	0
0	1	1
1	0	2
1	1	3

Table 18.16 lists sample N settings of the SCI3nBRR register in asynchronous mode. **Table 18.17** shows the maximum configurable bit rates.

Table 18.16 Examples of BRR Settings for Bit Rates (Asynchronous Mode)

Bit Rate (bit/s)	Operating Frequency $\phi = 40$ (MHz)							
	SCI3nSEMR.ABCS = 0				SCI3nSEMR.ABCS = 1			
	m	N	Actual Bit Rate (bit/s)	Error (%)	m	N	Actual Bit Rate (bit/s)	Error (%)
110	3	177	109.73	-0.25	—	—	—	—
150	3	129	150.24	0.16	3	255	152.59	1.73
300	3	64	300.48	0.16	3	129	300.48	0.16
600	2	129	600.96	0.16	3	64	600.96	0.16
1200	2	64	1201.92	0.16	2	129	1201.92	0.16
2400	1	129	2403.85	0.16	2	64	2403.85	0.16
4800	1	64	4807.69	0.16	1	129	4807.69	0.16
9600	0	129	9615.38	0.16	1	64	9615.38	0.16
19200	0	64	19230.77	0.16	0	129	19230.77	0.16
31250	0	39	31250.00	0.00	0	79	31250.00	0.00
38400	0	32	37878.79	-1.36	0	64	38461.54	0.16

Table 18.17 Maximum Bit Rate (Asynchronous Mode)

ϕ (MHz)	Setting			Maximum Bit Rate (bit/s)
	SCI3nSEMR.ABCS Setting	m	N	
40	0	0	0	1250000
	1	0	0	2500000

Table 18.18 lists sample N settings of the SCI3nBRR register in clock synchronous mode.

Table 18.19 shows the maximum configurable bit rates.

Table 18.18 Examples of Bit Rate Settings for Clock Synchronous Mode

Bit Rate (bit/s)	Operating Frequency $\phi = 40$ (MHz)		
	m	N	Actual Bit Rate (bit/s)
1 k	3	155	1001.60
2.5 k	2	249	2500.00
5 k	2	124	5000.00
10 k	1	249	10000.00
25 k	1	99	25000.00
50 k	1	49	50000.00
100 k	0	99	100000.00
250 k	0	39	250000.00
500 k	0	19	500000.00
1 M	0	9	1000000.00
2 M	0	4	2000000.00
2.5 M	0	3	2500000.00
5 M	0	1	5000000.00

Table 18.19 Maximum Bit Rate when Internal Clock is Output (Clock Synchronous Mode)

ϕ (MHz)	m	N	Maximum Bit Rate (bit/s)
40	0	1	5000000.00

18.3.12 SCI3nMDDR — Modulation Duty Register

SCI3nMDDR is a register to correct the bit rate adjusted by SCI3nBRR. The value of SCI3nMDDR after a reset is FF_H. When the BRME bit in SCI3nSEMR is set to 1, the bit rate generated by the on-chip baud rate generator is corrected to SCI3nMDDR / 256 on average. **Table 18.21** shows the relationship between the SCI3nMDDR setting and the bit rate B. SCI3nMDDR is allocated at the same address as SCI3nBRR and is selected when MDDRS in SCI3nSEMR is 1. This register is writable only when TE = 0 and RE = 0. Bit 7 is fixed to 1.

Value after reset: FF_H

Bit	7	6	5	4	3	2	1	0
	MDDR							
Value after reset	1	1	1	1	1	1	1	1
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.20 SCI3nMDDR Register Contents

Bit Position	Bit Name	Function
7 to 0	MDDR	Baud rate generator setting value (128 ≤ MDDR ≤ 255)

Table 18.21 Relationship between SCI3nMDDR Setting and Bit Rate B when Bit Rate Modulation Function Is Used

Mode	ABCS Setting	Bit Rate	Mean Error
Asynchronous	0	$B = \frac{\phi \times 10^6}{64 \times 2^{2m-1} \times (256 / \text{MDDR}) \times (N + 1)}$	$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{B \times 64 \times 2^{2m-1} \times (256 / \text{MDDR}) \times (N + 1)} - 1 \right\} \times 100$
	1	$B = \frac{\phi \times 10^6}{32 \times 2^{2m-1} \times (256 / \text{MDDR}) \times (N + 1)}$	$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{B \times 32 \times 2^{2m-1} \times (256 / \text{MDDR}) \times (N + 1)} - 1 \right\} \times 100$
Clock synchronous	—	$B = \frac{\phi \times 10^6}{8 \times 2^{2m-1} \times (256 / \text{MDDR}) \times (N + 1)}$	—

Note: B: Bit rate (bit/s)

N: SCI3nBRR setting of baud rate generator (0 ≤ N ≤ 255)

φ: Operating frequency PCLK (MHz)

m: See **Table 18.15**.

MDDR: SCI3nMDDR setting (128 ≤ MDDR ≤ 255)

18.4 Operation in Asynchronous Mode

Figure 18.2 shows the general format for asynchronous serial communications. One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level). In asynchronous serial communications, the communication line is usually held in the mark state (high level). The SCI3 monitors the communication line, and when the SCI3 detects the space state (low level), it recognizes a start bit and starts serial communications. Inside the SCI3, the transmitter and the receiver are independent, enabling full-duplex communications. Both the transmitter and the receiver have a double-buffered structure so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

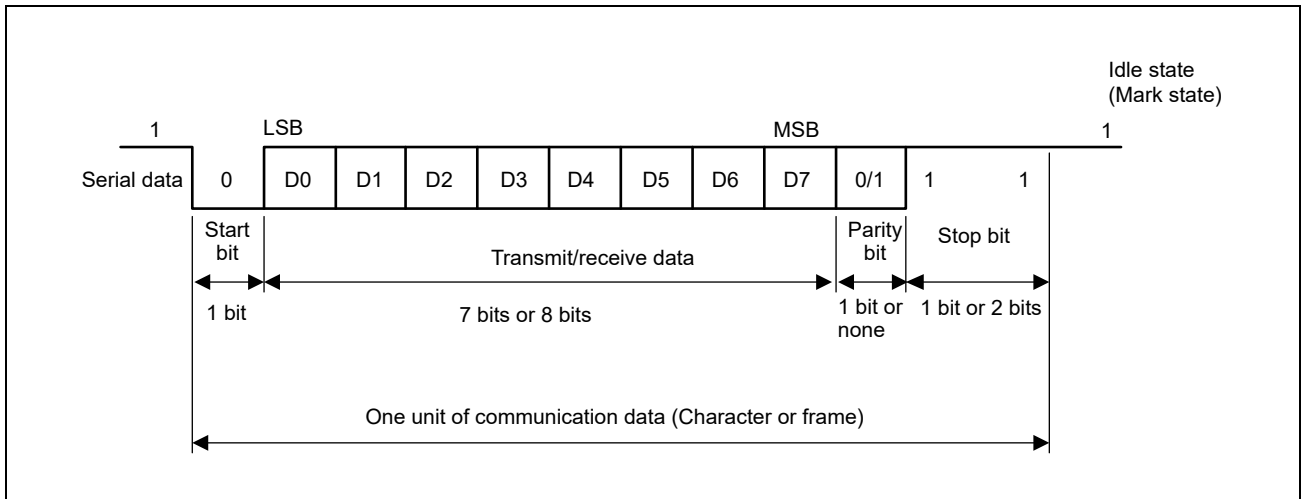


Figure 18.2 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, Two Stop Bits)

18.4.1 Transmission / Reception Format

Table 18.22 lists the transmission/reception formats that can be configured in asynchronous mode. Any of 12 transmission / reception formats can be selected according to the SCI3nSMR setting. For details of the multi-processor bit, see **Section 18.5, Multi-Processor Communication Function**.

Table 18.22 Serial Transmission / Reception Formats (Asynchronous Mode)

SMR Setting				Serial Transmission/Reception Format and Frame Length												
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12	
0	0	0	0	S	8-bit data								STOP			
0	0	0	1	S	8-bit data								STOP	STOP		
0	1	0	0	S	8-bit data								P	STOP		
0	1	0	1	S	8-bit data								P	STOP	STOP	
1	0	0	0	S	7-bit data							STOP				
1	0	0	1	S	7-bit data							STOP	STOP			
1	1	0	0	S	7-bit data							P	STOP			
1	1	0	1	S	7-bit data							P	STOP	STOP		
0	—	1	0	S	8-bit data								MPB	STOP		
0	—	1	1	S	8-bit data								MPB	STOP	STOP	
1	—	1	0	S	7-bit data							MPB	STOP			
1	—	1	1	S	7-bit data							MPB	STOP	STOP		

Note: S: Start bit; STOP: Stop bit; P: Parity bit; MPB: Multi-processor bit

18.4.2 Receive Data Sampling Timing and Reception Margin

In asynchronous mode, the SCI3 operates on a reference clock with a frequency of 16 times (8 times for the double-speed mode) the bit rate. In reception, the SCI3 samples the beginning of the start bit (low level) using the reference clock and performs internal synchronization. As shown in **Figure 18.3**, data is latched at the middle of each bit by sampling receive data at the rising edge of the eighth pulse (fourth pulse for the double-speed mode) of the reference clock.

Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100[\%] \quad \dots \text{formula (1)}$$

M: Reception margin

N: Ratio of bit rate to clock (N = 16 when ABCS in SCI3nSEMR = 0, N = 8 when ABCS in SCI3nSEMR = 1)

D: Duty cycle of clock (D = 0.5 to 1.0)

L: Frame length (L = 9 to 12)

F: Absolute value of clock frequency deviation

Assuming that F (absolute value of clock frequency deviation) = 0, D (duty cycle of clock) = 0.5, and N = 16 in formula (1), the reception margin is obtained by the formula below.

$$M = \left\{ 0.5 - \frac{1}{(2 \times 16)} \right\} \times 100[\%] = 46.875\%$$

However, this is only the calculated value, and a margin of 20% to 30% should be allowed in system design.

When the bit rate modulation function is used, the reference clock frequency is corrected on average.

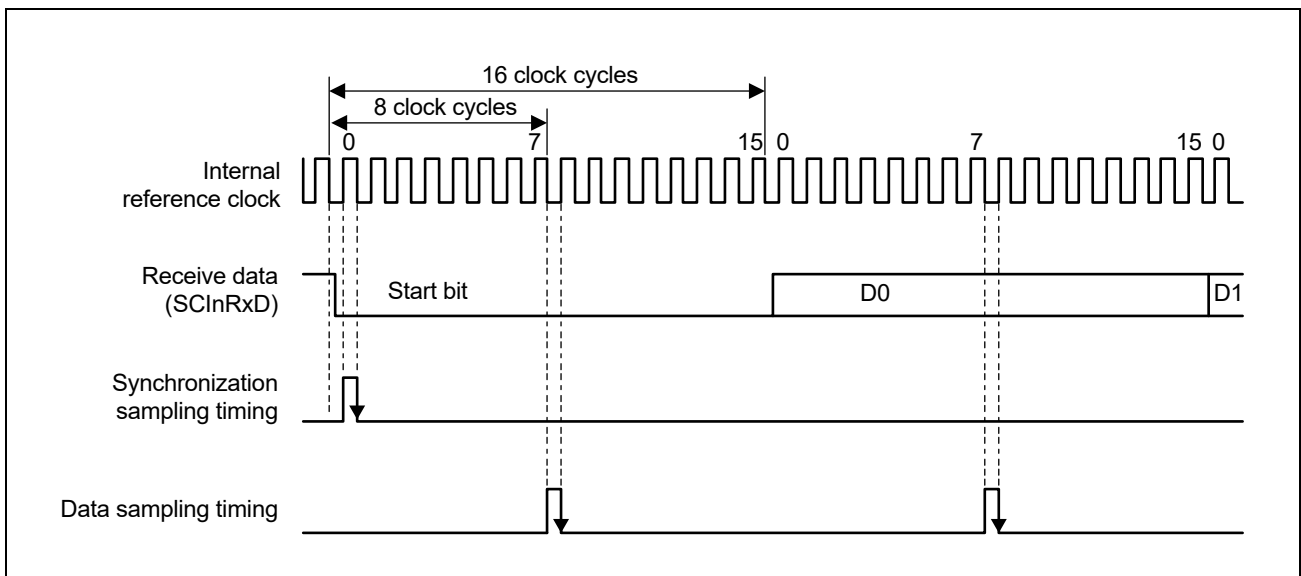


Figure 18.3 Receive Data Sampling Timing in Asynchronous Mode

18.4.3 Clock

An internal clock generated by the on-chip baud rate generator can be set as the SCI3's transmission / reception clock according to the settings of the CM bit in SCI3nSMR and the CKE1 and CKE0 bits in SCI3nSCR. When the SCI3 is operated on an internal clock, the clock can be output from the SCInSCK pin. For details of clock synchronous mode, see **Section 18.6, Operation in Clock Synchronous Mode**.

In asynchronous mode, the frequency of the clock output is equal to the bit rate and the phase is such that the rising edge of the clock comes at the center of the transmit data, as shown in **Figure 18.4**.

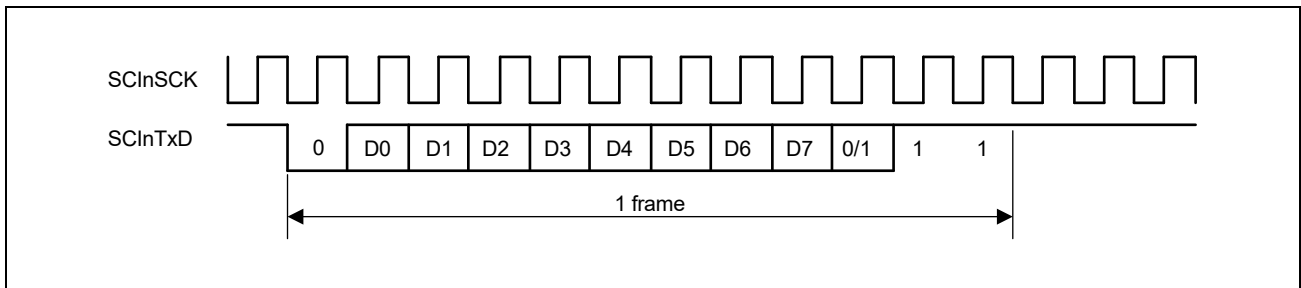


Figure 18.4 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode)

18.4.4 Double-Speed Operation

In addition to the operation described in **Section 18.4.3, Clock**, double-speed operation is enabled by the setting of the ABCS bit in SCI3nSEMR. In double-speed operation, the same operation on a clock with a frequency of 16 times the bit rate in normal operation can be conducted on a clock with a frequency of 8 times the bit rate, meaning that the SCI3 operates on a double transfer rate using the same reference clock.

Double-speed operation can be configured with either an internal clock generated by the on-chip baud rate generator.

18.4.5 SCI3 Initialization (Asynchronous Mode)

Before transmitting and receiving data, clear the TE and RE bits in SCI3nSCR and then initialize the SCI3 according to the sample flowchart in **Figure 18.5**. Before changing the operating mode or transfer format, be sure to clear the TE and RE bits to 0. Note that clearing the TE bit to 0 sets the TDRE flag to 1, but clearing the RE bit to 0 initializes neither the RDRF, PER, FER, and ORER flags nor SCI3nRDR.

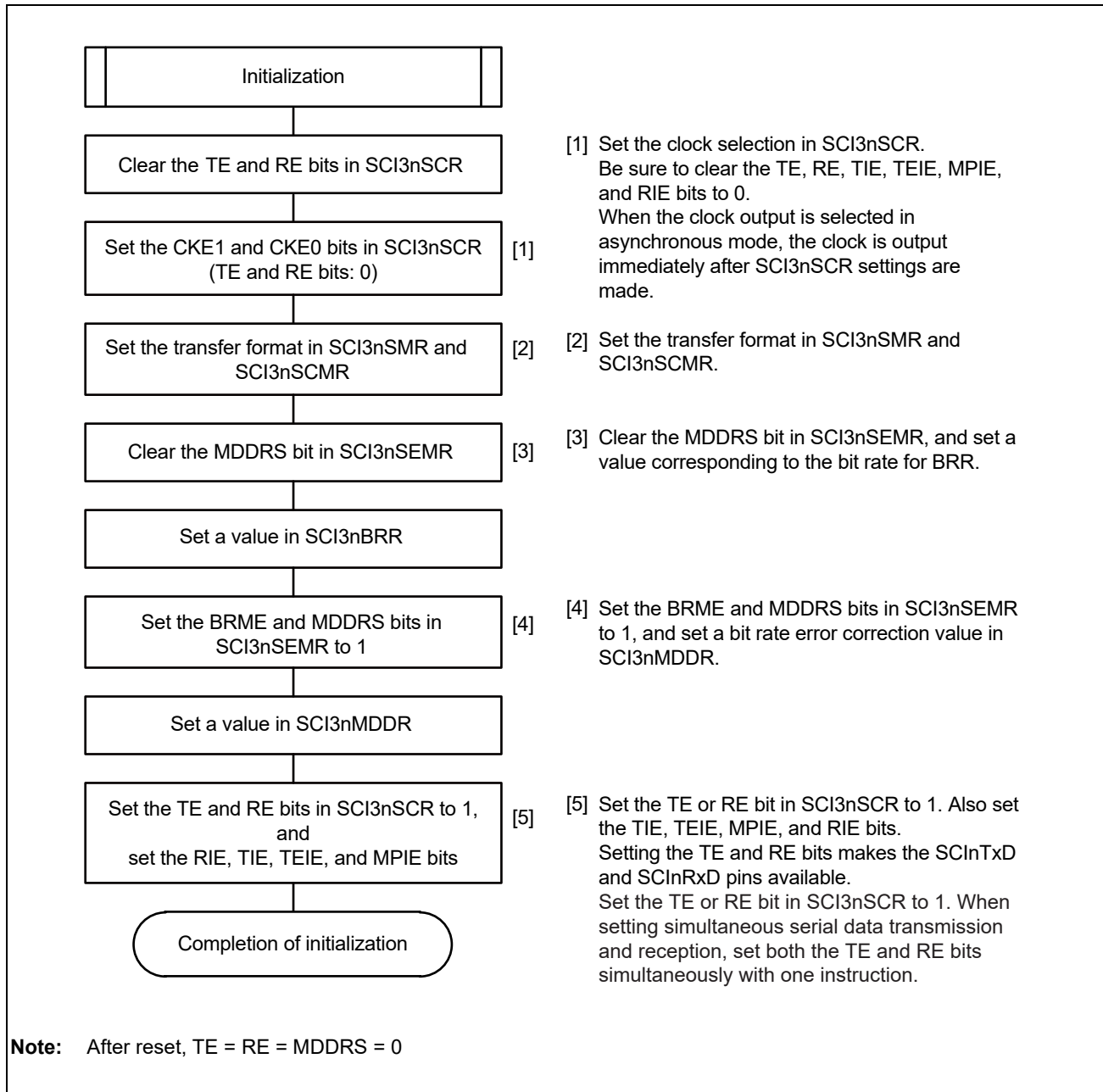


Figure 18.5 Sample Flowchart for SCI3 Initialization

18.4.6 Serial Data Transmission (Asynchronous Mode)

Figure 18.6 shows an example of operation for data transmission in asynchronous mode. In data transmission, the SCI3 operates as described below.

1. When the TE bit is set to 1, the High level (Preamble) for one frame is output.
2. When writing transmit data to SCI3nTDR at a trigger of TXI interrupt request, set the TIE bit to 1 and then set the TE bit to 1 or set both TIE and TE bits simultaneously with one instruction to generate a TXI interrupt request for starting data transfer. Set the TIE bit to 0 when a TXI interrupt request is not used.
3. When transmit data is written to SCI3nTDR, the TDRE flag is automatically cleared to 0. The SCI3 monitors the TDRE flag in SCI3nSSR. When the flag is cleared, the SCI3 recognizes that data has been written to SCI3nTDR and transfers data from SCI3nTDR to SCI3nTSR. When transmit data is written to SCI3nTDR while Preamble is output, the transmit data is transferred from SCI3nTDR to SCI3nTSR after Preamble is output.
4. Transmission starts after data is transferred from SCI3nTDR to SCI3nTSR and the TDRE flag is set to 1. If the TIE bit in SCI3nSCR is set to 1 at this time, a TXI interrupt request is generated.
5. Continuous transmission is enabled by writing the next transmit data to SCI3nTDR in this TXI interrupt processing routine before transmission of the previously transferred data is completed. When an TEI interrupt request is used, the TIE bit is cleared to 0 after the last transmit data has been written to SCI3nTDR, and the TEIE bit is set to 1.
6. Data is sent from the SCInTxD pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
7. The TDRE flag is checked when the stop bit is output.
8. When the TDRE flag is 0, the next transmit data is transferred from SCI3nTDR to SCI3nTSR, and re-performs transmission from the procedure No.4.
9. When the TDRE flag is 1, the TEND flag in SCI3nSSR is set to 1, the stop bit is sent, and then 1 is output to enter the mark state. If the TEIE bit in SCI3nSCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 18.8 shows a sample flowchart for data transmission. **Figure 18.9** shows a sample flowchart for stopping the SCI3 after data transmission.

It is unnecessary to clear the TE bit to 0 as long as communication format change, bit rate change, or transmission interruption are not required. Note that communication efficiency may be decreased by Preamble operation when the TE bit is set from 0 to 1 in each transmission.

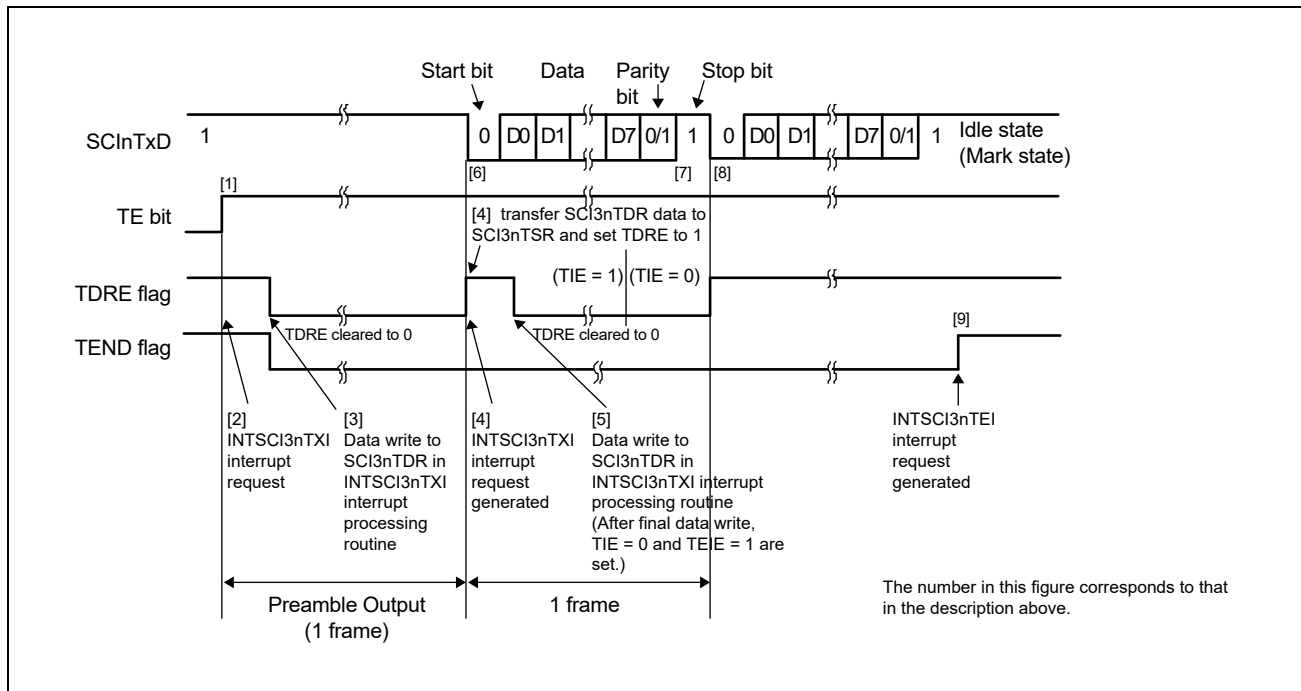


Figure 18.6 Example of Operation for Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

CAUTION

Do not clear the TDRE flag by writing SCI3nSSR register while the TE bit is set to 1. If the TDRE flag is cleared by register operation before writing data to SCI3nTDR, the register value of SCI3nTDR is transferred to SCI3nTSR at the next transmission start timing and its data value is transmitted. Thus, unintended data may be transferred.

The purpose of Preamble output:

When the TE bit is changed from 0 to 1, the High level (Preamble) for one frame is output so that SCI3 can communicate normally after the transmission is aborted and resumed as follows. Even if the receiver becomes uncertain due to the transmission abortion by the transmitter, the resumed transmission starts after the period of one frame, and the receiver can detect the start bit of the next frame.

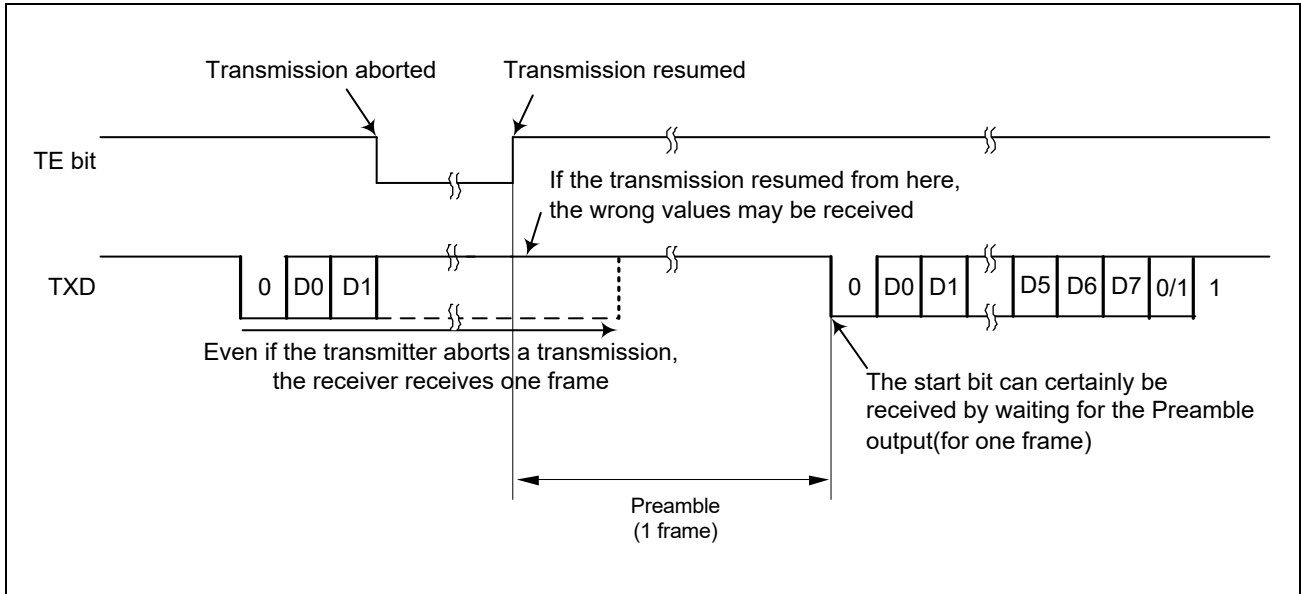


Figure 18.7 Example of Preamble Output

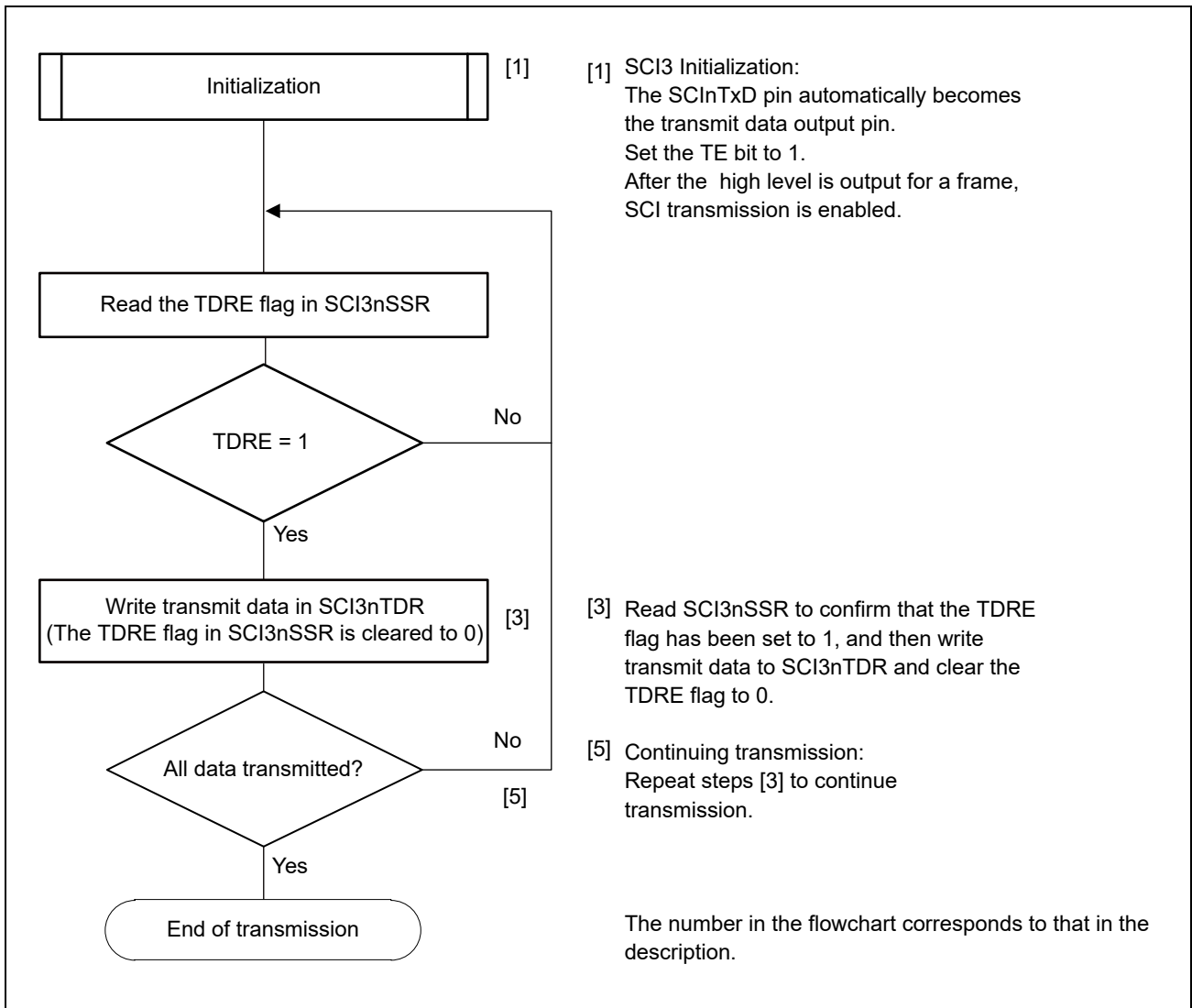


Figure 18.8 Example of Serial Transmission Flowchart

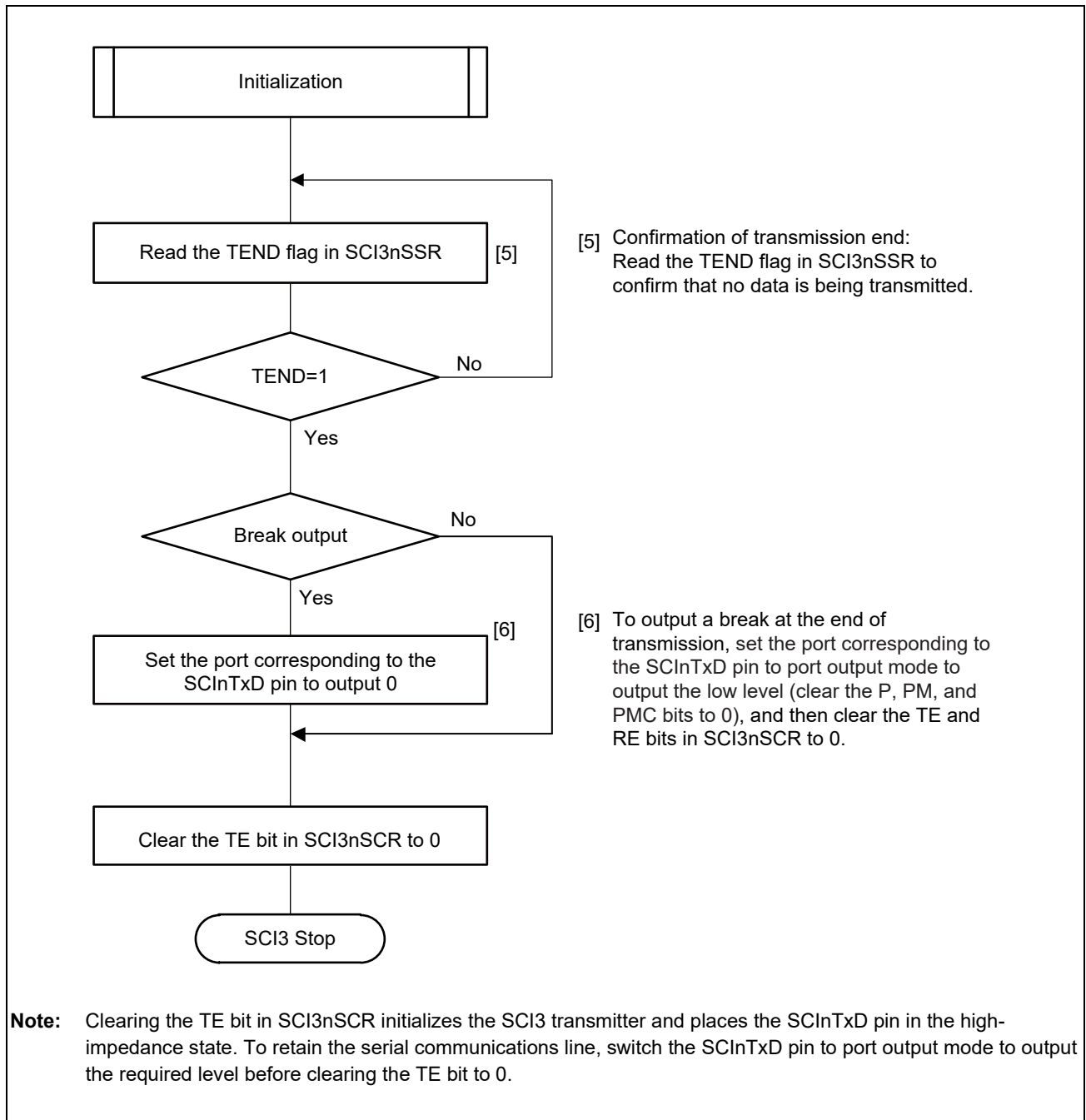


Figure 18.9 Example Flowchart for Stopping the SCI3 after Serial Transmission

18.4.7 Serial Data Reception (Asynchronous Mode)

Figure 18.10 shows an example of the operation for data reception in asynchronous mode. In data reception, the SCI3 operates as described below.

1. SCI3 monitors the communications line and upon detection of a start bit, it performs internal synchronization, stores receive data in SCI3nRSR, and checks the parity bit and the stop bit.
2. When an overrun error occurs (the next data has been received with the RDRF flag in SCI3nSSR set to 1), the ORER flag in SCI3nSSR is set to 1. If the RIE bit in SCI3nSCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to SCI3nRDR. The RDRF flag retains the state of being set to 1.
3. When a parity error is detected, the PER flag in SCI3nSSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCI3nSCR is set to 1 at this time, an ERI interrupt request is generated.
4. When a framing error (when the stop bit is 0) is detected, the FER flag in SCI3nSSR is set to 1 and receive data is transferred to SCI3nRDR. If the RIE bit in SCI3nSCR is set to 1 at this time, an ERI interrupt request is generated.
5. When reception finishes successfully, the RDRF flag in SCI3nSSR is set to 1 and receive data is transferred to SCI3nRDR. If the RIE bit in SCI3nSCR is set to 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to SCI3nRDR in this RXI interrupt processing routine before reception of the next receive data is completed. Reading the SCI3nRDR automatically clears the RDRF flag to 0.

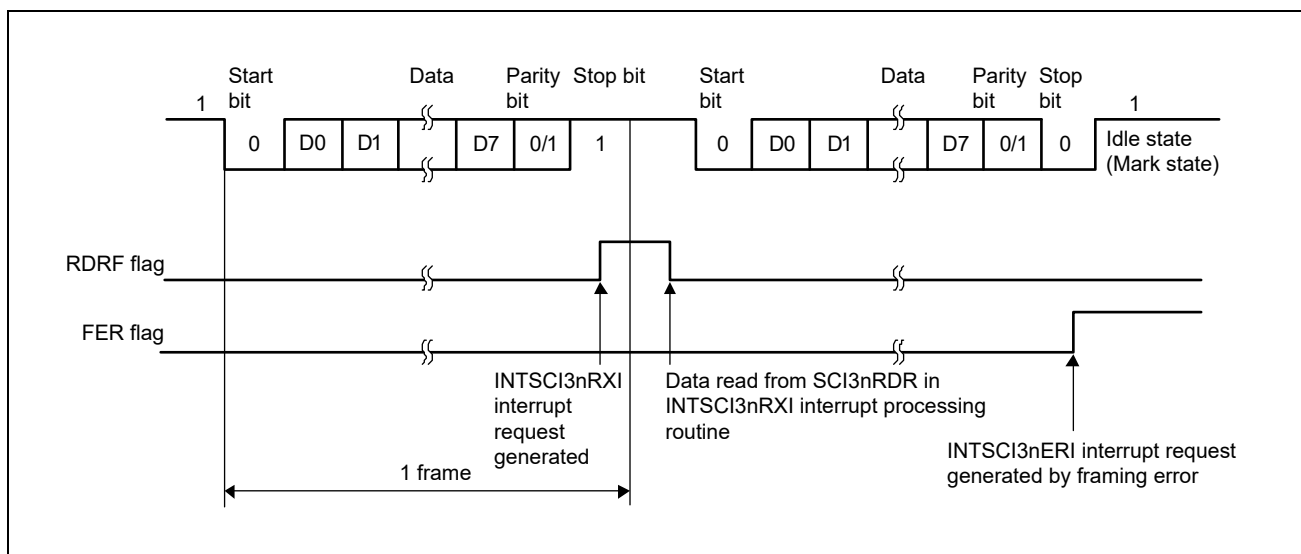


Figure 18.10 Example of Operation for Reception in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

Table 18.23 lists the states of the SCI3nSSR status flags and receive data handling when a receive error is detected. When a receive error is detected, the RDRF flag retains the status before receiving the data. Subsequent data reception is disabled while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF flags before continuing data reception. **Figure 18.11** or **Figure 18.12** shows sample flowchart for data reception.

Table 18.23 SCI3nSSR Status Flags and Receive Data Handling

SCI3nSSR Status Flags				Receive Data	Receive Status
RDRF*1	ORER	FER	PER		
1	0	0	0	Transferred to SCI3nRDR	Successful reception
0	0	1	0	Transferred to SCI3nRDR	Framing error
0	0	0	1	Transferred to SCI3nRDR	Parity error
0	0	1	1	Transferred to SCI3nRDR	Framing error + parity error
1*1	1	0	0	Lost	Overrun error
1*1	1	1	0	Lost	Overrun error + framing error
1*1	1	0	1	Lost	Overrun error + parity error
1*1	1	1	1	Lost	Overrun error + framing error + parity error

Note: "+" indicates that two or more receive statuses occur simultaneously in a single reception operation.

Note 1. In the case of an overrun error, the RDRF flag retains the state before the data reception.

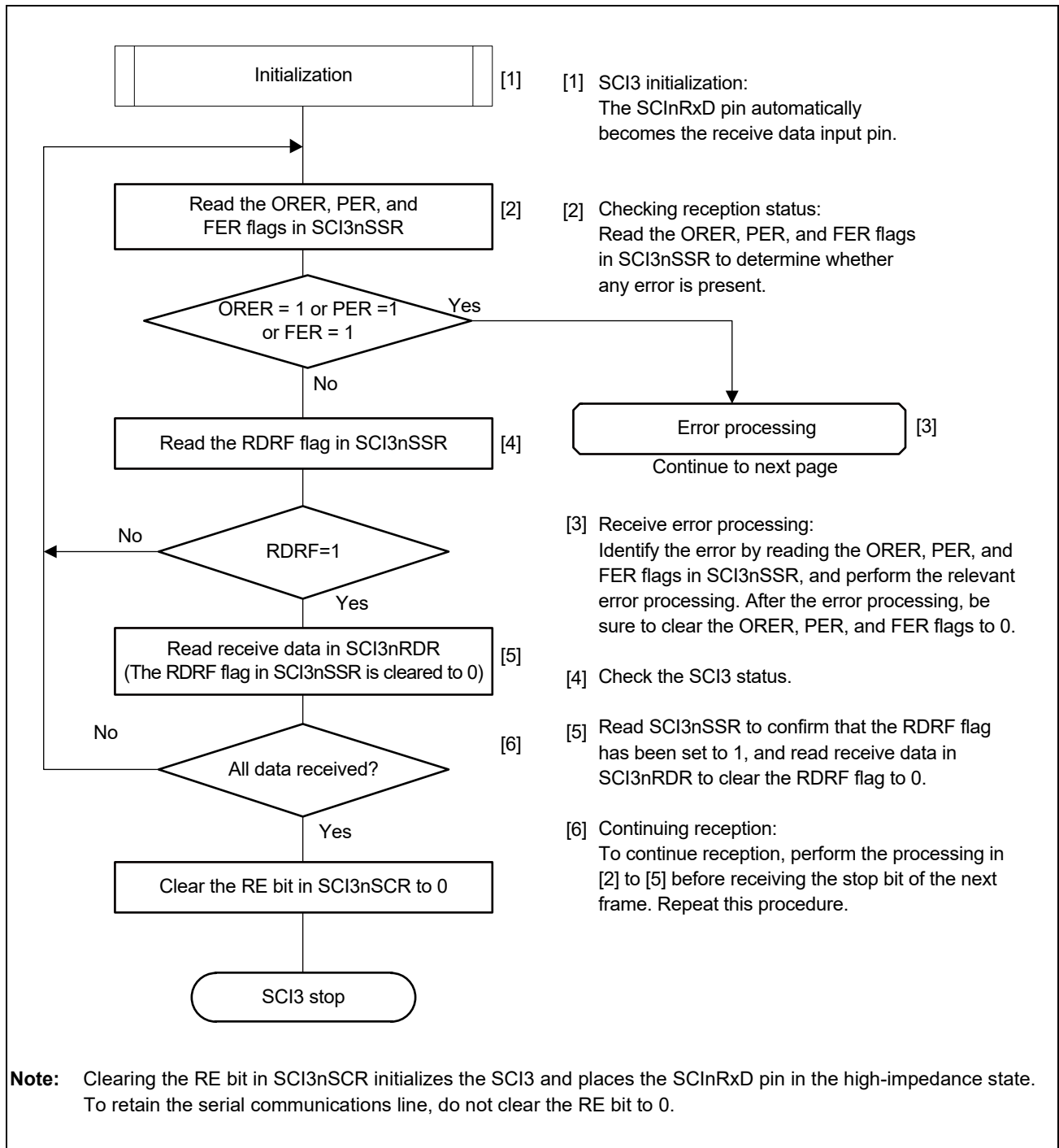


Figure 18.11 Example of Serial Reception Flowchart (1)

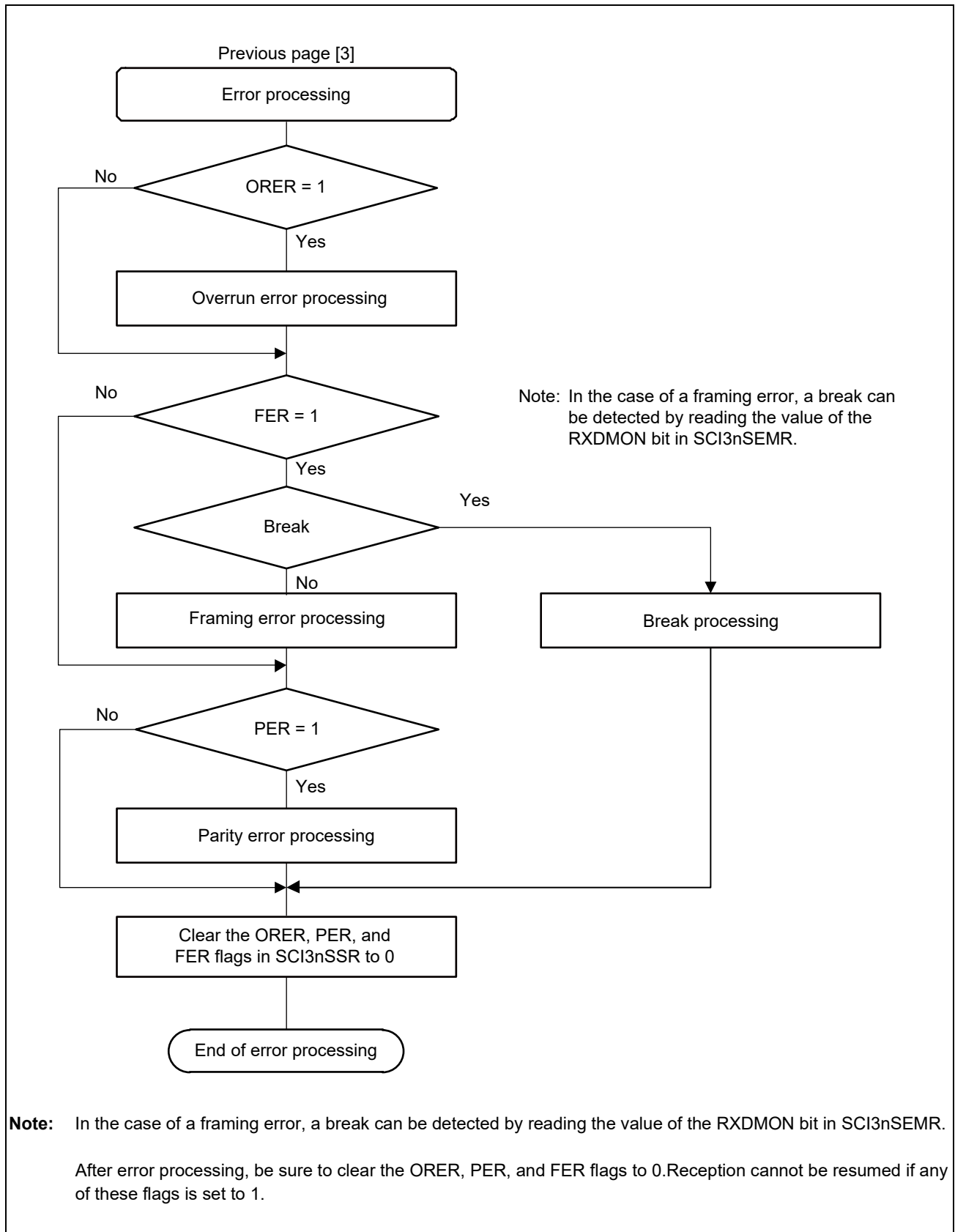


Figure 18.12 Example of Serial Reception Flowchart (2)

18.5 Multi-Processor Communication Function

18.5.1 Overview and Sample Connection

Using the multi-processor communication function allows data transmission and reception by sharing a communication line between multiple processors by using asynchronous serial communication in which the multi-processor bit is added. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station. The multi-processor bit is used to distinguish the ID transmission cycle from the data transmission cycle. When the multi-processor bit is set to 1, it indicates the ID transmission cycle. When the multi-processor bit is set to 0, it indicates the data transmission cycle. **Figure 18.13** shows an example of communication between processors by using the multi-processor format. First, a transmitting station sends communication data in which the multi-processor bit (= 1) is added to the ID code of the receiving station. Next, the transmitting station sends communication data in which the multi-processor bit (= 0) is added to the transmit data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself. When these IDs match, the receiving station receives communication data that is subsequently transmitted. If these IDs do not match, the receiving station skips communication data until it receives communication data in which the multi-processor bit is set to 1.

To support this function, the SCI3 provides the MPIE bit in SCI3nSCR. When the MPIE bit is set to 1, transfer of receive data from SCI3nRSR to SCI3nRDR, detection of a receive error, and setting the RDRF, FER, and ORER flags in SCI3nSSR are disabled until data in which the multi-processor bit is set to 1 is received. Upon receiving a character in which the multi-processor bit is set to 1, the MPB bit in SCI3nSSR is set to 1 and the MPIE bit is automatically cleared to 0, thus returning to a normal reception operation. When the RIE bit in SCI3nSCR is set to 1 at this time, an RXI interrupt request is generated. While the MPIE bit is cleared to 0, reception operation is conducted regardless of the multi-processor bit value. The multi-processor bit is stored in the MPB bit in SCI3nSSR.

When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the normal asynchronous mode. A clock that is used for multi-processor communications is also the same as the clock used in the normal asynchronous mode.

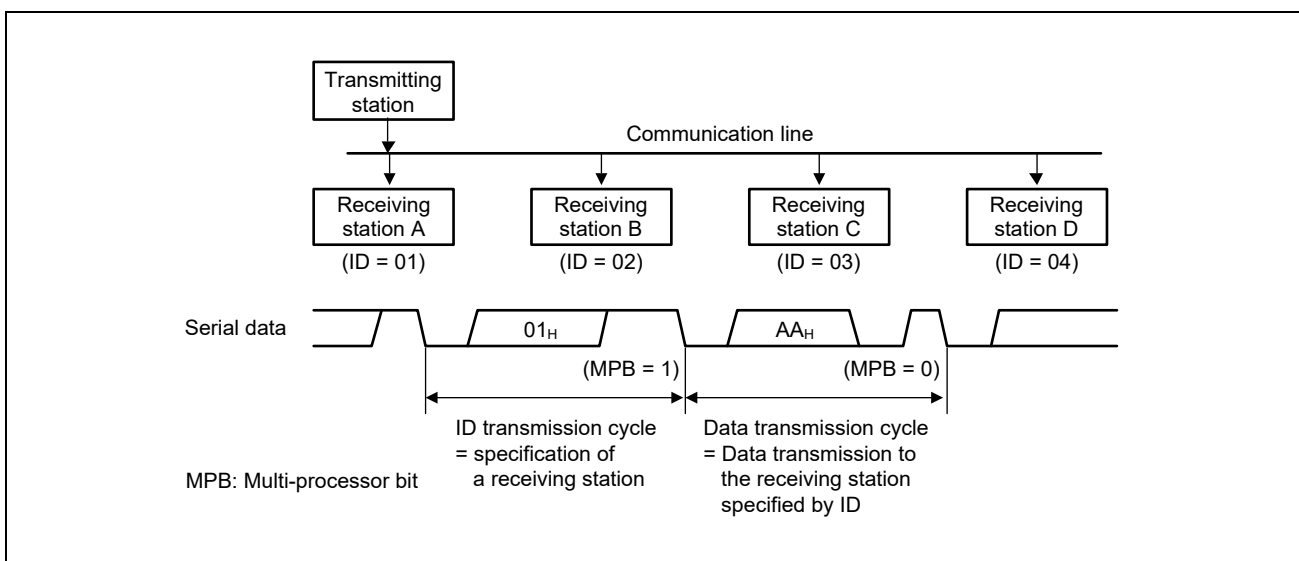


Figure 18.13 Example of Communication Using the Multi-Processor Format (Example of Transmission of Data AA_H to Receiving Station A)

18.5.2 Multi-Processor Serial Data Transmission

Figure 18.14 shows a sample flowchart of multi-processor data processing. In the ID transmission cycle, send the ID with the MPBT bit in SCI3nSSR set to 1. In the data transmission cycle, send data with the MPBT bit in SCI3nSSR cleared to 0. Other operations are the same as operations in asynchronous mode.

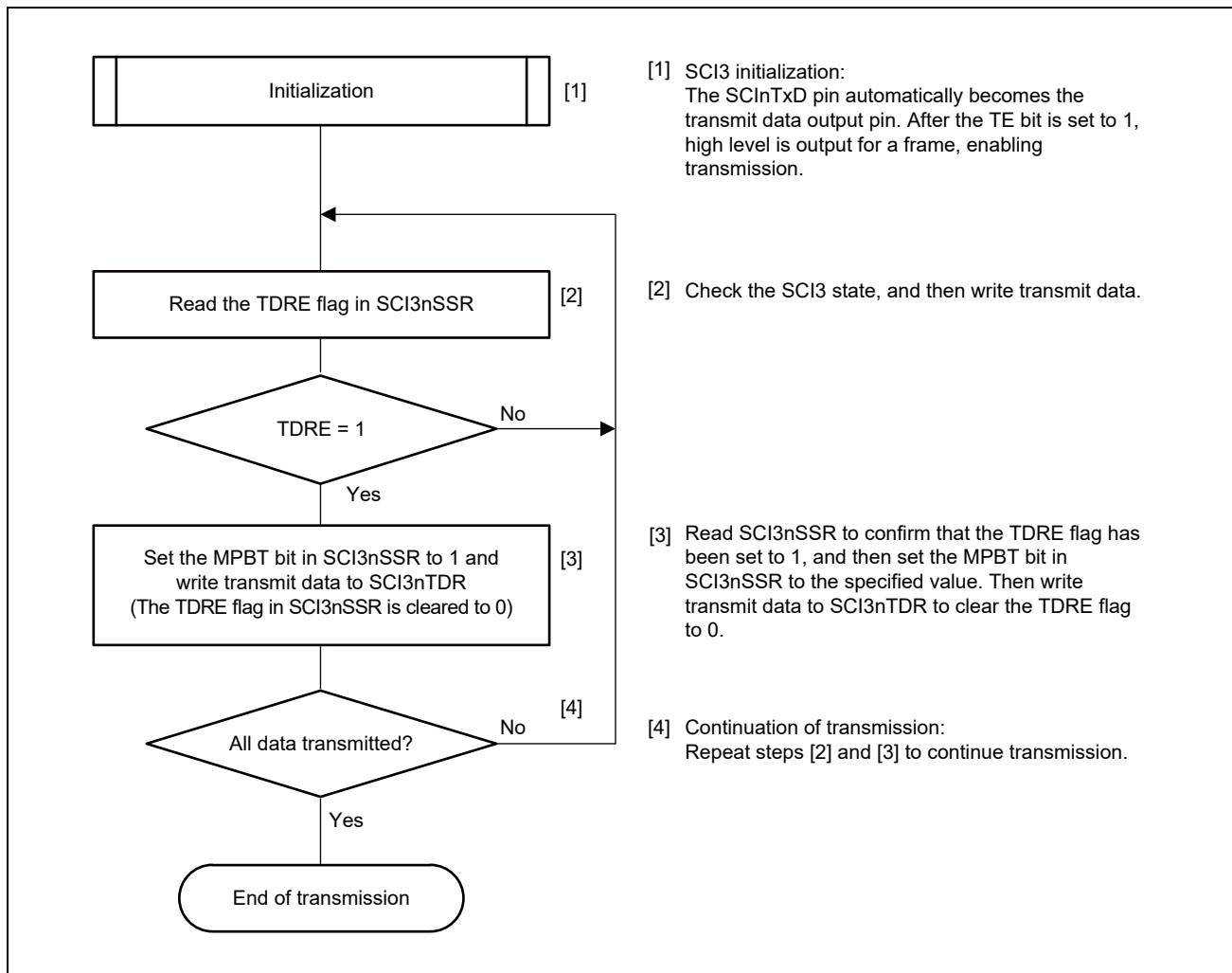


Figure 18.14 Example of Multi-Processor Serial Transmission Flowchart

18.5.3 Multi-Processor Serial Data Reception

Figure 18.16 to Figure 18.18 shows sample flowcharts of multi-processor data reception. When the MPIE bit in SCI3nSCR is set to 1, reading the communication data is skipped until communication data in which the multi-processor bit is set to 1 is received. When communication data in which the multi-processor bit is set to 1 is received, the receive data is transferred to SCI3nRDR. At this time, an RXI interrupt request is generated. Other operations are the same as operations in asynchronous mode. Figure 18.15 shows an example of operation for reception.

CAUTION

Do not write data to SCI3nSCR when communication data in which the multi-processor bit is set to 1 is received. The MPIE bit may not become the desired state.

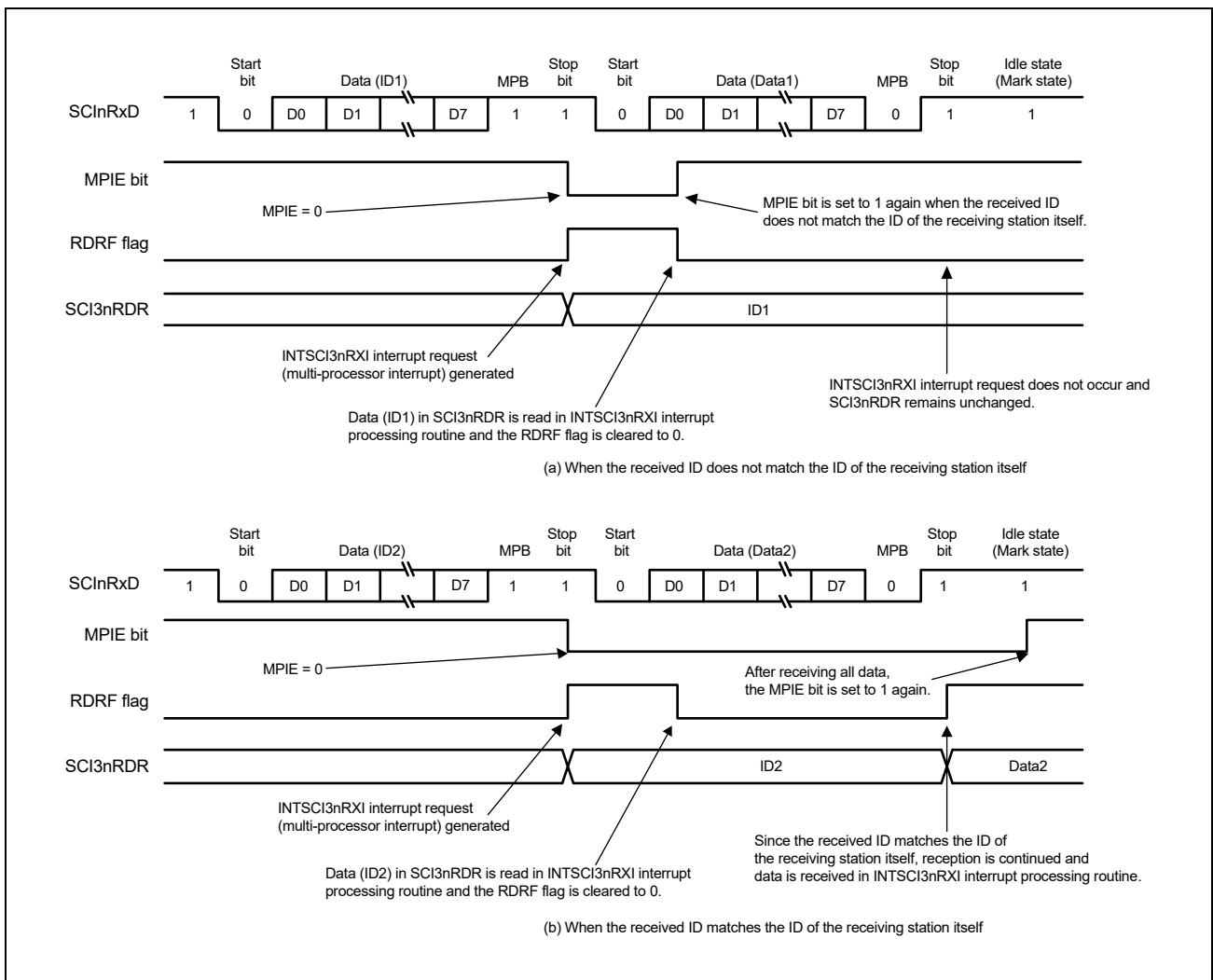


Figure 18.15 Example of SCI3 Reception (8-Bit Data, Multi-Processor Bit, One Stop Bit)

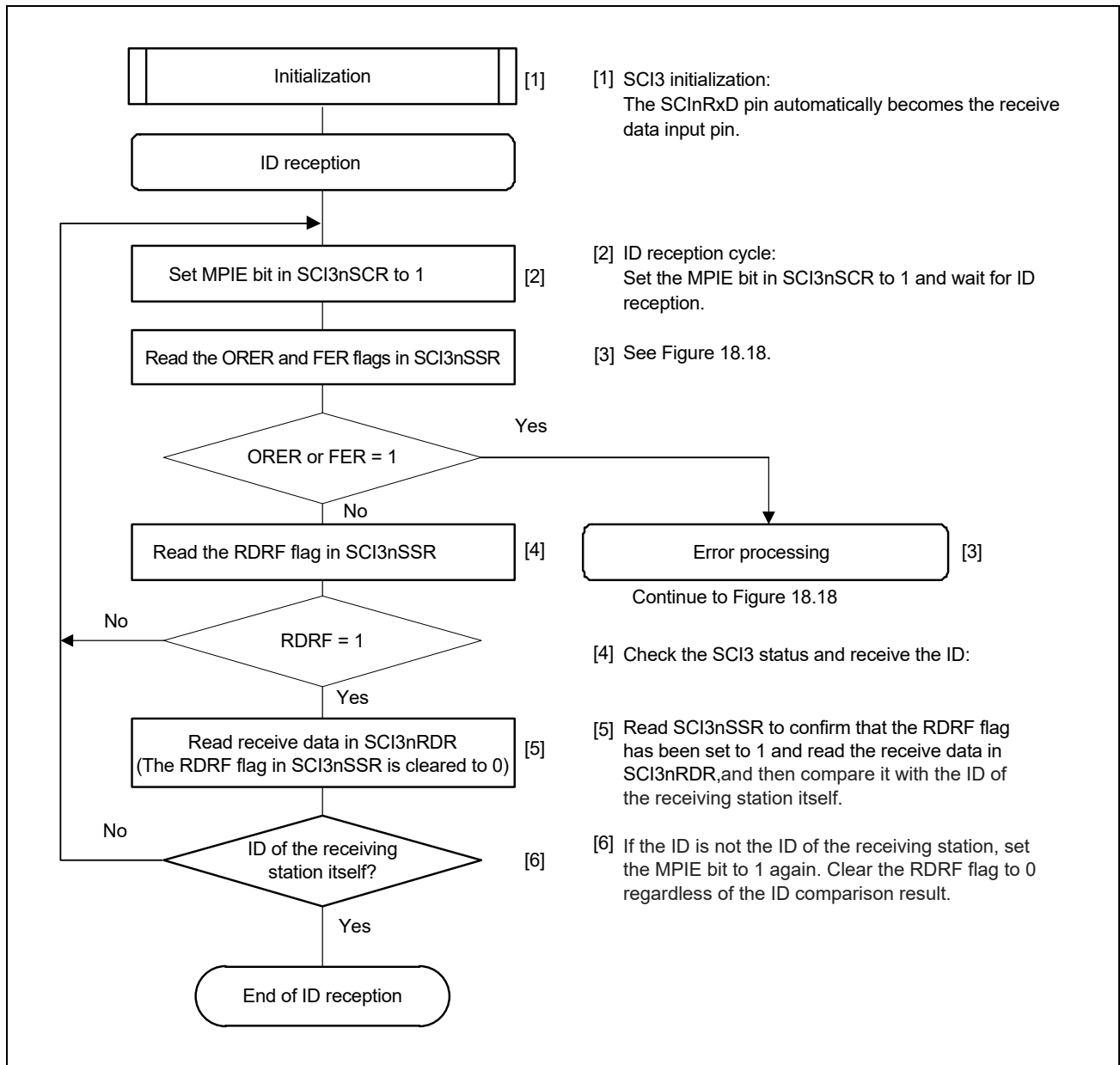
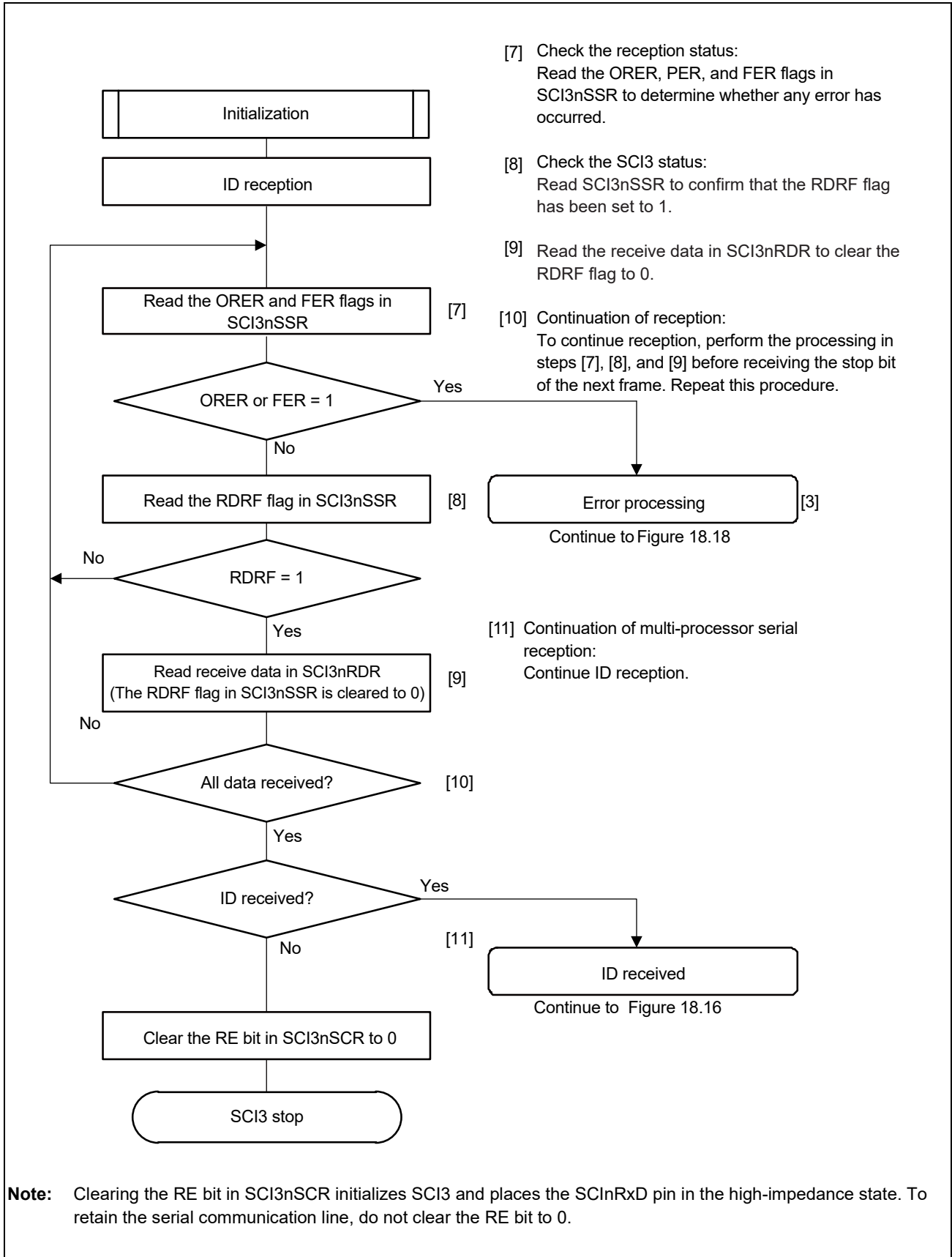


Figure 18.16 Example of Multi-Processor Serial Reception Flowchart (1)



Note: Clearing the RE bit in SCI3nSCR initializes SCI3 and places the SCInRxD pin in the high-impedance state. To retain the serial communication line, do not clear the RE bit to 0.

Figure 18.17 Example of Multi-Processor Serial Reception Flowchart (2)

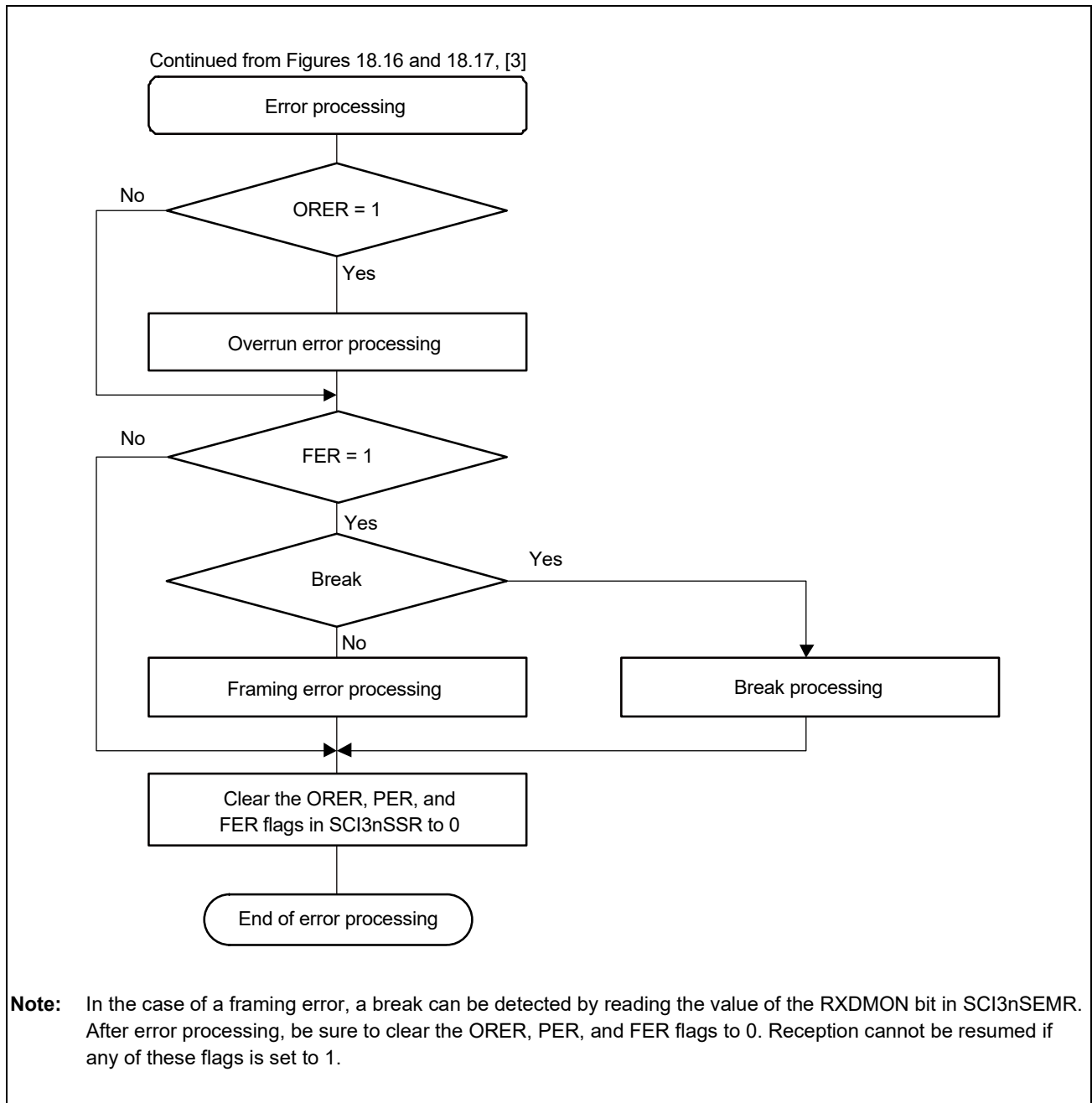


Figure 18.18 Example of Multi-Processor Serial Reception Flowchart (3)

18.6 Operation in Clock Synchronous Mode

Figure 18.19 shows the data format for clock synchronous serial data communication. In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In data transmission with the synchronization clock output, the SCI3 outputs data from one falling edge to the next falling edge of the synchronization clock. In data transmission with the synchronization clock input, the SCI3 outputs the first data (bit 0) after starting transfer immediately after clearing the TDRE bit in SCI3nSSR to 0, and then outputs the next bit data after 2 to 3 ϕ clock cycles from the falling edge of the synchronization clock. In data reception, the SCI3 receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the communication line holds the last-bit output state. In clock synchronous mode, neither parity bit nor multi-processor bit can be added. The transmitter and the receiver are independent in the SCI3, enabling full-duplex communication by using a common clock. Both the transmitter and the receiver have a double-buffered structure so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

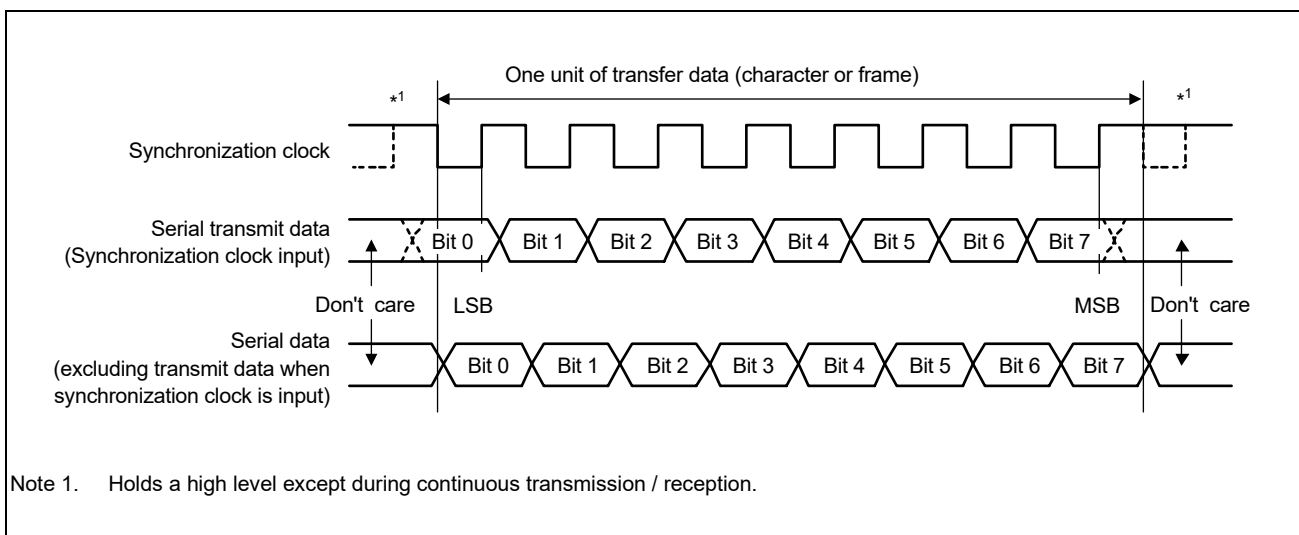


Figure 18.19 Data Format in Clock Synchronous Mode (LSB-First)

18.6.1 Clock

An internal clock generated by the on-chip baud rate generator or an external synchronous clock that is input from the SCInSCK pin can be selected by the setting of the CKE1 and CKE0 bits in SCI3nSCR. When operating the SCI3 on the internal clock, a synchronous clock is output from the SCInSCK pin. Eight pulses of the synchronous clock are output during transfer of one character, and the clock is held high while no data is transferred.

18.6.2 SCI3 Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, clear the TE and RE bits in SCI3nSCR to 0 and then initialize the SCI3 according to the sample flowchart in **Figure 18.20**. To switch the operation between transmission, reception, and transmission / reception, clear the TE and RE bits to 0 and then set these bits to the desired value. Before changing the transfer format, be sure to clear the TE and RE bits to 0. Note that clearing the TE bit to 0 sets the TDRE flag to 1, but clearing the RE bit to 0 initializes neither the RDRF, PER, FER, and ORER flags nor SCI3nRDR.

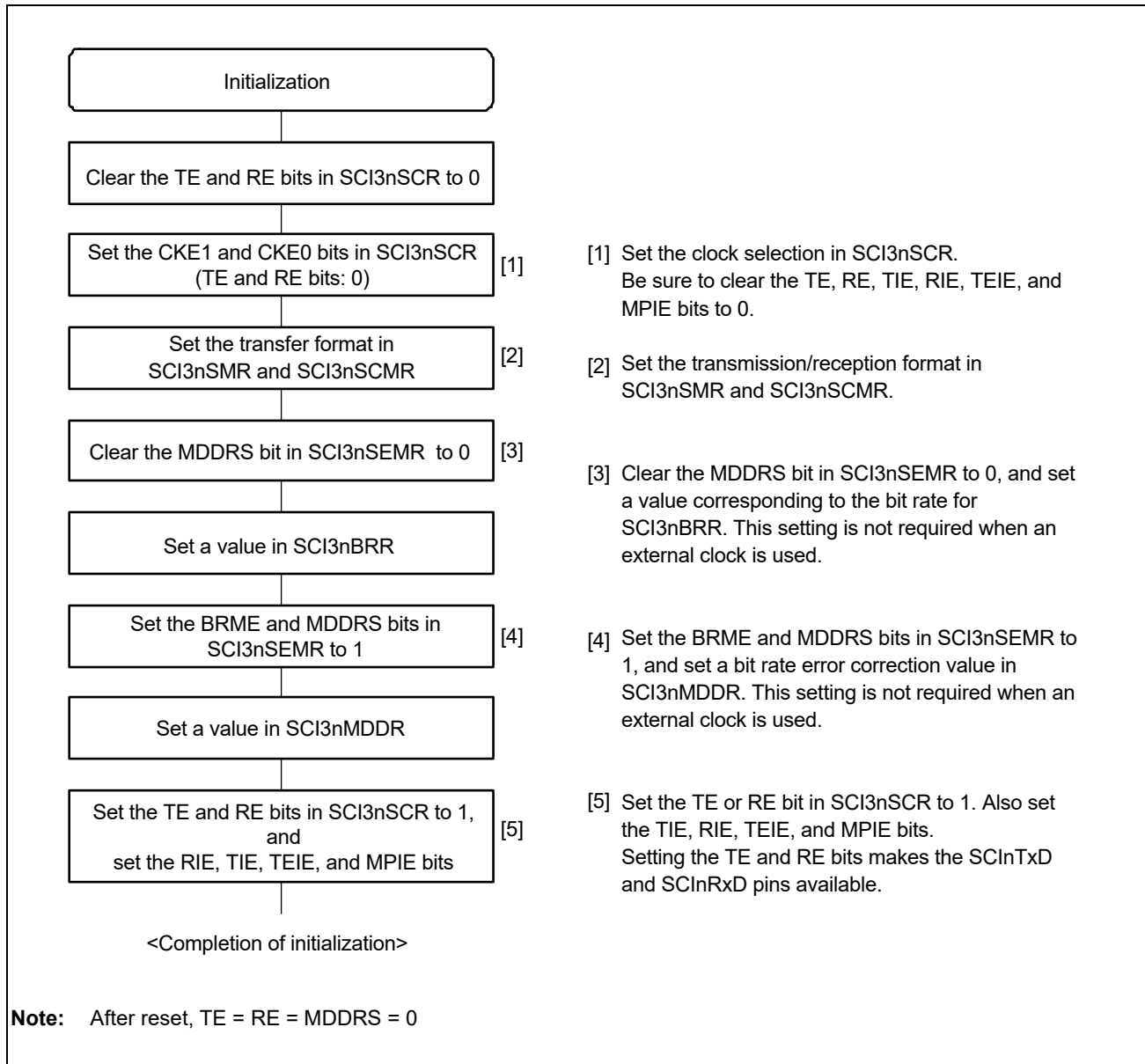


Figure 18.20 Example of SCI3 Initialization Flowchart

18.6.3 Serial Data Transmission (Clock Synchronous Mode)

Figure 18.21 shows an example of the operation for serial transmission in clock synchronous mode. In serial data transmission, the SCI3 operates as described below.

1. When transmit data is written to SCI3nTDR, the TDRE flag is automatically cleared to 0. The SCI3 monitors the TDRE flag in SCI3nSSR. When the flag is cleared, the SCI3 recognizes that data has been written to SCI3nTDR and transfers data from SCI3nTDR to SCI3nTSR, starting output of the first bit when the synchronization clock is input. To write transmit data to SCI3nTDR at a trigger of TXI interrupt request, set the TIE bit to 1 and then set the TE bit to 1 or set both TIE and TE bits simultaneously with one instruction to generate a TXI interrupt request for starting data transfer.
2. Transmission starts after data is transferred from SCI3nTDR to SCI3nTSR, and the TDRE flag is set to 1. When the TIE bit in SCI3nSCR is set to 1 at this time, a TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to SCI3nTDR in this TXI interrupt processing routine before the transmission of the previously transferred data is completed. When a TEI interrupt request is used, the TIE bit is cleared to 0 after the last transmit data has been written to SCI3nTDR, and the TEIE bit is set to 1.
3. 8-bit data is output from the SCInTxD pin in synchronization with the output clock (when clock output mode has been specified) or in synchronization with the input clock (when external clock has been specified).
4. The SCI3 checks the TDRE flag when the last bit is output.
5. When the TDRE flag is 0, the next transmit data is transferred from SCI3nTDR to SCI3nTSR, and serial transmission of the next frame starts.
6. When the TDRE flag is 1, the TEND flag in SCI3nSSR is set to 1 and the SCInSCK pin retains the output state of the last bit. If the TEIE bit in SCI3nSCR is set to 1 at this time, a TEI interrupt request is generated. The SCInSCK pin is held high.

Figure 18.22 shows a sample flowchart of serial data transmission. Also, **Figure 18.23** shows a sample flowchart for stopping the SCI3 after data transmission. Transmission will not start even if the TDRE flag is cleared while a receive error flag (ORER) is set to 1. Be sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.

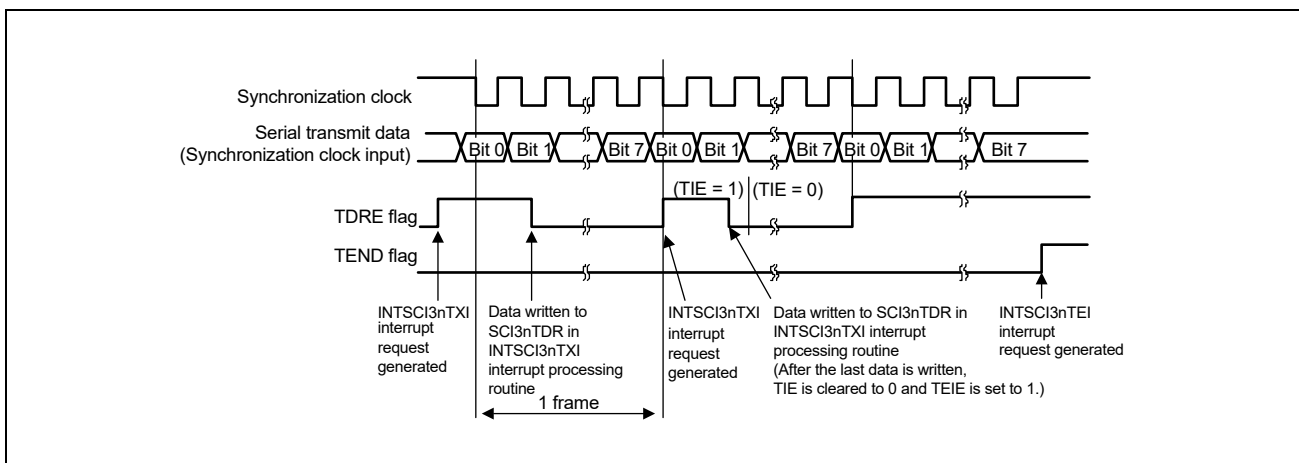


Figure 18.21 Example of Operation for Transmission in Clock Synchronous Mode

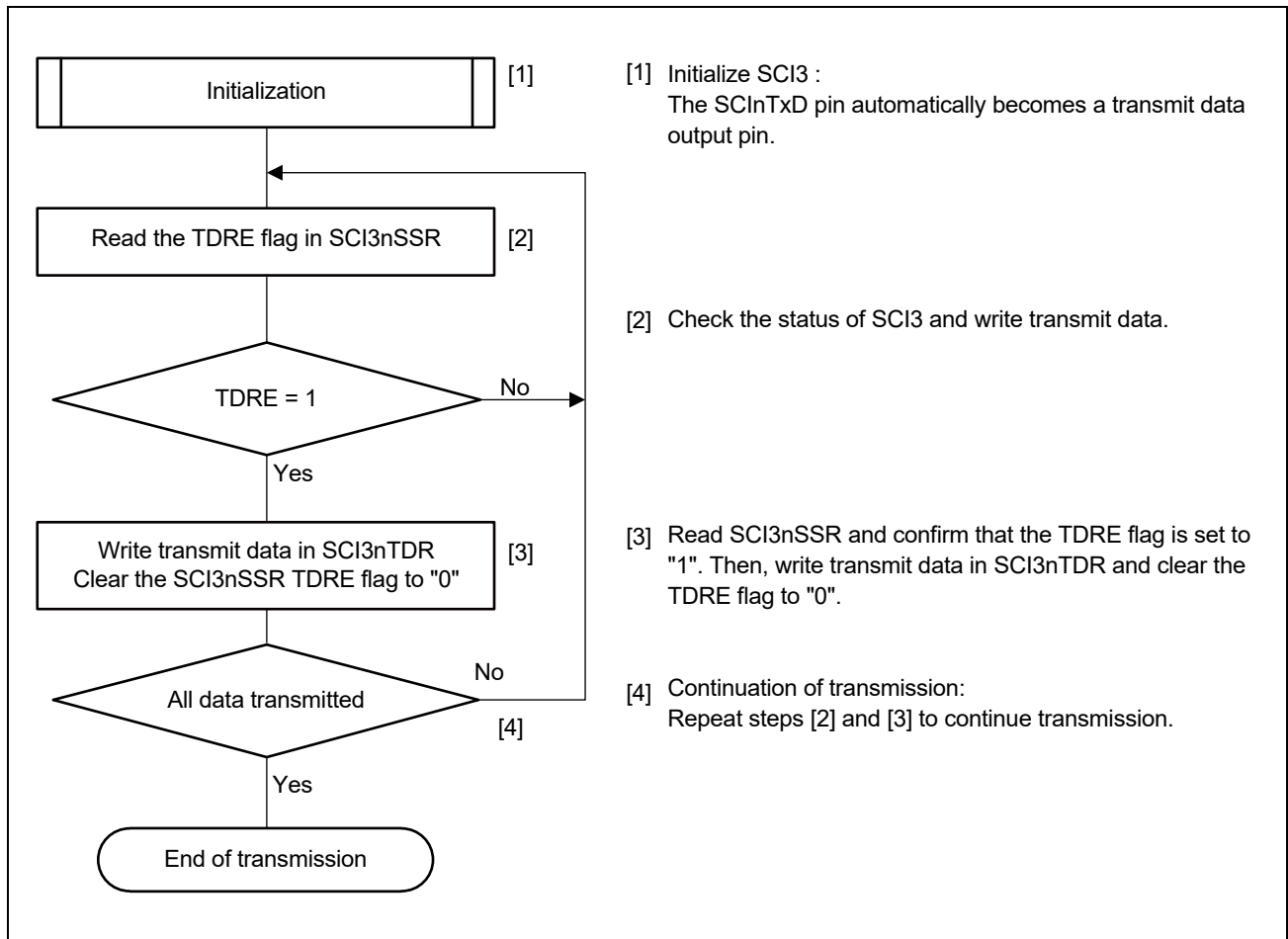


Figure 18.22 Example of Serial Transmission Flowchart

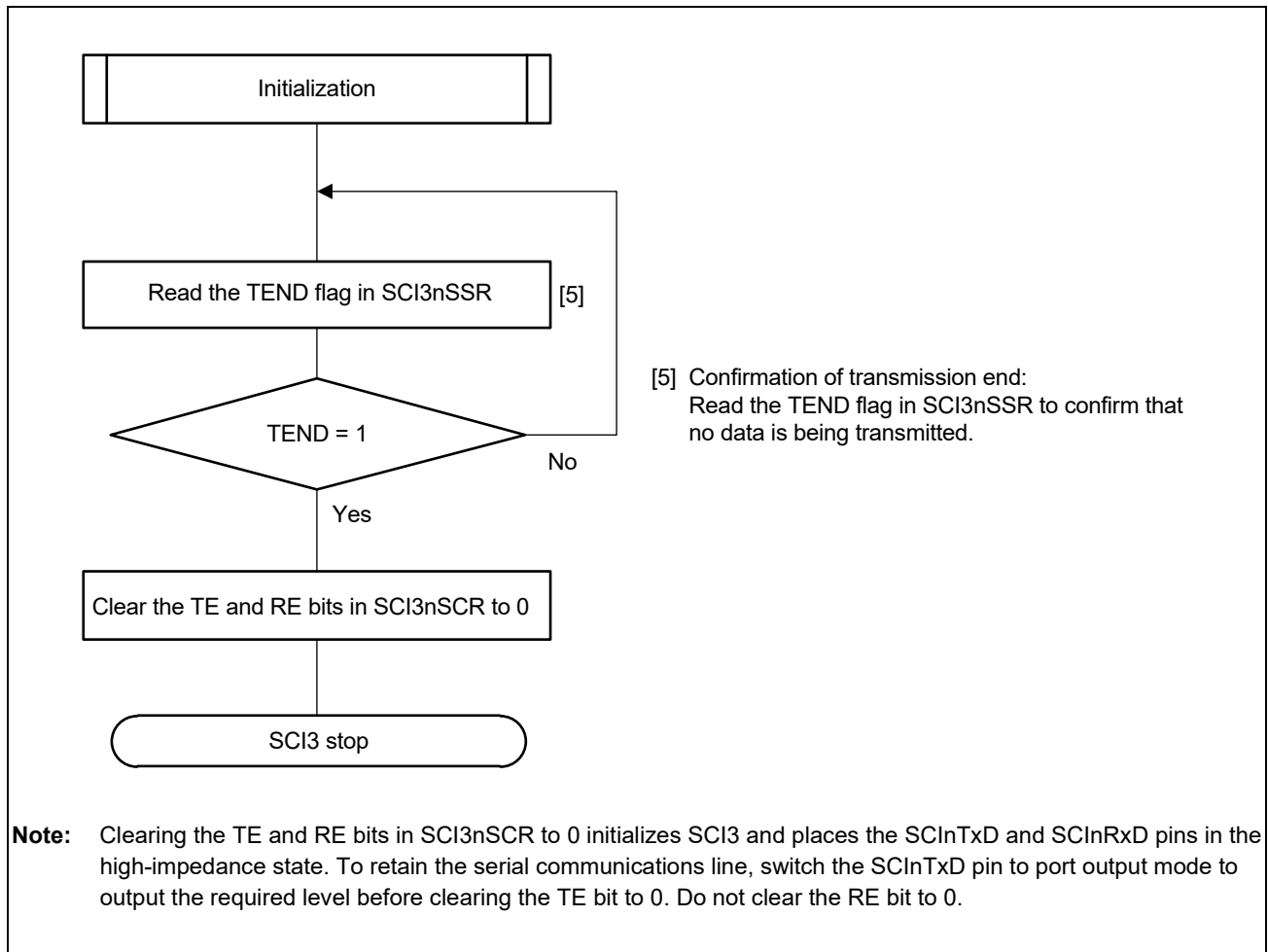


Figure 18.23 Example Flowchart for Stopping the SCI3 after Serial Transmission

18.6.4 Serial Data Reception (Clock Synchronous Mode)

Figure 18.24 shows an example of SCI3 operation for serial reception in clock synchronous mode. In serial data reception, the SCI3 operates as described below.

1. The SCI3 performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores receive data in SCI3nRSR.
2. When an overrun error occurs (the reception of the next data is completed with the RDRF flag in SCI3nSSR set to 1), the ORER flag in SCI3nSSR is set to 1. If the RIE bit in SCI3nSCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to SCI3nRDR. The RDRF flag retains the state of being set to 1.
3. When data has been successfully received, the RDRF flag in SCI3nSSR is set to 1 and the receive data is transferred to SCI3nRDR. When the RIE bit in SCI3nSCR is set to 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to SCI3nRDR in this RXI interrupt processing routine before reception of the next data is completed. Reading SCI3nRDR automatically clears the RDRF flag to 0.

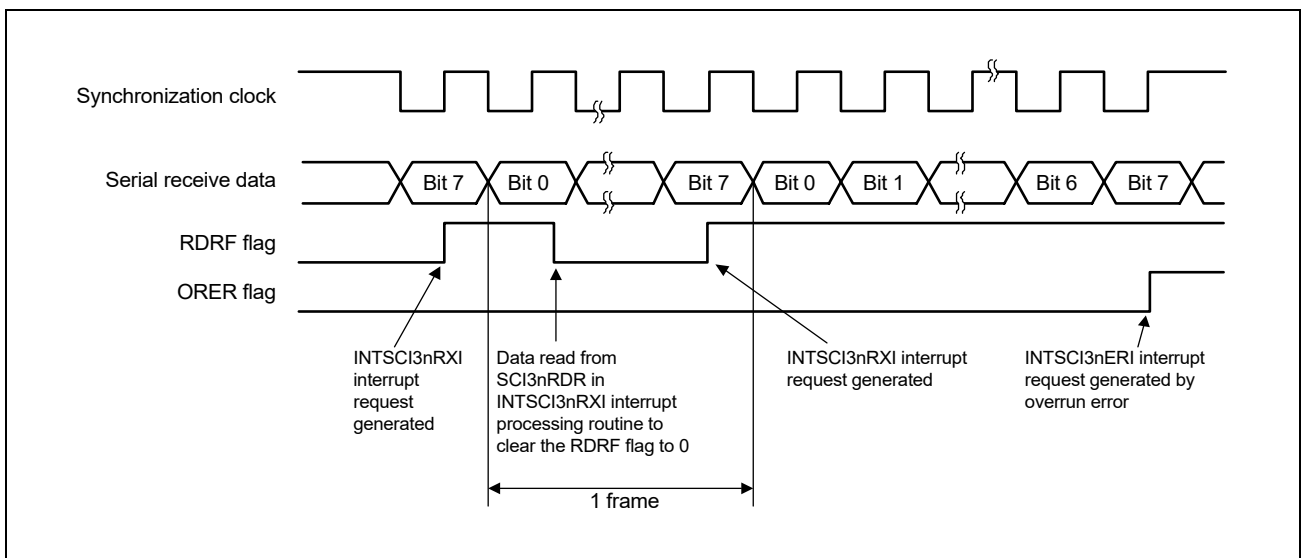


Figure 18.24 Example of SCI3 Operation for Reception

Subsequent transmission and reception are disabled with a receive error flag set to 1. Therefore, be sure to clear the ORER, FER, PER, and RDRF flags before continuing reception. **Figure 18.25** shows an example of flowchart for data reception.

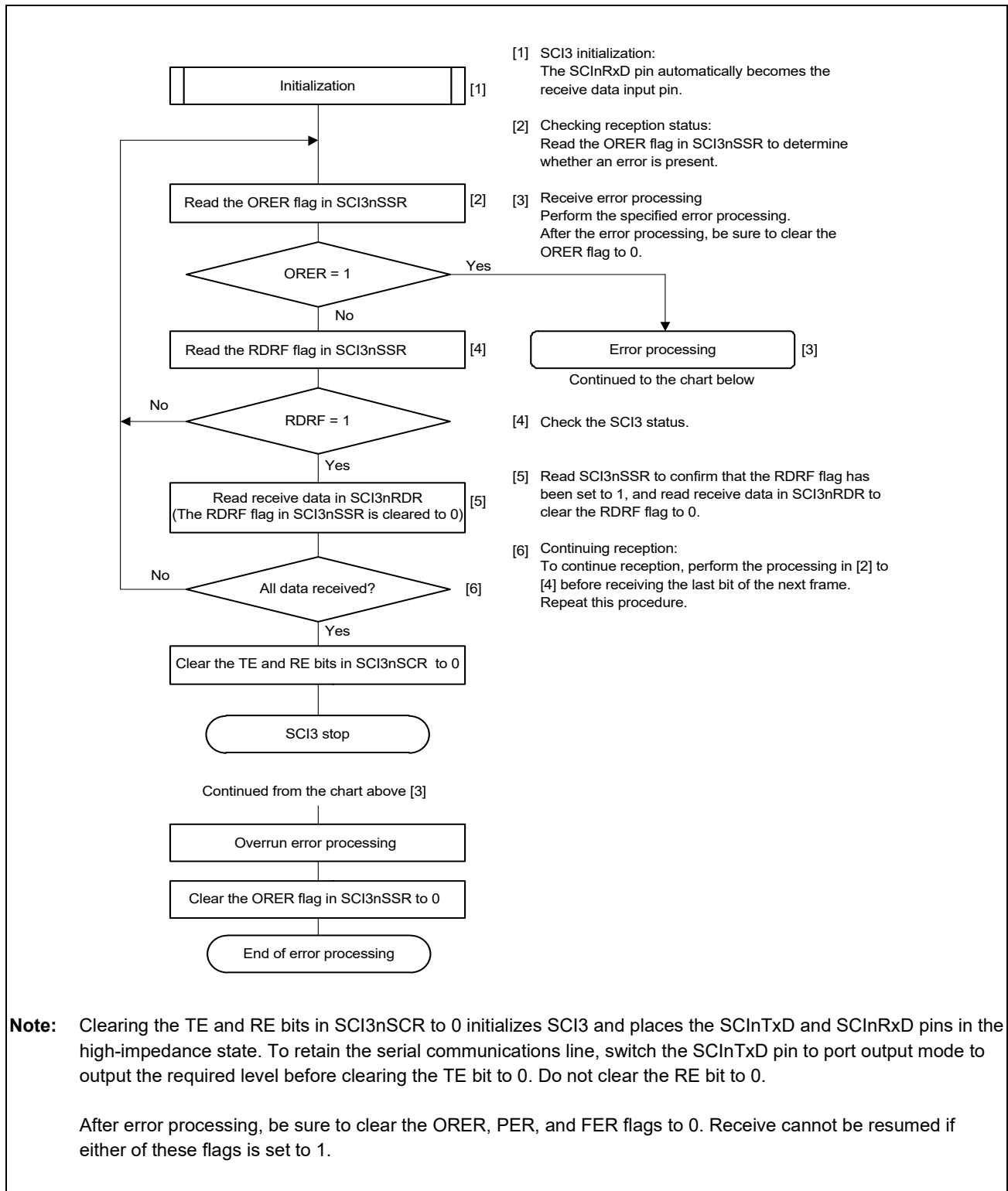


Figure 18.25 Example of Serial Reception Flowchart

18.6.5 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 18.26 shows a sample flowchart for simultaneous data transmit and receive operations. After the SCI3 is initialized, perform the following procedure for simultaneous data transmit and receive operations.

1. To switch from transmit mode to simultaneous transmit and receive mode, check that the SCI3 has finished transmission and the TDRE and TEND flags are set to 1. Then clear the TE bit to 0 and then set the TE and RE bits to 1 with a single instruction.
2. To switch from receive mode to simultaneous transmit and receive mode, check that the SCI3 has finished reception and clear the RE bit to 0. Then check that the RDRF and error flags (ORER, FER, and PER) are cleared to 0 and then set the TE and RE bits to 1 with a single instruction.

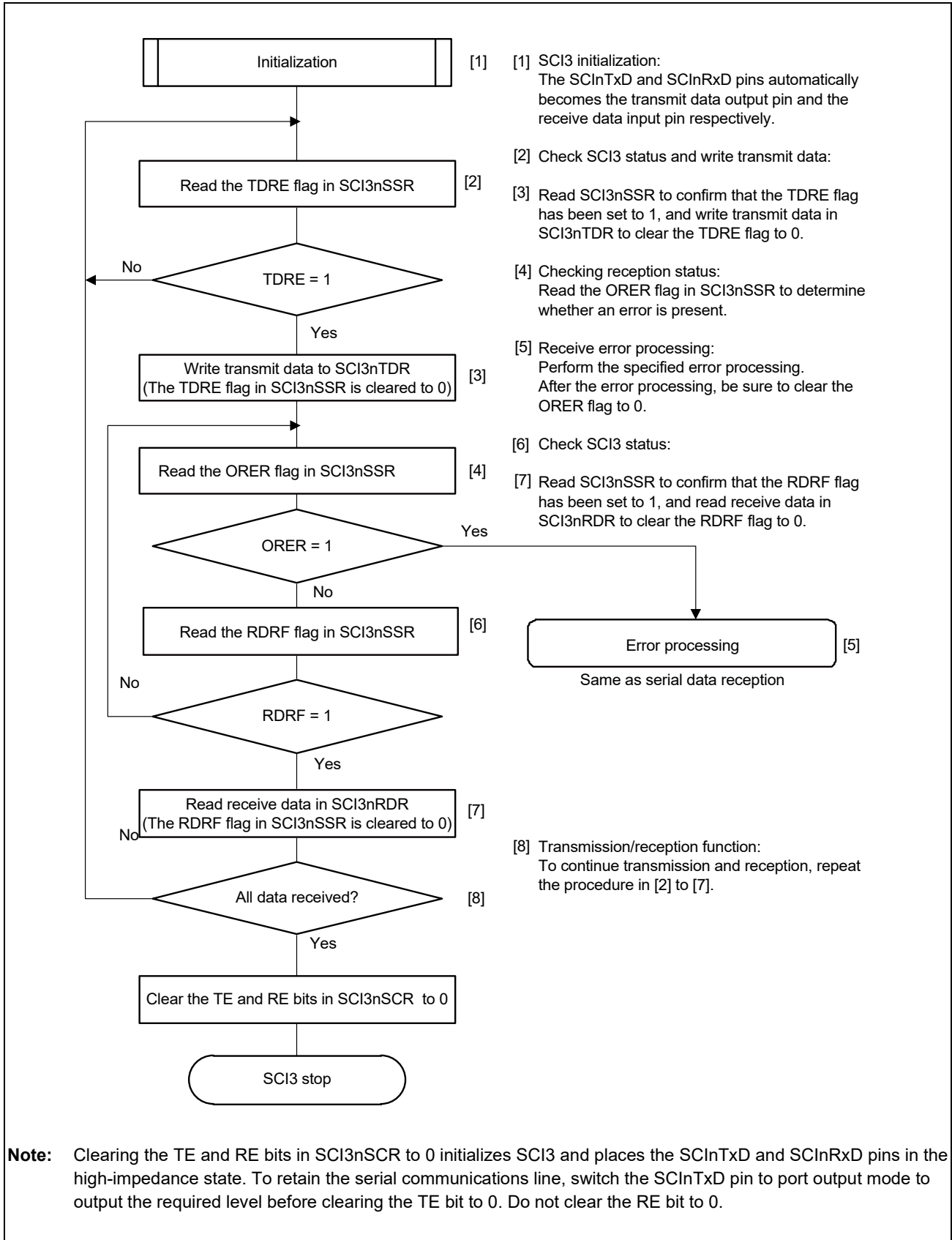


Figure 18.26 Example of Simultaneous Serial Transmission and Reception Flowchart

18.7 Bit Rate Modulation Function

The bit rate modulation function corrects a bit rate by averagely enabling the internal clocks specified by the CKS1 and CKS0 bits in SCI3nSMR for the number specified by SCI3nMDDR out of 256 clocks.

Figure 18.27 shows an example of asynchronous mode in which ϕ clock is selected with the CKS1 and CKS0 bits, SCI3nBRR is set to 0, and SCI3nMDDR is set to 160. In this example, the reference clock cycle is corrected to $256/160$ on average and the bit rate is corrected to $160 / 256$. Note that the pulse widths of the internal reference clock expand or contract for the amount of the selected internal clocks because there is a deviation in the enabling of the internal clocks.

Do not use this function with the maximum speed settings (CKS1 and CKS0 bits = 0 in SCI3nSMR, CKE1 bit = 0 in SCI3nSCR, and SCI3nBRR = 0) for clock synchronous mode.

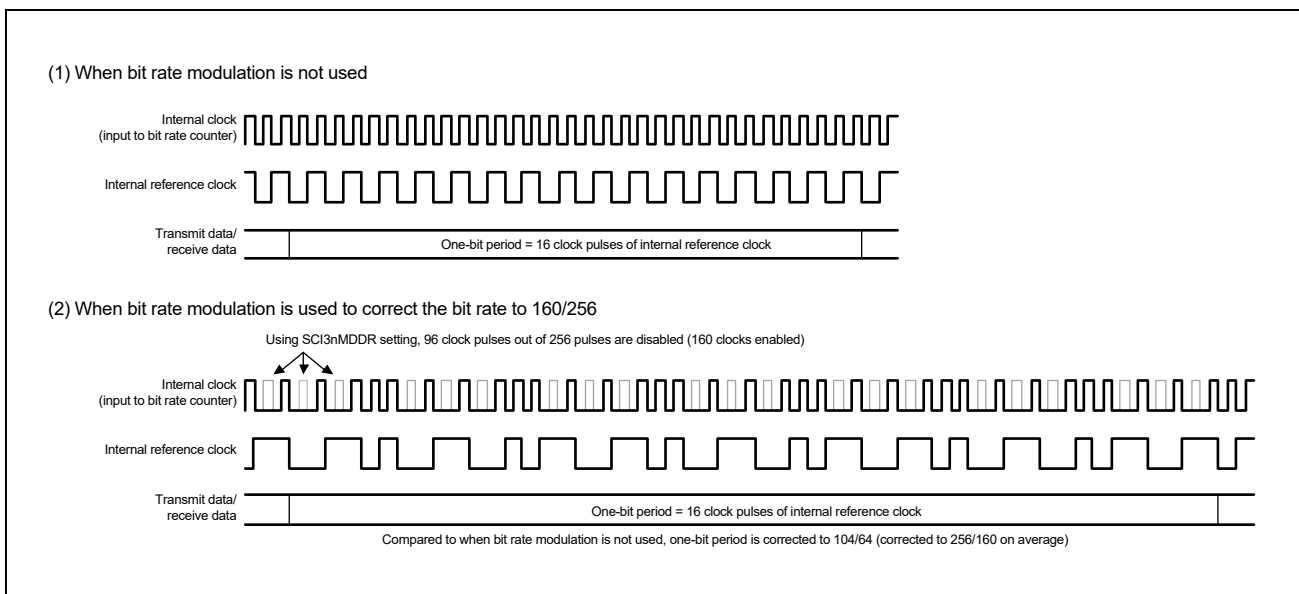


Figure 18.27 Example of Internal Reference Clock when the Bit Rate Modulation Function is Used

18.8 Interrupt Sources

Table 18.24 lists interrupt sources. Each interrupt source outputs an individual interrupt request signal. These interrupt sources can be enabled independently with the enable bit in SCI3nSCR.

A TXI interrupt request is generated when the TDRE flag in SCI3nSSR is set to 1.

A TEI interrupt request is generated when the TEND flag in SCI3nSSR is set to 1.

A TXI interrupt request can activate the DMA to handle data transfer. The TDRE flag is automatically cleared to 0 when data is transferred using the DMA.

CAUTION

The TDRE flag and TEND flag cannot be cleared to 0 when the TE bit in SCI3nSCR is 0. Do not set the TEIE bit in SCI3nSCR to 1 when the TE bit is set to 0 because the TEND flag is a level interrupt request flag of TEI interrupt.

An RXI interrupt request is generated when the RDRF flag in SCI3nSSR is set to 1.

An ERI interrupt is generated when any of the ORER, PER, and FER flags in SCI3nSSR is set to 1.

An RXI interrupt request can activate the DMA to handle data transfer. The RDRF flag is automatically cleared to 0 when data is transferred using the DMA.

A TEI interrupt request is generated when the TEND flag is set to 1 with the TEIE bit set to 1.

CAUTION

If a TEI interrupt request and a TXI interrupt request are generated at the same time, the TXI interrupt request is accepted first. Note that clearing the TDRE flag to 0 at this time in the TXI interrupt processing routine also clears the TEND flag to 0 automatically, disabling the branch to the TEI interrupt processing routine.

Table 18.24 SCI3 Interrupt Sources

Name	Interrupt Source	Interrupt Flag	sDMA / DTS Activation
ERI / TEI	Receive error / Transmit end	ORER, FER, PER / TEND	Not possible
RXI	Receive data full	RDRF	Possible
TXI	Transmit data empty	TDRE	Possible

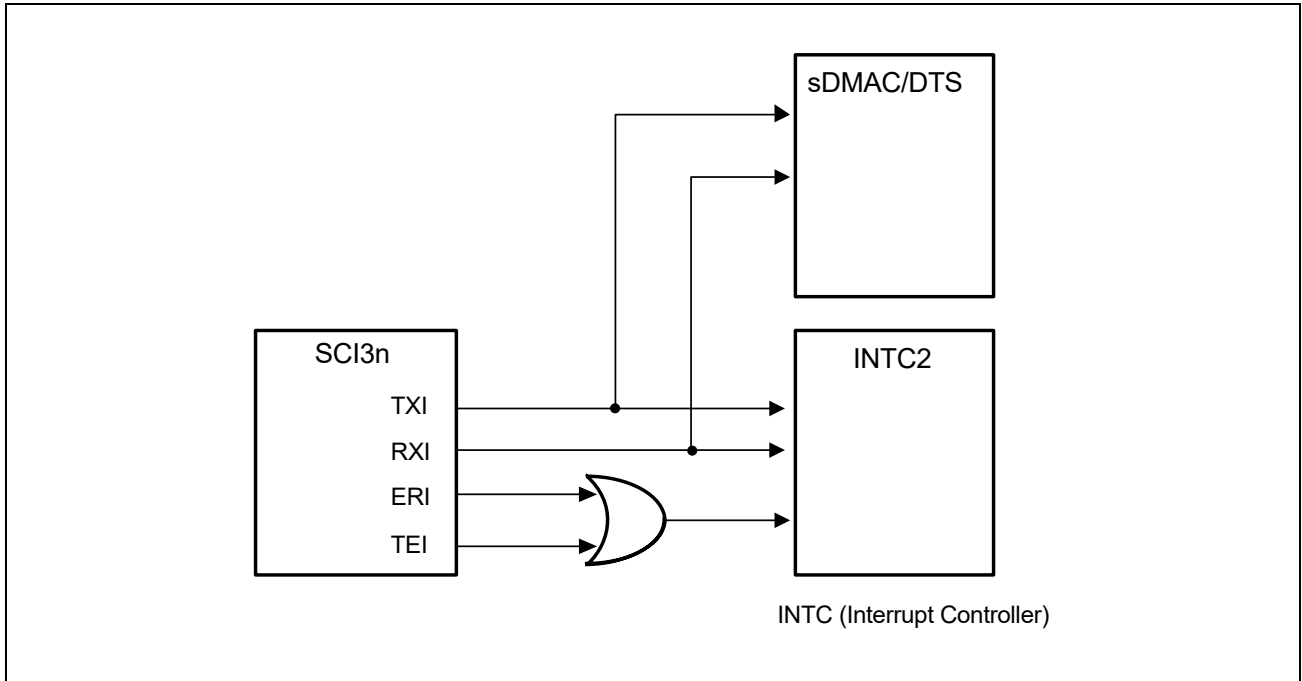


Figure 18.28 Relationship between the SCI3 Interrupt Signals, Interrupt Controller (INTC) and sDMAC / DTS

18.9 Usage Notes

18.9.1 Break Detection and Processing

A break can be detected by reading the RXDMON bit in SCI3nSEMR when detecting a framing error. Since all inputs from the SCInRxD pin are 0 in the break state, the FER flag is set to 1 and the PER flag may also be set to 1. The SCI3 continues data reception even after it receives a break. For this reason, note that FER flag is set to 1 again even after the FER flag is cleared to 0.

18.9.2 Mark State and Production of Breaks

While the TE bit is 0 (transmission/reception disabled), the SCInTxD pin can output any level by switching the SCInTxD pin to a general output port. This allows the SCInTxD pin to be in mark state or break output during data transmission.

18.9.3 Receive Error Flags and Transmit Operations in Clock Synchronous Mode

During the clock synchronous simultaneous data transmit / receive operation, transmission cannot be started when a receive error flag (ORER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note that the receive error flags cannot be cleared to 0 even by clearing the RE bit to 0.

18.9.4 Relationship between Writing to SCI3nTDR and the TDRE Flag

The TDRE flag in SCI3nSSR is a status flag indicating that transmit data in SCI3nTDR has been transferred to SCI3nTSR. The TDRE flag is set to 1 when the SCI3 transfers data from SCI3nTDR to SCI3nTSR.

Data can be written to SCI3nTDR regardless of the status of the TDRE flag. However, writing new data to SCI3nTDR while the TDRE flag is 0 will cause the data stored in SCI3nTDR to be lost because it has not been transferred to SCI3nTSR. Therefore, make sure to check that the TDRE flag is set to 1 before writing transmit data to SCI3nTDR.

18.9.5 Restrictions on Using an External Clock for Transmission in Clock Synchronous Mode

When using an external synchronization clock, clear the TDRE flag to 0 and then input a transmission clock (Figure 18.29). Also in continuous transmission mode, clear the TDRE flag to 0 and then input a transmission clock for the next frame.

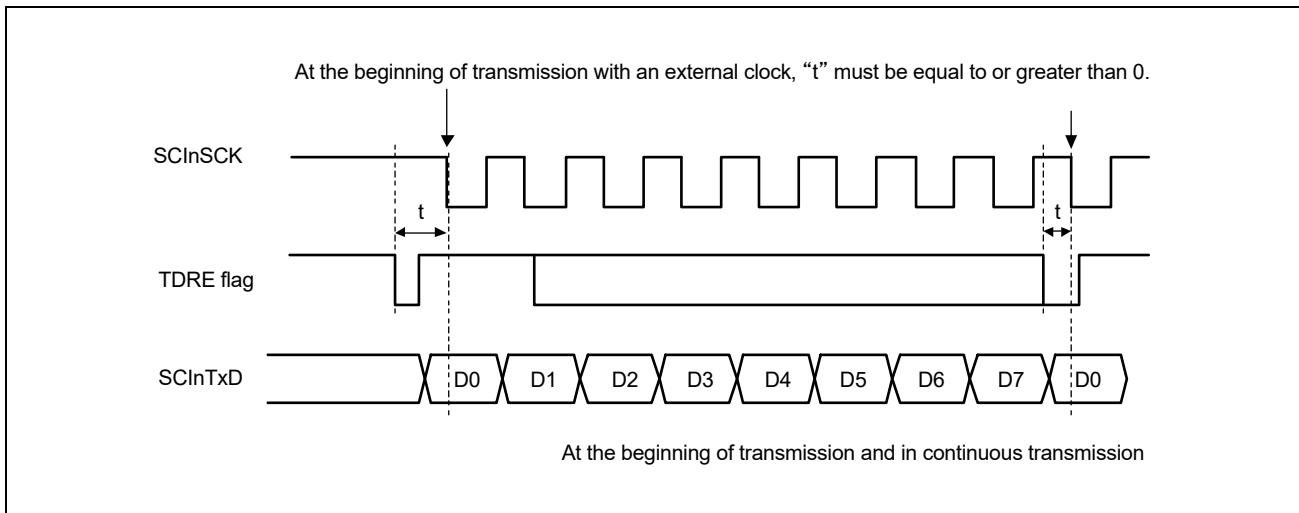


Figure 18.29 Restrictions on Using an External Clock for Transmission in Clock Synchronous Mode

18.9.6 External Clock Input in Clock Synchronous Mode

In clock synchronous mode, set at least two cycles of the external SCInSCK clock input for the widths at both high and low levels, and at least six cycles as the period.

Section 19 LIN/UART Interface (RLIN3)

This section contains a generic description of the LIN/UART interface (RLIN3).

The first part of this section describes all this product specific properties, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of RLIN3.

19.1 Features of RLIN3

19.1.1 Units and Channels

This microcontroller has the following number of RLIN3 units.

Each RLIN3 unit has one channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 19.1 Number of Channels (For E2x-FCC1)

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Channels	6	6
Name	RLIN3n (n = 0 to 5)	

Table 19.2 Number of Channels (For E2M)

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Channels	6	6
Name	RLIN3n (n = 0 to 5)	

Note: The channel names are same as those of the corresponding units.

Table 19.3 Indices

Index	Meaning
n	Throughout this section, the individual RLIN3 units are identified by the index “n” (“n” is channel numbers.); for example, the LIN control register of RLIN3n is described as RLIN3nLCUC.
b	Throughout this section, the individual transmit/receive data buffers of RLIN3n are identified by the index “b” (b = 1 to 8); for example, the data buffer register of the index “b” is described as RLIN3nLDBRb.

19.1.2 Register Base Addresses

RLIN3 base addresses are listed in the following table.

RLIN3 register addresses are given as offsets from the base addresses.

Table 19.4 Register Base Addresses

Base Address Name	Base Address	Bus Group
<RLIN30_base>	FFD2 0000H	Peripheral Group 3
<RLIN31_base>	FFD2 0040H	Peripheral Group 3
<RLIN32_base>	FFD2 0080H	Peripheral Group 3
<RLIN33_base>	FFD2 00C0H	Peripheral Group 3
<RLIN34_base>	FFD2 0100H	Peripheral Group 3
<RLIN35_base>	FFD2 0140H	Peripheral Group 3

19.1.3 Clock Supply

The RLIN3 clock supply is shown in the following table.

Table 19.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
RLIN3n	LIN communication clock source	CLK_LSB
	Register access clock	CLK_HSB

19.1.4 Interrupt Requests

RLIN3 interrupt requests are listed in the following table.

Table 19.6 Interrupt Requests

Interrupt symbol name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
RLIN30					
INTRLIN30UR0	INTRLIN3nUR0 (n = 0)	RLIN30 transmit interrupt	404	group0-153	group1-61
INTRLIN30UR1	INTRLIN3nUR1 (n = 0)	RLIN30 receive completion interrupt	405	group0-154	group1-62
INTRLIN30UR2	INTRLIN3nUR2 (n = 0)	RLIN30 status interrupt	406	—	—
RLIN31					
INTRLIN31UR0	INTRLIN3nUR0 (n = 1)	RLIN31 transmit interrupt	407	group0-155	group1-63
INTRLIN31UR1	INTRLIN3nUR1 (n = 1)	RLIN31 receive completion interrupt	408	group0-156	group1-64
INTRLIN31UR2	INTRLIN3nUR2 (n = 1)	RLIN31 status interrupt	409	—	—
RLIN32					
INTRLIN32UR0	INTRLIN3nUR0 (n = 2)	RLIN32 transmit interrupt	410	group0-157	group1-65
INTRLIN32UR1	INTRLIN3nUR1 (n = 2)	RLIN32 receive completion interrupt	411	group0-158	group1-66
INTRLIN32UR2	INTRLIN3nUR2 (n = 2)	RLIN32 status interrupt	412	—	—
RLIN33					
INTRLIN33UR0	INTRLIN3nUR0 (n = 3)	RLIN33 transmit interrupt	413	group0-159	group1-67
INTRLIN33UR1	INTRLIN3nUR1 (n = 3)	RLIN33 receive completion interrupt	414	group0-160	group1-68
INTRLIN33UR2	INTRLIN3nUR2 (n = 3)	RLIN33 status interrupt	415	—	—
RLIN34					
INTRLIN34UR0	INTRLIN3nUR0 (n = 4)	RLIN34 transmit interrupt	416	group0-161	group1-69
INTRLIN34UR1	INTRLIN3nUR1 (n = 4)	RLIN34 receive completion interrupt	417	group0-162	group1-70
INTRLIN34UR2	INTRLIN3nUR2 (n = 4)	RLIN34 status interrupt	418	—	—
RLIN35					
INTRLIN35UR0	INTRLIN3nUR0 (n = 5)	RLIN35 transmit interrupt	419	group0-163	group1-71
INTRLIN35UR1	INTRLIN3nUR1 (n = 5)	RLIN35 receive completion interrupt	420	group0-164	group1-72
INTRLIN35UR2	INTRLIN3nUR2 (n = 5)	RLIN35 status interrupt	421	—	—

19.1.5 Reset Sources

RLIN3 reset sources are listed in the following table. RLIN3 is initialized by these reset sources.

Table 19.7 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
RLIN3n	All registers	✓	✓	✓	✓	—	—

19.1.6 External Input/Output Signals

External input/output signals of RLIN3 are listed below.

Table 19.8 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
RLIN30		
RLIN3nRX (n = 0)	RLIN30 receive data input	LRX0
RLIN3nTX (n = 0)	RLIN30 transmit data output	LTX0
RLIN31		
RLIN3nRX (n = 1)	RLIN31 receive data input	LRX1
RLIN3nTX (n = 1)	RLIN31 transmit data output	LTX1
RLIN32		
RLIN3nRX (n = 2)	RLIN32 receive data input	LRX2
RLIN3nTX (n = 2)	RLIN32 transmit data output	LTX2
RLIN33		
RLIN3nRX (n = 3)	RLIN33 receive data input	LRX3
RLIN3nTX (n = 3)	RLIN33 transmit data output	LTX3
RLIN34		
RLIN3nRX (n = 4)	RLIN34 receive data input	LRX4
RLIN3nTX (n = 4)	RLIN34 transmit data output	LTX4
RLIN35		
RLIN3nRX (n = 5)	RLIN35 receive data input	LRX5
RLIN3nTX (n = 5)	RLIN35 transmit data output	LTX5

CAUTION

This AC specification is only applied to the specific pin groups. For details, please refer to Appendix file "Limited_conditions_for_AC_specification.xlsx".

19.2 Overview

19.2.1 Functional Overview

The LIN/UART interface is a hardware LIN communication controller that supports LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAE J2602, and automatically performs frame communication and error determination.

The LIN/UART interface is provided with UART mode and can also be used as a UART.

The appropriate mode should be used for the LIN/UART interface according to the application: LIN master, LIN slave or UART.

LIN master

- LIN reset mode
- LIN mode (LIN master mode)
 - LIN wake-up mode
 - LIN operation mode
- LIN Self-Test mode

LIN slave

- LIN reset mode
- LIN mode (LIN slave mode [auto baud rate] or LIN slave mode [fixed baud rate])
 - LIN wake-up mode
 - LIN operation mode
- LIN Self-Test mode

UART

- LIN reset mode
- UART mode

Table 19.9 shows the LIN/UART interface specifications.

Table 19.9 LIN/UART Interface Specifications (1/3)

Item	Specifications	
LIN communication function	Protocol	LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAE J2602
	Variable frame structure	Master <ul style="list-style-type: none"> • Break transmission width: 13 to 28 Tbits • Break delimiter transmission width: 1 to 4 Tbits • Transmission inter-byte space width (header): 0 to 7 Tbits (space between Sync field and ID field)*¹ • Transmission response space width: 0 to 7 Tbits*¹ • Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area) • Transmission wake-up width: 1 to 16 Tbits
		Slave <ul style="list-style-type: none"> • Break reception width: 9.5 or 10.5 Tbits [for fixed baud rate] : 10 or 11 Tbits [for auto baud rate] • Transmission response space width: 0 to 7 Tbits • Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area) • Transmission wake-up width: 1 to 16 Tbits
	Checksum	<ul style="list-style-type: none"> • Automatic operation for both transmission and reception • Classic or enhanced selectable (for each frame)
	Response field data byte count	Variable from 0 to 8 bytes Multi-byte (9 or more bytes) response transmission and reception also possible
	Frame communication modes	Master <ul style="list-style-type: none"> • Mode in which header transmission and response transmission/reception are started with a single transmission start request • Mode in which header transmission and response transmission are started with separate transmission start requests(frame separate mode)
Slave <ul style="list-style-type: none"> • Mode in which header is automatically received with fixed baud rate • Mode in which header is automatically received with the baud rate set according to the sync field measurement result of the sync field and break field detected 		
Wake-up transmission and reception	LIN wake-up mode provided <ul style="list-style-type: none"> • Wake-up transmission (1 to 16 Tbits) • Wake-up reception Low-level width of input signals measured 	
LIN communication function	Status	Master <ul style="list-style-type: none"> • Successful frame/wake-up transmission • Successful header transmission • Successful frame/wake-up reception*² • Successful data 1 reception • Error detection • Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN Self-Test mode)
		Slave <ul style="list-style-type: none"> • Successful response/wake-up transmission • Successful response/wake-up reception*² • Successful header reception • Successful data 1 reception • Error detection • Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN Self-Test mode)

Table 19.9 LIN/UART Interface Specifications (2/3)

Item	Specifications	
LIN communication function	Error status	Master <ul style="list-style-type: none"> • Bit error • Checksum error • Frame timeout error/response timeout error • Physical bus error • Framing error • Response preparation error
		Slave <ul style="list-style-type: none"> • Bit error • Checksum error • Frame timeout error/response timeout error • Sync field error • ID parity error • Framing error • Response preparation error
	Baud rate selection	Baud rates conforming to the LIN specifications generated using baud rate generator
	Test mode	Self-Test mode for user evaluation
Interrupt function	Master <ul style="list-style-type: none"> • Successful header/frame/wake-up transmission • Successful frame/wake-up reception*2 • Error detection 	
	Slave <ul style="list-style-type: none"> • Successful response/wake-up transmission • Successful header/response/wake-up reception*2 • Error detection 	
UART communication function	Data buffer	<ul style="list-style-type: none"> • Transmission data buffer/transmission data buffer for wait (exclusively for transmission; data length of 1. Character length of 7, 8, and 9 bits supported) • UART buffer (exclusively for transmission; variable data length from 1 to 9. Character length of 7 and 8 bits supported) • Reception data buffer (exclusively for reception; data length of 1. Character length of 7, 8, and 9 bits supported)
	Data format	Character length: 7 or 8 bits Length of 9 bits supported by using the expansion bit. Transmission stop bit: 1 or 2 bits Parity function: odd, even, 0, or none LSB- or MSB-first transfer selectable Reverse input/output of transmission/reception data possible
	Status	<ul style="list-style-type: none"> • Transmission status • Reception status • Successful UART buffer transmission • Error detection • Expansion bit detection • ID match • Reset mode status
	Error status	<ul style="list-style-type: none"> • Bit error • Framing error • Parity error • Overrun error
	Baud rate selection	With the built-in baud rate generator, any baud rate can be set.

Table 19.9 LIN/UART Interface Specifications (3/3)

Item	Specifications
UART communication function	When a certain expansion bit is at the expected level, the data received can be compared to the 8-bit data preset in the register. Reception of the stop bit is guaranteed (start of transmission can be delayed when start of transmission is attempted during reception of the stop bit).
Interrupt function	<ul style="list-style-type: none"> • Transmission start/complete • Reception complete • Status/error detection

Note 1. Since the same register is used for setting, the inter-byte space (header) = response space.

Note 2. For wake-up reception, the input signal low-level width count is indicated.

19.2.2 Block Diagram

Figure 19.1 shows a block diagram of the LIN/UART interface.

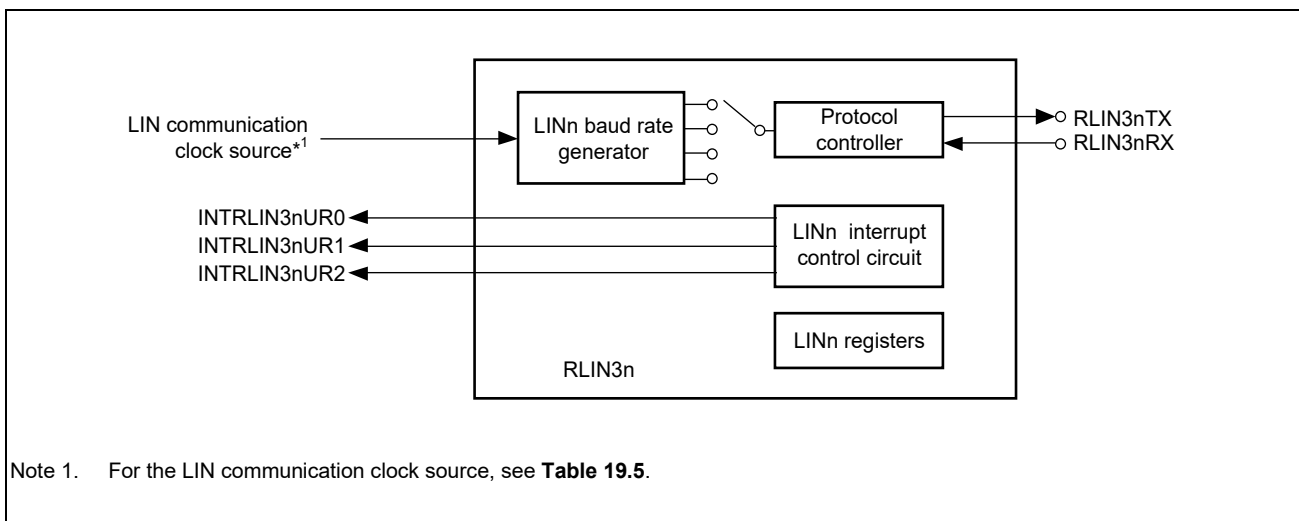


Figure 19.1 LIN/UART Interface Block Diagram

19.2.3 Terms Used in Block Diagram

- RLIN3nTX, RLIN3nRX: LIN/UART interface I/O pins
- LINn baud rate generator: Generates the LIN/UART interface communication clock.
- LINn registers: LIN/UART interface registers
- LINn interrupt control circuit: Controls interrupt requests generated by the LIN/UART interface

19.3 Registers

19.3.1 List of Registers

RLIN3 registers are listed in the following table.

For details about <RLIN3n_base>, see **Section 19.1.2, Register Base Addresses**.

Table 19.10 List of Registers

Module Name	Register Name	Symbol	Address	LIN Master	LIN Slave	UART	Access	Access Protection
RLN3n	LIN / UART Wake-Up Baud Rate Select Register	RLN3nLWBR	<RLIN3n_base> + 01 _H	✓	✓	✓	8	—
RLN3n	LIN / UART Baud Rate Prescaler 01 Register	RLN3nLBRP01	<RLIN3n_base> + 02 _H	—	✓	✓	16	—
RLN3n	LIN / UART Baud Rate Prescaler 0 Register	RLN3nLBRP0	<RLIN3n_base> + 02 _H	✓	✓	✓	8	—
RLN3n	LIN / UART Baud Rate Prescaler 1 Register	RLN3nLBRP1	<RLIN3n_base> + 03 _H	✓	✓	✓	8	—
RLN3n	LIN Self-Test Control Register	RLN3nLSTC	<RLIN3n_base> + 04 _H	✓	✓	—	8	—
RLN3n	LIN / UART Mode Register	RLN3nLMD	<RLIN3n_base> + 08 _H	✓	✓	✓	8	—
RLN3n	LIN Break Field Configuration Register/ UART Configuration Register	RLN3nLBFC	<RLIN3n_base> + 09 _H	✓	✓	✓	8	—
RLN3n	LIN / UART Space Configuration Register	RLN3nLSC	<RLIN3n_base> + 0A _H	✓	✓	✓	8	—
RLN3n	LIN Wake-Up Configuration Register	RLN3nLWUP	<RLIN3n_base> + 0B _H	✓	✓	—	8	—
RLN3n	LIN Interrupt Enable Register	RLN3nLIE	<RLIN3n_base> + 0C _H	✓	✓	—	8	—
RLN3n	LIN / UART Error Detection Enable Register	RLN3nLEDE	<RLIN3n_base> + 0D _H	✓	✓	✓	8	—
RLN3n	LIN / UART Control Register	RLN3nLCUC	<RLIN3n_base> + 0E _H	✓	✓	✓	8	—
RLN3n	LIN / UART Transmission Control Register	RLN3nLTRC	<RLIN3n_base> + 10 _H	✓	✓	✓	8	—
RLN3n	LIN / UART Mode Status Register	RLN3nLMST	<RLIN3n_base> + 11 _H	✓	✓	✓	8	—
RLN3n	LIN / UART Status Register	RLN3nLST	<RLIN3n_base> + 12 _H	✓	✓	✓	8	—
RLN3n	LIN / UART Error Status Register	RLN3nLEST	<RLIN3n_base> + 13 _H	✓	✓	✓	8	—
RLN3n	LIN / UART Data Field Configuration Register	RLN3nLDFC	<RLIN3n_base> + 14 _H	✓	✓	✓	8	—
RLN3n	LIN / UART ID Buffer Register	RLN3nLIDB	<RLIN3n_base> + 15 _H	✓	✓	✓	8	—
RLN3n	LIN Checksum Buffer Register	RLN3nLCBR	<RLIN3n_base> + 16 _H	✓	✓	—	8	—
RLN3n	UART Data Buffer 0 Register	RLN3nLUDB0	<RLIN3n_base> + 17 _H	—	—	✓	8	—
RLN3n	LIN / UART Data Buffer 1 Register	RLN3nLDBR1	<RLIN3n_base> + 18 _H	✓	✓	✓	8	—
RLN3n	LIN / UART Data Buffer 2 Register	RLN3nLDBR2	<RLIN3n_base> + 19 _H	✓	✓	✓	8	—
RLN3n	LIN / UART Data Buffer 3 Register	RLN3nLDBR3	<RLIN3n_base> + 1A _H	✓	✓	✓	8	—
RLN3n	LIN / UART Data Buffer 4 Register	RLN3nLDBR4	<RLIN3n_base> + 1B _H	✓	✓	✓	8	—
RLN3n	LIN / UART Data Buffer 5 Register	RLN3nLDBR5	<RLIN3n_base> + 1C _H	✓	✓	✓	8	—
RLN3n	LIN / UART Data Buffer 6 Register	RLN3nLDBR6	<RLIN3n_base> + 1D _H	✓	✓	✓	8	—
RLN3n	LIN / UART Data Buffer 7 Register	RLN3nLDBR7	<RLIN3n_base> + 1E _H	✓	✓	✓	8	—
RLN3n	LIN / UART Data Buffer 8 Register	RLN3nLDBR8	<RLIN3n_base> + 1F _H	✓	✓	✓	8	—
RLN3n	UART Operation Enable Register	RLN3nLUOER	<RLIN3n_base> + 20 _H	—	—	✓	8	—
RLN3n	UART Option Register 1	RLN3nLUOR1	<RLIN3n_base> + 21 _H	—	—	✓	8	—
RLN3n	UART Transmission Data Register	RLN3nLUTDR	<RLIN3n_base> + 24 _H	—	—	✓	16	—
RLN3n	UART Transmission Data Register L	RLN3nLUTDRL	<RLIN3n_base> + 24 _H	—	—	✓	8	—
RLN3n	UART Transmission Data Register H	RLN3nLUTDRH	<RLIN3n_base> + 25 _H	—	—	✓	8	—
RLN3n	UART Reception Data Register	RLN3nLURDR	<RLIN3n_base> + 26 _H	—	—	✓	16	—
RLN3n	UART Reception Data Register L	RLN3nLURDRL	<RLIN3n_base> + 26 _H	—	—	✓	8	—
RLN3n	UART Reception Data Register H	RLN3nLURDRH	<RLIN3n_base> + 27 _H	—	—	✓	8	—
RLN3n	UART Wait Transmission Data Register	RLN3nLUWTDRL	<RLIN3n_base> + 28 _H	—	—	✓	16	—
RLN3n	UART Wait Transmission Data Register L	RLN3nLUWTDRL	<RLIN3n_base> + 28 _H	—	—	✓	8	—
RLN3n	UART Wait Transmission Data Register H	RLN3nLUWTDRL	<RLIN3n_base> + 29 _H	—	—	✓	8	—

Note: ✓: Used, —: Not used

When writing to an unused register, write the value after reset.

19.3.2 LIN Master Related Registers

19.3.2.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NSPB[3:0]				LPRS[2:0]			LWBR0
Value after reset	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.11 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 samplings Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128
0	LWBR0	Wake-up Baud Rate Select 0: In LIN wake-up mode, the clock specified in the LCKS bit of the RLN3nLMD register is used. (LIN1.3) 1: In LIN wake-up mode, the clock fa is used regardless of the setting in the LCKS bit of the RLN3nLMD register. (LIN2.x)

Configure the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

These bits select the number of sampling in one Tbit (reciprocal of the baud rate).

In LIN master mode (LIN/UART mode select bits in LIN mode register = 00_B), set these bits to 0000_B (16 sampling).

LPRS[2:0] Bits (Prescaler Clock Select)

These bits select the frequency division ratio for the prescaler.

The LIN communication clock source is divided by this prescaler.

LWBR0 Bit (Wake-up Baud Rate Select)

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. This allows an input signal with a low-level width of 2.5 Tbits or more to be measured.

When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 selects fa as the LIN system clock (fLIN) during LIN wake-up mode regardless of the setting of the RLN3nLMD.LCKS bit (the LCKS bit is not changed). This allows an input signal with a low-level width of 2.5 Tbits or more to be measured.

Setting the baud rate to 19200 bps while fa is selected allows an input signal with a low-level width of 130 us or longer to be detected in the LIN wake-up mode regardless of the setting of the RLN3nLMD.LCKS bit.

19.3.2.2 RLN3nLBRP0 — LIN Baud Rate Prescaler 0 Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LBRP0[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.12 RLN3nLBRP0 Register Contents

Bit Position	Bit Name	Function
7 to 0	LBRP0[7:0]	Assuming that the value set in this register is N (0 to 255), the baud rate prescaler divides the frequency of the prescaler clock by N + 1. Setting range: 00 _H to FF _H

Configure the RLN3nLBRP0 register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock sources fa, fb, and fc.

Assuming that the value set in this register is N, baud rate prescaler 0 divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) by N + 1.

Registers RLN3nLBRP0 and RLN3nLBRP1 can be accessed as RLN3nLBRP01 in 16-bit units.

19.3.2.3 RLN3nLBRP1 — LIN Baud Rate Prescaler 1 Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LBRP1[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.13 RLN3nLBRP1 Register Contents

Bit Position	Bit Name	Function
7 to 0	LBRP1[7:0]	Assuming that the value set in this register is M (0 to 255), the baud rate prescaler divides the frequency of the prescaler clock by M + 1. Setting range: 00 _H to FF _H

Set the RLN3nLBRP1 register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock source f_d .

Assuming that the value set in this register is M, baud rate prescaler 1 divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) by M+1.

Registers RLN3nLBRP0 and RLN3nLBRP1 can be accessed as RLN3nLBRP01 in 16-bit units.

19.3.2.4 RLN3nLSTC — LIN Self-Test Control Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LSTM
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.14 RLN3nLSTC Register Contents

Bit Position	Bit Name	Function
7 to 0	—	LIN Self-Test Mode Entry In writing to this bit, write value is meaningful only when A7H, 58H, and 01H are written successively to this register in LIN Self-Test mode. In other cases, write value is ignored.
0	LSTM	LIN Self-Test Mode 0: The module is not in LIN Self-Test mode. 1: The module is in LIN Self-Test mode.

The RLN3nLSTC register cancels protection of LIN Self-Test mode.

Set the RLN3nLSTC register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

Writing A7H, 58H, and 01H successively to the RLN3nLSTC register places the module into LIN Self-Test mode.

When successive writing is completed and the mode is changed to LIN self-test mode, the LSTM bit is set to 1.

Do not write any other value during successive writing. For making transition to LIN Self-Test mode, see **Section 19.5.5, LIN Self-Test Mode**.

When read, bits 6 to 1 return “000 000_B”, and bit 7 returns an undefined value.

LSTM Bit (LIN Self-Test Mode)

When transition to LIN Self-Test mode is completed, the LSTM bit is set to 1.

For exiting LIN Self-Test mode, see **Section 19.5.5, LIN Self-Test Mode**.

Writing 1 to this bit does not affect the value of the RLN3nLSTC register if it is not a part of successive writing of A7H, 58H, and 01H.

19.3.2.5 RLN3nLMD — LIN Mode Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	LIOS	LCKS[1:0]		LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.15 RLN3nLMD Register Contents

Bit Position	Bit Name	Function															
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.															
5	LRDNFS	LIN Reception Data Noise Filter Disable 0: The noise filter is enabled. 1: The noise filter is disabled.															
4	LIOS	LIN Interrupt Output Select 0: Not supported, setting is prohibited. 1: RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are used.															
3, 2	LCKS[1:0]	LIN System Clock Select <table border="0"> <tr> <td>b3</td> <td>b2</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>fa (Clock generated by baud rate prescaler 0)</td> </tr> <tr> <td>0</td> <td>1</td> <td>fb (1/2 clock generated by baud rate prescaler 0)</td> </tr> <tr> <td>1</td> <td>0</td> <td>fc (1/8 clock generated by baud rate prescaler 0)</td> </tr> <tr> <td>1</td> <td>1</td> <td>fd (1/2 clock generated by baud rate prescaler 1)</td> </tr> </table>	b3	b2		0	0	fa (Clock generated by baud rate prescaler 0)	0	1	fb (1/2 clock generated by baud rate prescaler 0)	1	0	fc (1/8 clock generated by baud rate prescaler 0)	1	1	fd (1/2 clock generated by baud rate prescaler 1)
b3	b2																
0	0	fa (Clock generated by baud rate prescaler 0)															
0	1	fb (1/2 clock generated by baud rate prescaler 0)															
1	0	fc (1/8 clock generated by baud rate prescaler 0)															
1	1	fd (1/2 clock generated by baud rate prescaler 1)															
1, 0	LMD[1:0]	LIN/UART Mode Select <table border="0"> <tr> <td>b1</td> <td>b0</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>LIN master mode</td> </tr> </table>	b1	b0		0	0	LIN master mode									
b1	b0																
0	0	LIN master mode															

Set the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode)

LRDNFS Bit (LIN Reception Data Noise Filtering Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.

With 0 set, the noise filter is enabled when receiving data.

With 1 set, the noise filter is disabled when receiving data.

LIOS Bit (LIN Interrupt Output Select)

The LIOS bit selects the number of interrupt outputs from the LIN/UART interface.

With 0 set, Not supported, setting is prohibited.

With 1 set, the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are generated from the LIN/UART interface.

For each interrupt source, see **Section 19.4, Interrupt Sources**.

The LIOS bit must always be set when using LIN mode operation, in order to enable the interrupt generation by the RLIN3 module. For UART mode, this bit is not relevant.

LCKS[1:0] Bits (LIN System Clock Select)

The LCKS bits select the clock to be input to the protocol controller.

With 00_B set, the protocol controller is provided with fa (clock generated by baud rate prescaler 0).

With 01_B set, the protocol controller is provided with fb (1/2 clock generated by baud rate prescaler 0).

With 10_B set, the protocol controller is provided with fc (1/8 clock generated by baud rate prescaler 0).

With 11_B set, the protocol controller is provided with fd (1/2 clock generated by baud rate prescaler 1).

When 1_B is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x), and the RLN3nLMST register is 01_H (in LIN wake-up mode), the protocol controller is supplied with fa regardless of the setting of this bit (the LCKS bit is not changed).

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.

To use the LIN/UART interface as an LIN master, set these bits to 00_B.

19.3.2.6 RLN3nLBFC — LIN Break Field Configuration Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	BDT[1:0]		BLT[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.16 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function																					
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.																					
5, 4	BDT[1:0]	Transmission Break Delimiter (High Level) Width Select <table border="0"> <tr> <td>b5</td> <td>b4</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>1 Tbit</td> </tr> <tr> <td>0</td> <td>1</td> <td>2 Tbits</td> </tr> <tr> <td>1</td> <td>0</td> <td>3 Tbits</td> </tr> <tr> <td>1</td> <td>1</td> <td>4 Tbits</td> </tr> </table>	b5	b4		0	0	1 Tbit	0	1	2 Tbits	1	0	3 Tbits	1	1	4 Tbits						
b5	b4																						
0	0	1 Tbit																					
0	1	2 Tbits																					
1	0	3 Tbits																					
1	1	4 Tbits																					
3 to 0	BLT[3:0]	Transmission Break (Low Level) Width Select <table border="0"> <tr> <td>b3</td> <td>b0</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0: 13 Tbits</td> </tr> <tr> <td>0</td> <td>0</td> <td>1: 14 Tbits</td> </tr> <tr> <td>0</td> <td>0</td> <td>1: 15 Tbits</td> </tr> <tr> <td></td> <td></td> <td>:</td> </tr> <tr> <td>1</td> <td>1</td> <td>1: 27 Tbits</td> </tr> <tr> <td>1</td> <td>1</td> <td>1: 28 Tbits</td> </tr> </table>	b3	b0		0	0	0: 13 Tbits	0	0	1: 14 Tbits	0	0	1: 15 Tbits			:	1	1	1: 27 Tbits	1	1	1: 28 Tbits
b3	b0																						
0	0	0: 13 Tbits																					
0	0	1: 14 Tbits																					
0	0	1: 15 Tbits																					
		:																					
1	1	1: 27 Tbits																					
1	1	1: 28 Tbits																					

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

Some combinations of the set values result in the length of a frame exceeding the timeout time. Set the appropriate values in this register.

BDT[1:0] Bits (Transmission Break Delimiter (High Level) Width Select)

The BDT bits set the break delimiter (high level) width of transmission frame header. 1 Tbit to 4 Tbits can be set.

BLT[3:0] Bits (Transmission Break (Low Level) Width Select)

The BLT bits set the break (low level) width of transmission frame header. 13 Tbits to 28 Tbits can be set.

19.3.2.7 RLN3nLSC — LIN Space Configuration Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]		—	IBHS[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 19.17 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	IBHS[2:0]	Inter-Byte Space (Header)/Response Space Select b2 b0 0 0 0: 0 Tbit 0 0 1: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits

Set the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

Some combinations of the set values result in the length of a frame or a response exceeding the timeout time. Set the appropriate values in this register.

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the inter-byte space of the transmission frame response field.

0 Tbit to 3 Tbits can be set.

These bits are enabled only during response transmission; these are disabled during response reception.

IBHS[2:0] Bits (Inter-Byte Space (Header)/Response Space Select)

The IBHS bits set the width of the response space before response transmission.

0 Tbit to 7 Tbits can be set.

The response space setting is enabled only during response transmission; setting is disabled during response reception.

The inter-byte space (header) is equal to the response space.

19.3.2.8 RLN3nLWUP — LIN Wake-Up Configuration Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	WUTL[3:0]				—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 19.18 RLN3nLWUP Register Contents

Bit Position	Bit Name	Function
7 to 4	WUTL[3:0]	Wake-up Transmission Low Level Width Select b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits
3 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN3nLWUP register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

WUTL[3:0] Bits (Wake-up Transmission Low Level Width Select)

The WUTL bits set the low level width of the wake-up signal transmission.

1 Tbit to 16 Tbits can be set.

With 1 is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x), fa is selected as the LIN system clock (fLIN) regardless of the setting of the RLN3nLMD.LCKS bit (the LCKS bit is not changed).

19.3.2.9 RLN3nLIE — LIN Interrupt Enable Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SHIE	ERRIE	FRCIE	FTCIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 19.19 RLN3nLIE Register Contents

Bit Position	Bit Name	Function
7 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	SHIE	Successful Header Transmission Interrupt Enable 0: Disables successful header transmission interrupt request. 1: Enables successful header transmission interrupt request.
2	ERRIE	Error Detection Interrupt Enable 0: Disables error detection interrupt request. 1: Enables error detection interrupt request.
1	FRCIE	Successful Frame/Wake-up Reception Interrupt Enable 0: Disables successful frame/wake-up reception interrupt request. 1: Enables successful frame/wake-up reception interrupt request.
0	FTCIE	Successful Frame/Wake-up Transmission Interrupt Enable 0: Disables successful frame/wake-up transmission interrupt request. 1: Enables successful frame/wake-up transmission interrupt request.

Set the RLN3nLIE register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

SHIE Bit (Successful Header Transmission Interrupt Enable)

The SHIE bit enables or disables interrupt requests upon successful transmission of a header.

With 0 set, the interrupt request for RLIN3n transmission is not generated when the HTRC flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for RLIN3n transmission is generated when the HTRC flag in the RLN3nLST register is set to 1.

ERRIE Bit (Error Detection Interrupt Enable)

The ERRIE bit enables or disables an interrupt request upon detection of an error.

With 0 set, the interrupt request for RLIN3n status is not generated when the ERR flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for RLIN3n status is generated when the ERR flag in the RLN3nLST register is set to 1.

Occurrence factors are bit errors, physical bus errors, frame/response timeout errors, framing errors, checksum errors, and response preparation errors.

Detection of the bit error, physical bus error, frame/response timeout error, and framing error can be enabled or disabled using the RLN3nLEDE register.

FRCIE Bit (Successful Frame/Wake-up Reception Interrupt Enable)

The FRCIE bit enables or disables an interrupt request upon successful reception of a frame or a wake-up signal (input signal low-level width count).

With 0 set, the interrupt request for successful RLIN3n reception is not generated when the FRC flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for successful RLIN3n reception is generated when the FRC flag in the RLN3nLST register is set to 1.

FTCIE Bit (Successful Frame/Wake-up Transmission Interrupt Enable)

The FTCIE bit enables or disables an interrupt request upon successful transmission of a frame or a wake-up signal.

With 0 set, the interrupt request for RLIN3n transmission is not generated when the FTC flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for RLIN3n transmission is generated when the FTC flag in the RLN3nLST register is set to 1.

19.3.2.10 RLN3nLEDE —LIN Error Detection Enable Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LTES	—	—	—	FERE	FTERE	PBERE	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

Table 19.20 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7	LTES	Timeout Error Select 0: Frame timeout error 1: Response timeout error
6 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	FERE	Framing Error Detection Enable 0: Disables framing error detection. 1: Enables framing error detection.
2	FTERE	Timeout Error Detection Enable 0: Disables frame/response timeout error detection. 1: Enables frame/response timeout error detection.
1	PBERE	Physical Bus Error Detection Enable 0: Disables physical bus error detection. 1: Enables physical bus error detection.
0	BERE	Bit Error Detection Enable 0: Disables bit error detection. 1: Enables bit error detection.

Set the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode)

LTES Bit (Timeout Error Select)

The LTES bit selects the timeout function to be used.

With 0 set, the timeout function is used for frame timeout.

With 1 set, the timeout function is used for response timeout.

For details on the timeout error, see **Section 19.5.3.7, Error Status**.

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.

With 0 set, the framing error is not detected.

With 1 set, the framing error is detected.

When this bit is set to 1, the detection result is indicated in the FER flag in the RLN3nLEST register.

For details on the framing error, see **Section 19.5.3.7, Error Status**.

FTERE Bit (Timeout Error Detection Enable)

The FTERE bit enables or disables detection of the frame timeout error or the response timeout error.

With 0 set, the frame timeout error or response timeout error is not detected.

With 1 set, the frame timeout error or response timeout error is detected.

When this bit is set to 1, the detection result is indicated in the FTER flag in the RLN3nLEST register.
With the LTES bit, either the frame timeout error or response timeout error can be selected.

Do not use the timeout error if response data of 9 bytes or more is to be transmitted or received.
For details on the timeout error, see **Section 19.5.3.7, Error Status**.

Timeout error should be disabled for data group communication.

PBERE Bit (Physical Bus Error Detection Enable)

The PBERE bit enables or disables detection of the physical bus error.

With 0 set, the physical bus error is not detected.

With 1 set, the physical bus error is detected.

When this bit is set to 1, the detection result is indicated in the PBER flag in the RLN3nLEST register.

For details on the physical bus error, see **Section 19.5.3.7, Error Status**.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.

With 0 set, the bit error is not detected.

With 1 set, the bit error is detected.

When this bit is set to 1, the detection result is indicated in the BER flag in the RLN3nLEST register.

For details on the bit error, see **Section 19.5.3.7, Error Status**.

19.3.2.11 RLN3nLCUC — LIN Control Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OM1	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 19.21 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode. 1: LIN operation mode.
0	OM0	LIN Reset 0: LIN reset mode. 1: LIN reset mode is canceled.

Set the RLN3nLCUC register to 01_H to transition to LIN wake-up or to 03_H to transition to LIN operation mode after exiting LIN reset mode.

In LIN Self-Test mode, set the RLN3nLCUC register to 03_H after a transition to LIN Self-Test mode is completed.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

OM1 Bit (LIN Mode Select)

The OM1 bit selects the specific operation mode (either LIN wake-up mode or LIN operation mode) after canceling LIN reset mode.

With 0 set, LIN wake-up mode.

With 1 set, LIN operation mode.

This bit is valid only when the OMM0 bit in the RLN3nLMST register is 1.

Writing a value to this bit is disabled while the FTS bit in the RLN3nLTRC register is 1.

OM0 Bit (LIN Reset)

The OM0 bit selects either causing a transition to LIN reset mode or canceling LIN reset mode.

With 0 set, LIN reset mode is caused.

With 1 set, LIN reset mode is canceled.

19.3.2.12 RLN3nLTRC — LIN Transmission Control Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTS	FTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 19.22 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	RTS	Response Transmission/Reception Start 0: Response transmission/reception is stopped in frame separate mode. 1: Response transmission/reception is started in frame separate mode.
0	FTS	Frame Transmission/Wake-up Transmission /Reception Start 0: Frame Transmission/wake-up transmission/reception is stopped. 1: Frame Transmission/wake-up transmission reception is started.

RTS Bit (Response Transmission/Reception Start)

Set the RTS bit to 1 in frame separate mode after header transmission is started (FTS bit is 1) and response transmission data is ready. Once set, this bit is automatically cleared to 0 upon completion of frame communication (including error detection) or transition to LIN reset mode.

Only 1 can be written to this bit; 0 cannot be written.

To write 1 to this bit, write 02_H to the RLN3nLTRC register using the store instruction.

Writing a value to this bit is disabled when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

Writing a value to this bit is disabled when the FTS bit is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, set this bit to 1 each time a data group (variable from 0 to 8 bytes) is transmitted or received. Once set, this bit is automatically cleared to 0 at the end of data group communication or transition to LIN reset mode.

FTS Bit (Frame Transmission/Wake-up Transmission/Reception Start)

Set the FTS bit to 1 to start frame transmission and reception.

Also set this bit to 1 to allow wake-up transmission and reception (input signal low-level width count).

Only 1 can be written to this bit; 0 cannot be written.

Writing a value to this bit is disabled when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

This bit is set to 0 upon completion of frame or wake-up communication (including error detection) and transition to LIN reset mode.

19.3.2.13 RLN3nLMST — LIN Mode Status Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OMM1	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 19.23 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. Writing is ignored.
1	OMM1	LIN Mode Status Monitor 0: LIN wake-up mode. 1: LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

OMM1 Bit (LIN Mode Status Monitor)

The OMM1 bit indicates the current operating mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

It takes 3 peripheral clock cycles + 4 communication clock cycles, until the OMM0 bit in the RLN3nLMST register reflects the OMM0 bit setting in the RLN3nLCUC register.

19.3.2.14 RLN3nLST — LIN Status Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	HTRC	D1RC	—	—	ERR	—	FRC	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

Table 19.24 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7	HTRC	Successful Header Transmission Flag 0: Header transmission has not been completed. 1: Header transmission has been completed.
6	D1RC	Successful Data 1 Reception Flag 0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.
5, 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	FRC	Successful Frame/Wake-up Reception Flag 0: Frame or wake-up reception has not been completed. 1: Frame or wake-up reception has been completed.
0	FTC	Successful Frame/Wake-up Transmission Flag 0: Frame or wake-up transmission has not been completed. 1: Frame or wake-up transmission has been completed.

The RLN3nLST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

HTRC Flag (Successful Header Transmission Flag)

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The HTRC flag is set to 1 upon completion of header transmission. Here, an interrupt is generated if the SHIE bit in the RLN3nLIE register is 1 (interrupt is enabled). This bit is cleared automatically when LTRC.FTS is set.

D1RC Flag (Successful Data 1 Reception Flag)

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The D1RC flag is set to 1 upon completion of data 1 reception. Here, an interrupt request is not generated.

To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

When response data of 9 bytes or more is to be received, this bit is set to 1 each time data 1 of a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

ERR Flag (Error Detection Flag)

The ERR flag is set to 1 upon detection of an error (when at least one of the flags of the RLN3nLEST register is set to 1). Here, an interrupt request for RLN3n status is generated if the ERRIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the RPER, CSER, FER, FTER, PBER, and BER flags in the RLN3nLEST register in LIN operation mode or LIN wake-up mode. This clears the ERR flag to 0.

FRC Flag (Successful Frame/Wake-up Reception Flag)

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The FRC flag is set to 1 upon completion of frame or wake-up reception. Here, an interrupt request for RLN3n reception complete is generated if the FRCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

When response data of 9 bytes or more is to be received, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

FTC Flag (Successful Frame/Wake-up Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The FTC flag is set to 1 upon completion of frame or wake-up transmission. Here, an interrupt request for RLN3n transmission is generated if the FTCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

When response data of 9 bytes or more is to be transmitted, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is transmitted. Write 0 before starting transmission of the next data group.

19.3.2.15 RLN3nLEST — LIN Error Status Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RPER	—	CSER	—	FER	FTER	PBER	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Table 19.25 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	RPER	Response Preparation Error Flag 0: Response preparation error has not been detected. 1: Response preparation error has been detected.
6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	CSER	Checksum Error Flag 0: Checksum error has not been detected. 1: Checksum error has been detected.
4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	FTER	Timeout Error Flag 0: Frame/response timeout error has not been detected. 1: Frame/response timeout error has been detected.
1	PBER	Physical Bus Error Flag 0: Physical bus error has not been detected. 1: Physical bus error has been detected.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

When the FTS bit in the RLN3nLTRC register is 1 (frame transmission or wake-up transmission/reception is started), do not write a value to this register.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

RPER Flag (Response Preparation Error Flag)

Only 0 can be written to the RPER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The RPER flag is set to 1 upon response preparation error detection. This happens, if response preparation was not completed before the first byte of a response has been received.

This bit is cleared automatically when LTRC.FTS is set.

CSER Flag (Checksum Error Flag)

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The CSER flag is set to 1 upon checksum error detection.

This bit is cleared automatically when LTRC.FTS is set.

FER Flag (Framing Error Flag)

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

When the value of the FERE bit of the RLIN3nLEDE register is 1 (framing error detection enabled), the FER flag is set to 1 upon framing error detection. This bit is cleared automatically when LTRC.FTS is set.

FTER Flag (Timeout Error Flag)

Only 0 can be written to the FTER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

When the FTERE bit of the RLIN3nLEDE register is 1 (frame/response timeout error detection enabled), the FTER flag is set to 1 upon frame timeout error or response timeout error detection. This bit is cleared automatically when LTRC.FTS is set.

PBER Flag (Physical Bus Error Flag)

Only 0 can be written to the PBER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

When the PBERE bit of the RLIN3nLEDE register is 1 (physical bus error detection enabled), the PBER flag is set to 1 upon physical bus error detection. This bit is cleared automatically when LTRC.FTS is set.

BER Flag (Bit Error Flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

When the BERE bit of the RLIN3nLEDE register is 1 (bit error detection enabled), the BER flag is set to 1 upon bit error detection. This bit is cleared automatically when LTRC.FTS is set.

19.3.2.16 RLN3nLDFC — LIN Data Field Configuration Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LSS	FSM	CSM	RFT	RFDL[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.26 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function																																							
7	LSS	Transmission/Reception Continuation Select 0: The data group to be transmitted/received next is the last one. 1: The data group to be transmitted/received next is not the last one. (Checksum is not included.)																																							
6	FSM	Frame Separate Mode Select 0: Frame separate mode is not set. 1: Frame separate mode is set.																																							
5	CSM	Checksum Select 0: Classic checksum 1: Enhanced checksum																																							
4	RFT	Response Field Communication Direction Select 0: Reception 1: Transmission																																							
3 to 0	RFDL[3:0]	Response Field Length Select <table border="0"> <tr> <td>b3</td> <td>b0</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0: 0 byte (+ checksum)</td> </tr> <tr> <td>0</td> <td>0</td> <td>1: 1 byte (+ checksum)</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0: 2 bytes (+ checksum)</td> </tr> <tr> <td colspan="3" style="text-align: center;">:</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1: 7 bytes (+ checksum)</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0: 8 bytes (+ checksum)</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1: 8 bytes (+ checksum)</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0: 8 bytes (+ checksum)</td> </tr> <tr> <td colspan="3" style="text-align: center;">:</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1: 8 bytes (+ checksum)</td> </tr> </table>	b3	b0		0	0	0: 0 byte (+ checksum)	0	0	1: 1 byte (+ checksum)	0	0	1	0: 2 bytes (+ checksum)	:			0	1	1	1: 7 bytes (+ checksum)	1	0	0	0: 8 bytes (+ checksum)	1	0	0	1: 8 bytes (+ checksum)	1	0	1	0: 8 bytes (+ checksum)	:			1	1	1	1: 8 bytes (+ checksum)
b3	b0																																								
0	0	0: 0 byte (+ checksum)																																							
0	0	1: 1 byte (+ checksum)																																							
0	0	1	0: 2 bytes (+ checksum)																																						
:																																									
0	1	1	1: 7 bytes (+ checksum)																																						
1	0	0	0: 8 bytes (+ checksum)																																						
1	0	0	1: 8 bytes (+ checksum)																																						
1	0	1	0: 8 bytes (+ checksum)																																						
:																																									
1	1	1	1: 8 bytes (+ checksum)																																						

LSS Bit (Transmission/Reception Continuation Select)

The LSS bit indicates that the data group to be transmitted or received next is not the last data group when response data of 9 bytes or more is to be transmitted or received.

With 0 set, data and checksum are transmitted or received because the next data group to be transmitted or received is the last one.

With 1 set, only data is transmitted or received, and the checksum is not included because the next data group to be transmitted or received is not the last one.

Set the LSS bit only when the FSM bit is 1 (frame separate mode) and response data of 9 bytes or more is to be transmitted or received.

Set the LSS bit only when the RTS bit in the RLN3nLTRC is 0 (response transmit/receive is stopped).

FSM Bit (Frame Separate Mode Select)

The FSM bit sets the response communication mode.

With 0 set, frame separate mode is not selected. In this case, after header transmission is started (the FTS bit in the

RLN3nLTRC register is 1), response is transmitted/received without setting the RTS bit in the RLN3nLTRC register. With 1 set, frame separate mode is selected. If the RTS bit of the RLN3nLTRC register is set to 1 during header transmission, response transmission is executed after header transmission is completed.

For response reception which is 8 bytes or less (the RFT bit is 0), set the FSM bit to 0.

When causing a transition to LIN Self-Test mode, set this bit to 0 before transition.

For details on frame separate mode, see **Section 19.5.3.4(1), Transmission of LIN Frames**.

Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, set the FSM bit to 1.

CSM Bit (Checksum Select)

The CSM bit sets the checksum mode.

With 0 set, classic checksum mode is selected.

With 1 set, enhanced checksum mode is selected.

When the timeout error is used (the FTERE bit in the RLN3nLEDE register is 1), the specific timeout time depends on the setting of this bit. For details on the bit error, see **Section 19.5.3.7, Error Status**.

Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, do not change the CSM bit setting after the first data group through the last data group.

During communication of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

RFT Bit (Response Field Communication Direction Select)

The RFT bit sets the direction of the response field/wake-up signal communication.

With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (input signal low-level width count).

With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed. Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, do not change the RFT bit setting after the first data group through the last data group.

RFDL[3:0] Bits (Response Field Length Select)

The RFDL bits set the length of the response field data.

The data length can be 0 to 8 bytes excluding the checksum size.

To transmit response data with the FSM bit set to 0 (not frame separate mode), set the RFDL bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).

To transmit response data with the FSM bit set to 1 (frame separate mode), set the RFDL bits before response transmission (the RTS bit in the RLN3nLTRC register is 0).

To receive response data, set the RFDL bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).

When response data of 9 bytes or more is to be transmitted or received, set the RFDL bits before data group transmission/reception (RTS bit in the RLN3nLTRC register is 0).

Only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

19.3.2.17 RLN3nLIDB — LIN ID Buffer Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	IDP1	IDP0	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.27 RLN3nLIDB Register Contents

Bit Position	Bit Name	Function
7	IDP1	Parity Setting (P1) Sets the parity bits (P1) to be transmitted in the ID field.
6	IDP0	Parity Setting (P0) Sets the parity bits (P0) to be transmitted in the ID field.
5 to 0	ID[5:0]	ID Setting Sets the 6-bit ID value to be transmitted in the ID field.

Set the RLN3nLIDB register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

In LIN Self-Test mode, this register operates as described below.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN Self-Test mode, see **Section 19.5.5, LIN Self-Test Mode**.

IDP[1:0] Bits (Parity Setting)

The IDP bits set the parity bits (P0 and P1) to be transmitted in the ID field of the LIN frame. IDP0 is for P0 and IDP1 for P1.

Since parity is not automatically calculated, set the calculation result. Note that if the erroneous result is set, it is transmitted as is.

ID[5:0] Bits (ID Setting)

The ID bits set the 6-bit ID value to be transmitted in the ID field of the LIN frame.

19.3.2.18 RLN3nLCBR — LIN Checksum Buffer Register

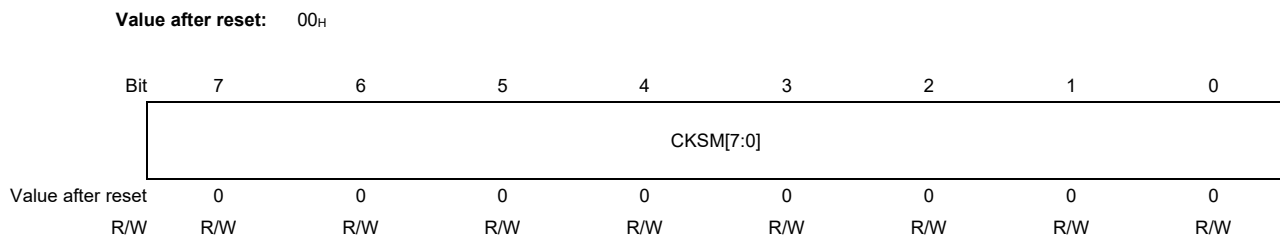


Table 19.28 RLN3nLCBR Register Contents

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Holds the checksum data transmitted or received.

In LIN mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):
The value transmitted can be read from the register. Read the value after transmission is completed.
Writing to this register is invalid.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):
The value received can be read from the register. Read the value after reception is completed.
Writing to this register is invalid.

In LIN Self-Test mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):
After completion of the frame transmission (after loopback), the reversed value of the received value can be read.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):
Write the value to be received before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN Self-Test mode, see **Section 19.5.5, LIN Self-Test Mode**.

Set the this register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, the checksum is appended only to the last data group; this register is not updated for other data groups.

19.3.2.19 RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)

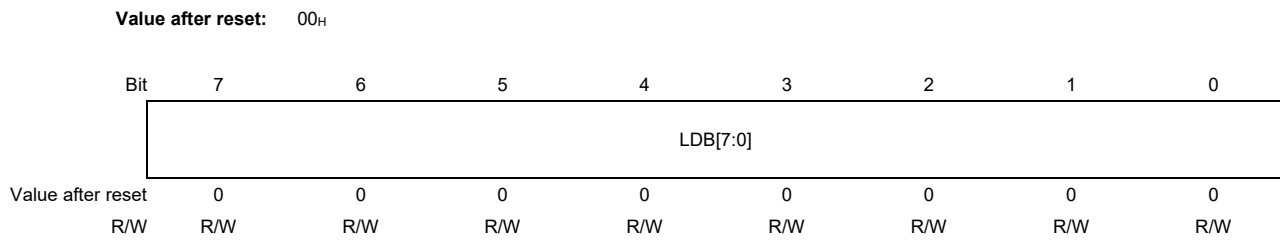


Table 19.29 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or reads the received data. Setting range: 00 _H to FF _H

- For response transmission:
 - These registers set the data to be transmitted in the response field.
 - Use these registers with the following settings.
 - RFT in RLN3nLDFC register is 1 (transmission)
 - FSM in RLN3nLDFC register is 0 (not frame separate mode)
 - FTS in RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted)
 - or
 - RFT in RLN3nLDFC register is 1 (transmission)
 - FSM in RLN3nLDFC register is 1 (frame separate mode)
 - RTS in RLN3nLTRC register is 0 (response transmission/reception is halted)
- For response reception:
 - These registers hold the data received in the response field.
 - The received data is overwritten. If an error is detected, the data up to the byte in which the error was detected are stored in the register.
 - Do not read these registers when the FTS bit is 1 (frame transmission or wake-up transmission/reception is started)
- For transmission of response data of 9 bytes or more:
 - Use these registers with the following settings.
 - RFT in RLN3nLDFC register is 1 (transmission)
 - FSM in RLN3nLDFC register is 1 (frame separate mode)
 - RTS in RLN3nLTRC register is 0 (response transmission/reception is halted)
- For reception of response data of 9 bytes or more:
 - Do not read these registers when the RTS bit is 1 (response transmission/reception is started).

In LIN Self-Test mode, these registers operate as described below.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN Self-Test mode, see **Section 19.5.5, LIN Self-Test Mode**.

19.3.3 LIN Slave Related Registers

19.3.3.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NSPB[3:0]				LPRS[2:0]			—
Value after reset	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 19.30 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 sampling 0 0 1 1: 4 sampling 0 1 1 1: 8 sampling Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

The NSPB bits select the number of sampling in one Tbit (reciprocal of the baud rate).

When the communication is performed in LIN slave mode (fixed baud rate) (LMD[1:0] bits in the RLN3nLMD register = 11_B), set these bits to “0000_B (16 sampling).

When the communication is performed in LIN slave mode (auto baud rate) (LMD[1:0] bits in the RLN3nLMD register = 10_B), set these bits to “0011_B” (4 sampling) or “0111_B” (8 sampling).

LPRS[2:0] Bits (Prescaler Clock Select)

The LPRS bits select the frequency division ratio for the prescaler. The LIN communication clock source is divided by this prescaler.

In LIN slave mode (auto baud rate) (LMD[1:0] bits in the RLN3nLMD register = 10_B), set these bits so that the prescaler clock becomes as follows according to the target baud rate.

[Target baud rate]	[Prescaler clock]
1 kbps to 20 kbps:	4 MHz* ¹
1 kbps to 2.4 kbps (excluding 2.4 kbps):	4 MHz
2.4 kbps to 20 kbps:	8 MHz to 12 MHz

Note 1. Use the clock with NSPB bits set to "0011_B" (four samplings).

19.3.3.2 RLN3nLBRP01 — LIN Baud Rate Prescaler 01 Register

Value after reset: 0000_H

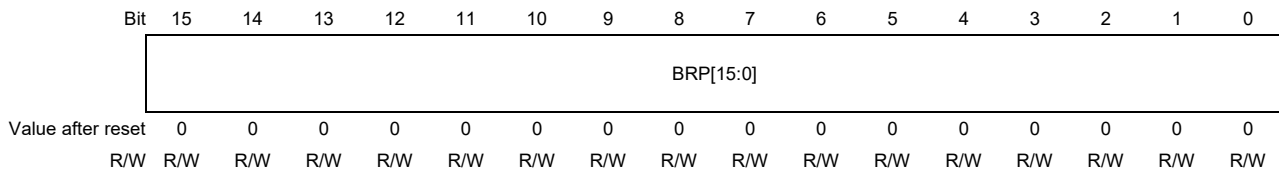


Table 19.31 RLN3nLBRP01 Register Contents

Bit Position	Bit Name	Function
15 to 0	BRP[15:0]	Assuming that the value set in this register is L (0 to 65535), the baud rate prescaler divides the frequency of the prescaler clock by L + 1. Setting range: 0000 _H to FFFF _H

Set the RLN3nLBRP01 register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

Assuming that the value set in this register is L, the baud rate prescaler divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) in the RLN3nLWBR register by L + 1.

The RLN3nLBRP01 register can be accessed in 8-bit units by registers RLN3nLBRP0 and RLN3nLBRP1.

19.3.3.3 RLN3nLSTC — LIN Self-Test Control Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LSTM
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.32 RLN3nLSTC Register Contents

Bit Position	Bit Name	Function
7 to 0	—	LIN Self-Test Mode Entry In writing to this bit, the write value is meaningful only when A7H, 58H, and 01H are written successively to this register in LIN Self-Test mode. In other cases, the write value is ignored.
0	LSTM	LIN Self-Test Mode 0: LIN Self-Test mode is not set. 1: LIN Self-Test mode is set.

The RLN3nLSTC register cancels protection of LIN Self-Test mode.

Set the RLN3nLSTC register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

Writing A7_H, 58_H, and 01_H successively to the RLN3nLSTC register places the module into LIN Self-Test mode.

When successive writing is completed thus placing LIN Self-Test mode to be entered, the LSTM bit is set to 1.

Do not write any other value during successive writing.

For making transition to LIN Self-Test mode, see **Section 19.5.5, LIN Self-Test Mode**.

When reading bits 6 to 1, they return “000000_B”, and reading bit 7, it returns an undefined value.

LSTM Bits (LIN Self-Test Mode)

When transition to LIN Self-Test mode is completed, the LSTM bit is set to 1.

For exiting LIN Self-Test mode, see **Section 19.5.5, LIN Self-Test Mode**.

Writing 1 to this bit does not affect the value of the RLN3nLSTC register if it is not a part of successive writing of A7_H, 58_H, and 01_H.

19.3.3.4 RLN3nLMD — LIN Mode Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	LIOS	—	—	LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W

Table 19.33 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	LRDNFS	LIN Reception Data Noise Filtering Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4	LIOS	LIN Interrupt Output Select 0: Not supported, setting is prohibited. 1: RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are used.
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	LMD[1:0]	LIN/UART Mode Select b1 b0 1 0: LIN slave mode with auto baud rate 1 1: LIN slave mode with fixed baud rate

Set the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LRDNFS Bit (LIN Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.

With 0 set, the noise filter is enabled when receiving data.

With 1 set, the noise filter is disabled when receiving data.

LIOS Bit (LIN Interrupt Output Select)

The LIOS bit selects the number of interrupt outputs from the LIN/UART interface.

With 0 set, not supported, setting is prohibited.

With 1 set, the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n reception status interrupt are generated from the LIN/UART interface.

For each interrupt source, see **Section 19.4, Interrupt Sources**.

The LIOS bit must always be set when using LIN mode operation, in order to enable the interrupt generation by the RLIN3n module. For UART mode, this bit is not relevant.

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.

To use this module as an LIN slave, set these bits to “10_B” (auto baud rate) or “11_B” (fixed baud rate).

19.3.3.5 RLN3nLBFC — LIN Break Field Configuration Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LBLT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 19.34 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	LBLT	Reception Break (Low-Level) Detection Width Setting 0: A break (low-level) is detected in 9.5 or 10 Tbits 1: A break (low-level) is detected in 10.5 or 11 Tbits

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LBLT Bit (Reception Break (Low-Level) Detection Width Setting)

- When RLN3nLMD.LMD is “10_B” (in LIN slave mode (auto baud rate))
 - 0: Low-level width of 10 Tbits or longer is detected.
 - 1: Low-level width of 11 Tbits or longer is detected.
- When RLN3nLMD.LMD is “11_B” (in LIN slave mode (fixed baud rate))
 - 0: Low-level width of 9.5 Tbits or longer is detected.
 - 1: Low-level width of 10.5 Tbits or longer is detected.

19.3.3.6 RLN3nLSC — LIN Space Configuration Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]		—	IBHS[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 19.35 RLN3nLSC Register Contents

Bit Position	Bit Name	Function																																				
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.																																				
5, 4	IBS[1:0]	Inter-Byte Space Select <table border="0"> <tr> <td>b5</td> <td>b4</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0 Tbit</td> </tr> <tr> <td>0</td> <td>1</td> <td>1 Tbit</td> </tr> <tr> <td>1</td> <td>0</td> <td>2 Tbits</td> </tr> <tr> <td>1</td> <td>1</td> <td>3 Tbits</td> </tr> </table>	b5	b4		0	0	0 Tbit	0	1	1 Tbit	1	0	2 Tbits	1	1	3 Tbits																					
b5	b4																																					
0	0	0 Tbit																																				
0	1	1 Tbit																																				
1	0	2 Tbits																																				
1	1	3 Tbits																																				
3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.																																				
2 to 0	IBHS[2:0]	Response Space Setting <table border="0"> <tr> <td>b2</td> <td>b1</td> <td>b0</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0 Tbit</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1 Tbit</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>2 Tbits</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>3 Tbits</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>4 Tbits</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>5 Tbits</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>6 Tbits</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>7 Tbits</td> </tr> </table>	b2	b1	b0		0	0	0	0 Tbit	0	0	1	1 Tbit	0	1	0	2 Tbits	0	1	1	3 Tbits	1	0	0	4 Tbits	1	0	1	5 Tbits	1	1	0	6 Tbits	1	1	1	7 Tbits
b2	b1	b0																																				
0	0	0	0 Tbit																																			
0	0	1	1 Tbit																																			
0	1	0	2 Tbits																																			
0	1	1	3 Tbits																																			
1	0	0	4 Tbits																																			
1	0	1	5 Tbits																																			
1	1	0	6 Tbits																																			
1	1	1	7 Tbits																																			

Set the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

This register is enabled only for response transmission, and disabled for response reception.

Some combinations of the setting values result in the length of a frame or a response exceeding the timeout time. Set the appropriate values in this register.

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the inter-byte space of the response transmission.

0 Tbit to 3 Tbits can be set.

IBHS[2:0] Bits (Inter-Byte Space (Header)/Response Space Select)

These bits set the transmission width of the response space.

0 Tbit to 7 Tbits can be set.

19.3.3.7 RLN3nLWUP — LIN Wake-Up Configuration Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	WUTL[3:0]				—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 19.36 RLN3nLWUP Register Contents

Bit Position	Bit Name	Function
7 to 4	WUTL[3:0]	Wake-up Transmission Low level Width Select b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits
3 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN3nLWUP register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

WUTL[3:0] Bits (Wake-up Transmission Low Level Width Select)

The WUTL bits set the low-level width of the wake-up frame transmission.

1 Tbit to 16 Tbits can be set.

19.3.3.8 RLN3nLIE — LIN Interrupt Enable Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SHIE	ERRIE	FRCIE	FTCIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 19.37 RLN3nLIE Register Contents

Bit Position	Bit Name	Function
7 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	SHIE	Successful Header Reception Interrupt Enable 0: Disables successful header reception interrupt request. 1: Enables successful header reception interrupt request.
2	ERRIE	Error Detection Interrupt Enable 0: Disables error detection interrupt request. 1: Enables error detection interrupt request.
1	FRCIE	Successful Response/Wake-up Reception Interrupt Enable 0: Disables successful Response/wake-up reception interrupt request. 1: Enables successful Response/wake-up reception interrupt request.
0	FTCIE	Successful Response/Wake-up Transmission Interrupt Enable 0: Disables successful Response/wake-up transmission interrupt request. 1: Enables successful Response/wake-up transmission interrupt request.

Set the RLN3nLIE register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

SHIE Bit (Successful Header Reception Interrupt Enable)

The SHIE bit enables or disables an interrupt request upon successful reception of a header.

With 0 set, the interrupt request for RLIN3n reception complete is not generated when the HTRC flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for RLIN3n reception complete is generated when the HTRC flag in the RLN3nLST register is set to 1.

ERRIE Bit (Error Detection Interrupt Enable)

The ERRIE bit enables or disables an interrupt request upon detection of an error.

With 0 set, the interrupt request for RLIN3n status is not generated when the ERR flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for RLIN3n status is generated when the ERR flag in the RLN3nLST register is set to 1.

Error types that are interrupt sources are the bit error, frame/response timeout error, framing error, sync filed error, ID parity error, checksum error, and response preparation error.

Detection of the bit error, frame/response timeout error, sync filed error, ID parity error, and framing error can be enabled or disabled using the RLN3nLEDE register.

FRCIE Bit (Successful Response/Wake-up Reception Interrupt Enable)

The FRCIE bit enables or disables an interrupt request upon successful reception of a response or a wake-up frame (input signal low-level width count).

With 0 set, the interrupt is not generated when the FTC flag in the RLIN3nLST register is set to 1.

With 1 set, the interrupt is generated when the FTC flag in the RLIN3nLST register is set to 1.

FTCIE Bit (Successful Response/Wake-up Transmission Interrupt Enable)

The FTCIE bit enables or disables an interrupt request upon successful transmission of a response or a wake-up frame.

With 0 set, the interrupt request for successful RLIN3n transmission is not generated when the FTC flag in the RLIN3nLST register is set to 1.

With 1 set, the interrupt request for successful RLIN3n transmission is generated when the FTC flag in the RLIN3nLST register is set to 1.

19.3.3.9 RLN3nLEDE — LIN Error Detection Enable Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LTES	IPERE	—	SFERE	FERE	TERE	—	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W

Table 19.38 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7	LTES	Timeout Error Select 0: Frame timeout error 1: Response timeout error
6	IPERE	ID Parity Error Detection Enable 0: Disables ID Parity error detection. 1: Enables ID Parity error detection.
5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	SFERE	Sync Field Error Detection Enable 0: Disables Sync Field error detection. 1: Enables Sync Field error detection.
3	FERE	Framing Error Detection Enable 0: Disables framing error detection. 1: Enables framing error detection.
2	TERE	Timeout Error Detection Enable 0: Disables frame/response timeout error detection. 1: Enables frame/response timeout error detection.
1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	BERE	Bit Error Detection Enable 0: Disables bit error detection. 1: Enables bit error detection.

Set the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

LTES Bit (Timeout Error Select)

The LTES bit selects the timeout function to be used.

With 0 set, the timeout function operates as frame timeout.

With 1 set, the timeout function operates as response timeout.

For details on the timeout error, see **Section 19.5.3.7, Error Status**.

IPERE Bit (ID Parity Error Detection Enable)

This bit enables or disables detection of the ID parity error.

With 0 set, the ID parity error is not detected.

With 1 set, the ID parity error is detected.

When this bit is set to 1, the detection result is reflected in the IPER flag in the RLN3nLEST register.

For details on the ID parity error, see **Section 19.5.3.7, Error Status**.

SFERE Bit (Sync Field Error Detection Enable)

This bit enables or disables detection of the sync field error.

With 0 set, the sync field error is not detected.

With 1 set, the sync field error is detected.

Regardless of the setting of this bit, when a sync field error is detected, this module waits for the next header.

When this bit is set to 1, the detection result is reflected in the SFER flag in the RLN3nLEST register.

For details on the sync field error, see **Section 19.5.3.7, Error Status**.

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.

With 0 set, the framing error is not detected.

With 1 set, the framing error is detected.

Set this bit to 1. The detection result is indicated in the FER flag in the RLN3nLEST register.

For details on the framing error, see **Section 19.5.3.7, Error Status**.

TERE Bit (Timeout Error Detection Enable)

The TERE bit enables or disables detection of the frame timeout error or the response timeout error.

With 0 set, the frame timeout error or response timeout error is not detected.

With 1 set, the frame timeout error or response timeout error is detected.

When this bit is set to 1, the detection result is reflected in the TER flag in the RLN3nLEST register.

With the LTES bit, either the frame timeout error or response timeout error can be selected.

The timeout error should not be used in LIN slave mode [auto baud rate] (when the LMD[1:0] bits in the RLN3nLMD register are "10_B").

Do not use the timeout error if response data of 9 bytes or more is to be transmitted or received.

For details on the timeout error, see **Section 19.5.3.7, Error Status**.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.

With 0 set, the bit error is not detected.

With 1 set, the bit error is detected.

When this bit is set to 1, the detection result is reflected in the BER flag in the RLN3nLEST register.

For details on the bit error, see **Section 19.5.3.7, Error Status**.

19.3.3.10 RLN3nLCUC — LIN Control Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OM1	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 19.39 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode. 1: LIN operation mode.
0	OM0	LIN Reset 0: LIN reset mode. 1: LIN reset mode is canceled.

Set the RLN3nLCUC register to 01_H to transition to LIN wake-up mode or to 03_H to transition to LIN operation mode after exiting LIN reset mode.

In LIN Self-Test mode, set the RLN3nLCUC register to 03_H after a transition to LIN Self-Test mode is completed.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

OM1 Bit (LIN Mode Select)

The OM1 bit selects the specific LIN operation mode (either LIN wake-up mode or LIN operation mode) after canceling LIN reset mode.

With 0 set, LIN wake-up mode is caused.

With 1 set, LIN operation mode is caused.

This bit is valid only when the OMM0 bit in the RLN3nLMST register is 1.

Writing a value to this bit is disabled while the FTS bit in the RLN3nLTRC register is 1.

OM0 Bit (LIN Reset)

The OM0 bit selects either causing a transition to LIN reset mode or canceling LIN reset mode.

With 0 set, LIN reset mode.

With 1 set, LIN reset mode is canceled.

It takes 3 peripheral clock cycles + 4 communication clock cycles, until the OMM0 bit in the RLN3nLMST register reflects the OM0 bit setting in the RLN3nLCUC register.

19.3.3.11 RLN3nLTRC — LIN Transmission Control Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	LNRR	RTS	FTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 19.40 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	LNRR	No LIN Response Request 0: Response for the reception ID 1: No response for the reception ID
1	RTS	Response Transmission/Reception Start 0: Response transmission/reception is stopped. 1: Response transmission/reception is started.
0	FTS	LIN Communication Start 0: Header reception/wake-up transmission/reception is stopped. 1: Header reception/wake-up transmission reception is started.

LNRR Bit (No LIN Response Request)

Set this bit to 1 if no response is to be transmitted/received after receiving the header and checking the received ID. Once set, this bit is automatically cleared to 0 upon detection of new sync field or transition to LIN reset mode.

Only 1 can be written to this bit; 0 cannot be written.

To write 1 to this bit, write 04_H using the store instruction.

Do not set this bit and the RTS bit to 1 simultaneously.

Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (in LIN reset mode).

Writing a value to this bit is disabled when the FTS bit is 0 (header reception or wake-up transmission/reception is halted).

RTS Bit (Response Transmission/Reception Start)

Set this bit to 1 to start response transmission or reception after receiving the header and checking the received ID.

Once set, this bit is automatically cleared to 0 upon completion of response transmission or reception (including error detection) or transition to LIN reset mode.

Only 1 can be written to this bit; 0 cannot be written.

To write 1 to this bit, write 02_H to the RLN3nLTRC register using the store instruction.

Do not set this bit and the LNRR bit to 1 simultaneously

Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (in LIN reset mode).

Writing a value to this bit is disabled when the FTS bit is 0 (header reception or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, set this bit to 1 each time a data group (variable from 0 to 8 bytes) is transmitted or received. Once set, this bit is automatically cleared to 0 upon completion of data group transmission/reception or transition to LIN reset mode.

FTS Bit (LIN Communication Start)

Set this bit to 1 to start header reception or wake-up transmission/reception.

Only 1 can be written to this bit; 0 cannot be written.

Writing a value to this bit is disabled when the OMM0 bit of the RLIN3nLMST register is 0_B (in LIN reset mode).

This bit is set to 0 upon completion of wake-up communication and transition to LIN reset mode.

19.3.3.12 RLN3nLMST — LIN Mode Status Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OMM1	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 19.41 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned.
1	OMM1	LIN Mode Status Monitor 0: LIN wake-up mode. 1: LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: The module is not in LIN reset mode.

OMM1 Bit (LIN Mode Status Monitor)

The OMM1 bit indicates the current operating mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

It takes 3 peripheral clock cycles + 4 communication clock cycles, until the OMM0 bit in the RLN3nLMST register reflects the OMM0 bit setting in the RLN3nLCUC register.

19.3.3.13 RLN3nLST — LIN Status Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	HTRC	D1RC	—	—	ERR	—	FRC	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

Table 19.42 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7	HTRC	Successful Header Reception Flag 0: Header reception has not been completed. 1: Header reception has been completed.
6	D1RC	Successful Data 1 Reception Flag 0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.
5, 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	FRC	Successful Response/Wake-up Reception Flag 0: Response or wake-up reception has not been completed. 1: Response or wake-up reception has been completed.
0	FTC	Successful Response/Wake-up Transmission Flag 0: Response or wake-up transmission has not been completed. 1: Response or wake-up transmission has been completed.

The RLN3nLST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, writing a value to this register is disabled. In LIN reset mode, the register retains 00_H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

HTRC Flag (Successful Header Reception Flag)

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value before 1 is written.

The HTRC flag is set to 1 upon completion of header transmission. Here, an interrupt is generated if the SHIE bit in the RLN3nLIE register is 1 (interrupt is enabled). This bit is cleared automatically when LTRC.FTS is set.

D1RC Flag (Successful Data 1 Reception Flag)

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The D1RC flag is set to 1 upon completion of data 1 reception. Here, an interrupt request is not generated. Write 0 to clear this bit.

When response data of 9 bytes or more is to be received, this bit is set to 1 each time data 1 of a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

ERR Flag (Error Detection Flag)

The ERR flag is set to 1 upon detection of an error (when at least one of the flags of the RLIN3nLEST register is set to 1). Here, an interrupt request for RLIN3n status is generated if the ERRIE bit in the RLIN3nLIE register is 1 (interrupt is enabled). However, if an error is detected while this bit is 1, an interrupt is not generated. To clear the ERR flag to 0, write 0 to the RPER, IPER, CSER, SFER, FER, TER, and BER flags in the RLIN3nLEST register. This clears the ERR flag to 0.

FRC Flag (Successful Response/Wake-up Reception Flag)

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The FRC flag is set to 1 upon completion of response or wake-up reception. Here, an interrupt request for successful RLIN3n reception is generated if the FRCIE bit in the RLIN3nLIE register is 1 (interrupt is enabled). This bit is cleared automatically when LTRC.FTS is set.

When response data of 9 bytes or more is to be received, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

FTC Flag (Successful Response/Wake-up Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The FTC flag is set to 1 upon completion of frame or wake-up transmission. Here, an interrupt request for RLIN3n transmission is generated if the FTCIE bit in the RLIN3nLIE register is 1 (interrupt is enabled). This bit is cleared automatically when LTRC.FTS is set.

When response data of 9 bytes or more is to be transmitted, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is transmitted. Write 0 before starting transmission of the next data group.

19.3.3.14 RLN3nLEST — LIN Error Status Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RPER	IPER	CSER	SFER	FER	TER	—	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Table 19.43 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	RPER	Response Preparation Error Flag 0: Response preparation error has not been detected. 1: Response preparation error has been detected.
6	IPER	ID Parity Error Flag 0: ID parity error has not been detected. 1: ID parity error has been detected.
5	CSER	Checksum Error Flag 0: Checksum error has not been detected. 1: Checksum error has been detected.
4	SFER	Sync Field Error Flag 0: Sync field error has not been detected. 1: Sync field error has been detected.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	TER	Timeout Error Flag 0: Frame/response timeout error has not been detected. 1: Frame/response timeout error has been detected.
1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode.

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

RPER Flag (Response Preparation Error Flag)

Only 0 can be written to the RPER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The RPER flag is set to 1 when response preparation is not completed before reception of the first byte of response is completed. Write 0 to clear this bit.

IPER Flag (ID Parity Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that has been retained before 1 is written. When the IPERE bit of the RLIN3nLEDE register is 1 (ID parity error detection enabled), this bit is set to 1 upon ID parity error detection. Write 0 to clear this bit.

CSER Flag (Checksum Error Flag)

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The CSER flag is set to 1 when the received checksum value during response reception does not match the internally calculated checksum value. Write 0 to clear this bit.

SFER Flag (Sync Field Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

This bit is set to 1 if the sync field is not detected as “55H” and the break low width is more than or equal to the configured break low width. Write 0 to clear this bit.

FER Flag (Framing Error Flag)

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The FER flag is set to 1 when a ‘0’ value is sampled as STOP bit during reception and the corresponding error detection is enabled. Write 0 to clear this bit.

TER Flag (Timeout Error Flag)

Only 0 can be written to the TER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

This flag is set to 1 when the internal timeout counter (frame or response timeout) reaches the error threshold value (calculated automatically) and the corresponding error detection is enabled. Write 0 to clear this bit.

BER Flag (Bit Error Flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The BER flag is set to 1 when the sampled bit value does not match the transmitted bit value during transmission and the corresponding error detection is enabled. Write 0 to clear this bit.

19.3.3.15 RLN3nLDFC — LIN Data Field Configuration Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LSS	—	LCS	RCDS	RFDL[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.44 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function																																							
7	LSS	Transmission/Reception Continuation Select 0: The data group to be transmitted/received next is the last one. 1: The data group to be transmitted/received next is not the last one. (Data transmission/reception continues without waiting for reception of the next header.)																																							
6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.																																							
5	LCS	Checksum Select 0: Classic checksum mode 1: Enhanced checksum mode																																							
4	RCDS	Response Field Communication Direction Select 0: Reception 1: Transmission																																							
3 to 0	RFDL[3:0]	Response Field Length Select <table border="0"> <tr> <td>b3</td> <td>b0</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0: 0 byte (+ checksum)</td> </tr> <tr> <td>0</td> <td>0</td> <td>1: 1 byte (+ checksum)</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0: 2 bytes (+ checksum)</td> </tr> <tr> <td colspan="3" style="text-align: center;">:</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1: 7 bytes (+ checksum)</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0: 8 bytes (+ checksum)</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1: 8 bytes (+ checksum)</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0: 8 bytes (+ checksum)</td> </tr> <tr> <td colspan="3" style="text-align: center;">:</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1: 8 bytes (+ checksum)</td> </tr> </table>	b3	b0		0	0	0: 0 byte (+ checksum)	0	0	1: 1 byte (+ checksum)	0	0	1	0: 2 bytes (+ checksum)	:			0	1	1	1: 7 bytes (+ checksum)	1	0	0	0: 8 bytes (+ checksum)	1	0	0	1: 8 bytes (+ checksum)	1	0	1	0: 8 bytes (+ checksum)	:			1	1	1	1: 8 bytes (+ checksum)
b3	b0																																								
0	0	0: 0 byte (+ checksum)																																							
0	0	1: 1 byte (+ checksum)																																							
0	0	1	0: 2 bytes (+ checksum)																																						
:																																									
0	1	1	1: 7 bytes (+ checksum)																																						
1	0	0	0: 8 bytes (+ checksum)																																						
1	0	0	1: 8 bytes (+ checksum)																																						
1	0	1	0: 8 bytes (+ checksum)																																						
:																																									
1	1	1	1: 8 bytes (+ checksum)																																						

LSS Bit (Transmission/Reception Continuation Select)

The LSS bit indicates that the data group to be transmitted or received next is not the last data group when response data of 9 bytes or more is to be transmitted or received. With 0 set, data and checksum are transmitted or received because the next data group to be transmitted or received is the last one.

With 1 set, only data is transmitted or received, and the checksum is not included because the next data group to be transmitted or received is not the last one.

During LIN communication, do not set “1” to this bit.

Set this bit when the RTS bit is 0 (response transmission/reception stopped).

LCS Bit (Checksum Select)

The LCS bit sets the checksum mode.

With 0 set, classic checksum mode is selected.

With 1 set, enhanced checksum mode is selected.

When the timeout error is used (the TERE bit in the RLN3nLEDE register is 1), the specific timeout time depends on

the setting of this bit. For details on the bit error, see **Section 19.5.3.7, Error Status**.

When the length of the response field data is 0 bytes (the RFDL bit is 0), do not set this bit to “1” (enhanced).

When response data of 9 bytes or more is to be transmitted or received, do not change the LCS bit setting after the first data group through the last data group.

During transmission or reception of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

When response data of 9 bytes or more is to be transmitted or received, set the FSM bit to 1.

Set this bit when the RTS bit is 0 (response transmission/reception stopped).

RCDS Bit (Response Field Communication Direction Select)

This bit selects the direction of the response field/wake-up signal communication.

With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (input signal low-level width count).

With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.

Set this bit when the RTS bit in the RLIN3nLTRC register is 0 in LIN operation mode (response transmission/reception stopped) or when the FTS bit is 0 in LIN wake-up mode (header reception or wake-up transmission/reception stopped).

When response data of 9 bytes or more is to be transmitted or received, do not change this bit setting after the first data group through the last data group.

RFDL[3:0] Bits (Response Field Length Select)

The RFDL bits set the length of the response field data.

The data length can be 0 to 8 bytes excluding the checksum size.

Set these bits when the RTS bit in the RLIN3nLTRC register is 0 (response transmission/reception stopped).

When response data of 9 bytes or more is to be transmitted or received, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit in the RLIN3nLDFC register is 1) include the checksum.

19.3.3.16 RLN3nLIDB — LIN ID Buffer Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	IDP1	IDP0	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.45 RLN3nLIDB Register Contents

Bit Position	Bit Name	Function
7, 6	IDP[1:0]	Parity Setting Stores the parity bits (P0 and P1) to be received in the ID field.
5 to 0	ID[5:0]	ID Setting Stores the 6-bit ID value received in the ID field.

The value in the RLN3nLIDB register is enabled after the completion of header reception. In LIN mode (LIN operation mode, LIN wake-up mode), writing to this register is disabled.

In LIN Self-Test mode, the operation is as follows.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN Self-Test mode, see **Section 19.5.5, LIN Self-Test Mode**.

IDP[1:0] Bits (Parity Setting)

The IDP bits store the parity bits (P0 and P1) received in the ID field of the LIN frame. IDP0 is for P0 and IDP1 is for P1.

When the IPERE bit in the RLN3nLEDE register is 1 (ID parity detection enabled), the received value and the value calculated internally are checked. If they do not match, IPER (ID parity error flag) is set.

ID[5:0] Bits (ID Setting)

The ID bits store the 6-bit ID value received in the ID field of the LIN frame.

19.3.3.17 RLN3nLCBR — LIN Checksum Buffer Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CKSM[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.46 RLN3nLCBR Register Contents

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Holds the transmitted or received checksum data.

In LIN operation mode, this register operates as follows:

- When the RCDS bit in the RLN3nLDFC register is 1 (transmission):
The value transmitted can be read from the register. Read the value after transmission is completed.
Writing to this register is invalid.
- When the RCDS bit in the RLN3nLDFC register is 0 (reception):
The value received can be read from the register. Read the value after reception is completed.
Writing to this register is invalid.

When response data of 9 bytes or more is to be transmitted or received, the checksum is appended only to the last data group; this register is not updated for the other data groups.

In LIN Self-Test mode, this register operates as follows:

- When the RCDS bit in the RLN3nLDFC register is 1 (transmission):
After completion of the frame transmission (after loopback), the reversed value of the received value can be read.
- When the RCDS bit in the RLN3nLDFC register is 0 (reception):
Write the value to be received before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN Self-Test mode, see **Section 19.5.5, LIN Self-Test Mode**.

Set the RLN3nLCBR register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

19.3.3.18 RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)

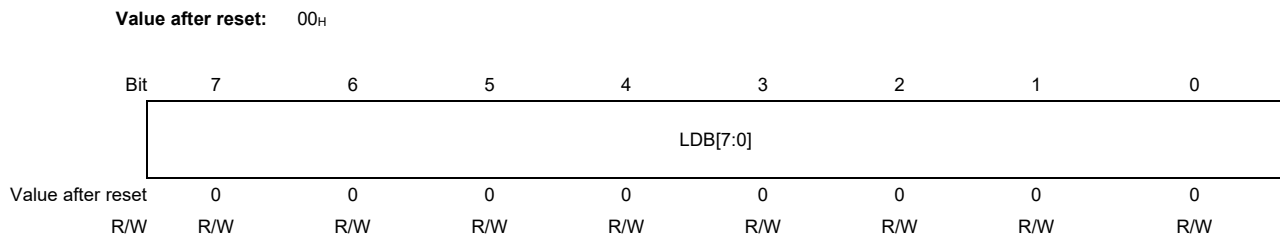


Table 19.47 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or holds the received data. Setting range: 00 _H to FF _H

- For response transmission:
The RLN3nLDBRb registers set the data to be transmitted in the response field.
Set these registers when the RTS bit in the RLN3nLTRC register is 0 (response transmission/reception is halted).
- For response reception:
The RLN3nLDBRb registers hold the data received in the response field.
The received data is overwritten. If an error is detected, the data up to the byte in which the error was detected are stored in the register.
Do not read these registers when the RTS bit is 1 (response transmission/reception is started)

In LIN Self-Test mode, the operation is as follows.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN Self-Test mode, see **Section 19.5.5, LIN Self-Test Mode**.

19.3.4 UART Related Registers

19.3.4.1 RLN3nLWBR — UART Wake-Up Baud Rate Select Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NSPB[3:0]				LPRS[2:0]			—
Value after reset	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 19.48 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 1 0 1: 6 sampling 0 1 1 0: 7 sampling 0 1 1 1: 8 sampling 1 0 0 0: 9 sampling 1 0 0 1: 10 sampling 1 0 1 0: 11 sampling 1 0 1 1: 12 sampling 1 1 0 0: 13 sampling 1 1 0 1: 14 sampling 1 1 1 0: 15 sampling 1 1 1 1: 16 sampling Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Set the LN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

The NSPB bits select the number of sampling in one Tbit (reciprocal of the baud rate).

In UART mode, it is possible to set the NSPB bits from 6 sampling to 16 sampling.

LPRS[2:0] Bits (Prescaler Clock Select)

The LPRS bits select the frequency division ratio for the prescaler.

The LIN communication clock source is divided by this prescaler.

19.3.4.2 RLN3nLBRP01 — UART Baud Rate Prescaler 01 Register

Value after reset: 0000_H

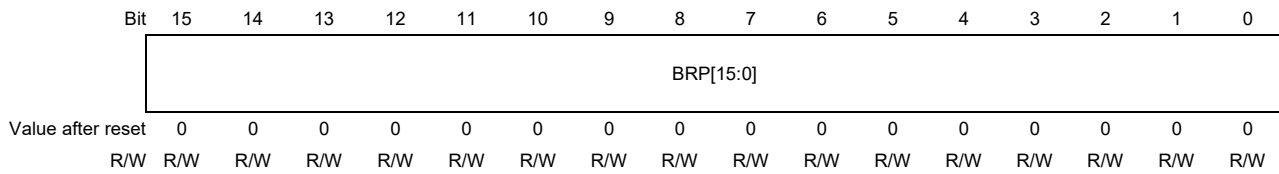


Table 19.49 RLN3nLBRP01 Register Contents

Bit Position	Bit Name	Function
15 to 0	BRP [15:0]	Assuming that the value set in this register is L (0 to 65535), the baud rate prescaler divides the frequency of the prescaler clock by L + 1. Setting range: 0000 _H to FFFF _H

Assuming that the value set in this register is L, the baud rate prescaler divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) in the RLN3nLWBR register by L + 1.

The RLN3nLBRP01 register can be accessed in 8-bit units by the registers RLN3nLBRP0 and RLN3nLBRP1.

19.3.4.3 RLN3nLMD — UART Mode Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	—	—	—	LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R	R	R/W	R/W

Table 19.50 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	LRDNFS	UART Reception Data Noise Filtering Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	LMD[1:0]	LIN/UART Mode Select b1 b0 0 1: UART mode

Set the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

LRDNFS Bit (UART Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.

With 0 set, the noise filter is enabled when receiving data.

With 1 set, the noise filter is disabled when receiving data.

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.

To use the LIN/UART interface as an UART, set these bits to 01_B.

19.3.4.4 RLN3nLBFC — UART Configuration Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	UTPS	URPS	UPS[1:0]		USBLS	UBOS	UBLS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.51 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	UTPS	UART Output Polarity Switch 0: Transmit data normal output 1: Transmit data inverted output
5	URPS	UART Input Polarity Switch 0: Reception data normal output 1: Reception data inverted output
4, 3	UPS[1:0]	UART Parity Select 00: Parity disabled 01: Even parity 10: 0 parity 11: Odd parity
2	USBLS	UART Stop Bit length Select 0: Stop bit: 1 bit 1: Stop bit: 2 bits
1	UBOS	UART Transfer Format Order Select 0: LSB first 1: MSB first
0	UBLS	UART Character Length Select 0: UART 8 bits communication 1: UART 7 bits communication

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

UTPS Bit (UART Output Polarity Switch)

Sets the output polarity for UART communication.

With 0 set, transmit data is output without inversion.

With 1 set, inverted transmit data is output.

The setting of this bit is valid for all the bits of the UART frame.

In half-duplex communication, this setting should match the setting of URPS bit.

URPS Bit (UART Input Polarity Switch)

Sets the input polarity for UART communication.

With 0 set, receive data is input without inversion.

With 1 set, receive data is input with inversion.

The setting of this bit is valid for all the bits of the UART frame.

In half-duplex communication, this setting should match the setting of UTPS bit.

UPS[1:0] Bits (UART Parity Select)

Sets the UART parity.

- When these bits are set to “00_B”, data is transmitted/received without the parity.

[Transmission]

A parity bit is not added to transmit data.

[Reception]

Data is received without parity processing. Therefore, a parity error does not occur.

- When these bits are set to “01_B”, data is transmitted/received with the even parity.

[Transmission]

If the number of 1s in transmit data is odd, “1” is added to the parity bit. If the number of 1s in transmit data is even, “0” is added to the parity bit.

[Reception]

If the number of 1s in receive data including the parity bit is odd, a parity error occurs.

- When these bits are set to “10_B”, data is transmitted/received with 0 parity.

[Transmission]

Regardless of the number of 1s in transmit data, “0” is added to the parity bit.

[Reception]

The value of the parity bit is not judged. Therefore, no parity error occurs.

- When these bits are set to “11_B”, data is transmitted/received with the odd parity.

[Transmission]

If the number of 1s in transmit data is odd, “0” is added to the parity bit. If the number of 1s in transmit data is even, “1” is added to the parity bit.

[Reception]

If the number of 1s in receive data including the parity bit is even, a parity error occurs.

USBLS Bit (UART Stop Bit Length Select)

Sets the stop bit length of data for UART communication.

With 0 set, stop bit length of 1 bit is selected.

With 1 set, stop bit length of 2 bits is selected.

UBOS Bit (UART Transfer Format Select)

Sets the bit order of data for UART communication.

With 0 set, LSB first is selected.

With 1 set, MSB first is selected.

UBLS Bit (UART Character Length Select)

Sets the character length of one frame for UART communication.

With 0 set, the character length is 8 bits.

With 1 set, the character length is 7 bits.

When the character length of one frame is 9 bits (the UEBE bit in the RLIN3nLUOR1 register is 1), the setting of this bit is invalid.

19.3.4.5 RLN3nLSC — UART Space Configuration Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]		—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R

Table 19.52 RLN3nLSC Register Contents

Bit Position	Bit Name	Function															
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.															
5, 4	IBS[1:0]	Inter-Byte Space Select <table border="0"> <tr> <td>b5</td> <td>b4</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0 Tbit</td> </tr> <tr> <td>0</td> <td>1</td> <td>1 Tbit</td> </tr> <tr> <td>1</td> <td>0</td> <td>2 Tbits</td> </tr> <tr> <td>1</td> <td>1</td> <td>3 Tbits</td> </tr> </table>	b5	b4		0	0	0 Tbit	0	1	1 Tbit	1	0	2 Tbits	1	1	3 Tbits
b5	b4																
0	0	0 Tbit															
0	1	1 Tbit															
1	0	2 Tbits															
1	1	3 Tbits															
3 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.															

Set the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the inter-byte space during multi-byte UART transmission.
0 to 3 Tbits can be set.

19.3.4.6 RLN3nLEDE —UART Error Detection Enable Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	FERE	OERE	—	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R/W

Table 19.53 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	FERE	Framing Error Detection Enable 0: Disables framing error detection. 1: Enables framing error detection.
2	OERE	Overrun Error Detection Enable 0: Disables overrun error detection. 1: Enables overrun error detection.
1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	BERE	Bit Error Detection Enable 0: Disables bit error detection. 1: Enables bit error detection.

Set the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.

With 0 set, the framing error is not detected.

With 1 set, the framing error is detected.

When this bit is set to 1, the detection result is indicated in the FER flag in the RLN3nLEST register.

For details on the framing error, see **Section 19.5.4.5, Error Status**.

OERE Bit (Overrun Error Detection Enable)

This bit enables or disables detection of the overrun error.

With 0 set, the overrun error is not detected.

With 1 set, the overrun error is detected.

When this bit is set to 1, the detection result is reflected in the OER flag in the RLN3nLEST register.

For details on the overrun error, see **Section 19.5.4.5, Error Status**.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.

With 0 set, the bit error is not detected.

With 1 set, the bit error is detected.

When this bit is set to 1, the detection result is indicated in the BER flag in the RLN3nLEST register.

In full-duplex communication, do not set this bit to “1”.

Do not set this register when the NSPB bits in the RLN3nLWBR register are 0101_B (6 samplings) and the LRDNFS bit

in the RLN3nLMD register is 0 (noise filtering is enabled).
For details on the bit error, see **Section 19.5.4.5, Error Status**.

19.3.4.7 RLN3nLCUC — UART Control Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 19.54 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	OM0	LIN Reset 0: LIN reset mode. 1: LIN reset mode is canceled.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

OM0 Bit (LIN Reset)

The OM0 bit selects either causing a transition to reset mode or canceling reset mode.

With 0 set, reset mode.

With 1 set, reset mode is canceled.

It takes 3 peripheral clock cycles + 4 communication clock cycles, until the OMM0 bit in the RLN3nLMST register reflects the OM0 bit setting in the RLN3nLCUC register.

19.3.4.8 RLN3nLTRC — UART Transmission Control Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTS	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R

Table 19.55 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	RTS	UART Buffer Transmission Start 0: UART Buffer transmission is stopped. 1: UART Buffer transmission is started.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

RTS Bit (UART Buffer Transmission Start)

When transmitting data from the UART buffer, set this bit to “1”.

Only 1 can be written to this bit; 0 cannot be written.

Write to this bit when the UTOE bit in the RLN3nLUOER register is 1 (transmission enabled) and the UTS bit in the RLN3nLST register is 0 (transmission is not in progress).

Once set, regardless of errors, this bit is automatically cleared to 0 upon completion of the number of data transmission specified by the MDL bit in the RLN3nLDFC register. Moreover, this bit is automatically cleared to 0 upon transition to reset mode.

Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (in LIN reset mode).

When writing 1 to this bit while the UTSW bit in the RLN3nLDFC register is 1 (when UART buffer transmission is requested, the start of transmission is delayed until the stop bit of reception data is completed), write only during the reception of stop bit.

19.3.4.9 RLN3nLMST — UART Mode Status Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 19.56 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

It takes 3 peripheral clock cycles + 4 communication clock cycles, until the OMM0 bit in the RLN3nLMST register reflects the OMM0 bit setting in the RLN3nLCUC register.

19.3.4.10 RLN3nLST — UART Status Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	URS	UTS	ERR	—	—	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 19.57 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	URS	UART Reception Status Flag 0: Reception is not operated. 1: Reception is operated.
4	UTS	UART Transmission Status Flag 0: Transmission is not operated. 1: Transmission is operated.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2, 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	FTC	Successful UART Buffer Transmission Flag 0: UART buffer transmission has not been completed. 1: UART buffer transmission has been completed.

The RLN3nLST register is automatically cleared to “00_H” upon transition to LIN reset mode. In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains “00_H”. To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

URS Flag (UART Reception Status Flag)

At the start of the reception, this flag is set to 1.

The reception is started under the following condition.

- When the start bit is detected

At the end of reception, this flag is cleared to 0. While reception is stopped, this flag retains 0.

The reception is ended under the following condition.

- Sampling point of the first bit of the stop bits

UTS Flag (UART Transmission Status Flag)

At the start of the transmission, this flag is set to 1. During the transmission, this flag retains 1.

The transmission is started under the following conditions.

- When transmission data is set to the RLN3nLUTDR or RLN3nLUWTDR register
- When the RTS bit in the RLN3nLTRC register is set to 1

This flag is cleared to 0 at the end of transmission.

The transmission is ended under the following conditions.

- When transmission of data set in the RLIN3nLUTDR or RLIN3nLUWTDRE register is completed and next data is not set
- When the UART buffer transmission is completed (when the RTS bit in the RLIN3nLTRC register is cleared to 0)
- When transmission operation enable bit UTOE in register LUOER is cleared.

ERR Flag (Error Detection Flag)

This flag is set to 1 upon detection of an error, detection of an expansion bit, or upon ID matching (when at least one of the flags of the RLIN3nLEST register is 1). At this time, an interrupt request for RLIN3n status is generated. However, when this bit is 1, an interrupt is not generated upon detection of an error, detection of an expansion bit, or upon ID matching. To clear the bit to 0, write 0 to the UPER, IDMT, EXBT, FER, OER, and BER flags in the RLIN3nLEST register.

FTC Flag (Successful UART Buffer Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that it was before 1 was written. Regardless of errors, this bit is set to 1 upon completion of transmission of the number of data specified by the MDL bit in the RLIN3nLDFC register from the UART buffer. At this time, an interrupt request for RLIN3n transmission is generated. Write 0 to clear this flag.

19.3.4.11 RLN3nLEST — UART Error Status Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	UPER	IDMT	EXBT	FER	OER	—	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R	R/W

Table 19.58 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	UPER	Parity Error Flag 0: Parity error has not been detected. 1: Parity error has been detected.
5	IDMT	ID Matching Flag 0: The receive data does not match with the ID value. 1: The receive data matches with the ID value.
4	EXBT	Expanded Bit Detection Flag 0: Expanded bit has not been detected. 1: Expanded bit has been detected.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	OER	Overrun Error Flag 0: Overrun error has not been detected. 1: Overrun error has been detected.
1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode. In LIN reset mode, this register cannot be written to, and the value of 00_H is held. To clear certain bits in this register, write 0 to those bits, and write 1 to the bits not to be cleared by using the store instruction.

UPER Flag (Parity Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that it was before 1 was written. This flag is set to 1 upon parity error detection. Write 0 to clear this flag.

IDMT Flag (ID Matching Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that it was before 1 was written. The IDMT flag becomes 1 when all the following conditions are met:

- The UEBE bit in the RLN3nLUOR1 register is 1 (expansion bit enabled)
- The UECD bit in the RLN3nLUOR1 register is 0 (expansion bit comparison enabled)

- The UEBDCE bit in the RLN3nLUOR1 register is 1 (data comparison after expansion bit is detected)
 - The received expansion bit and the value of the UEBDL bit of the RLN3nLUOR1 register match.
 - The value of the value of the 8-bit of the received data excluding the expansion bit and the value of the RLN3nLIDB register match.

Write 0 to clear this flag.

EXBT Flag (Expansion Bit Detection Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that it was before 1 was written.

When the UEBE bit in the RLN3nLUOR1 register is 1 (expansion bit enable), if the received expansion bit matches the UEBDL bit in the RLN3nLUOR1 register, this flag is set to 1.

Write 0 to clear this flag.

FER Flag (Framing Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value it was before 1 was written.

The FER flag is set to 1 upon framing error detection while the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled). Write 0 to clear this flag.

Framing error is detected always on the first stop bit, regardless of the stop bit amount setting.

OER Flag (Overrun Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value Write 0 to clear this flag.

The OER flag is set to 1 upon overrun error detection while the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). Write 0 to clear this flag.

BER Flag (Bit Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that it was before 1 was written.

The BER flag is set to 1 when the transmitted data and the data monitored by the receive pin do not match while the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled).

Write 0 to clear this flag.

19.3.4.12 RLN3nLDFC — UART Data Field Configuration Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	UTSW	—	MDL[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R/W	R/W	R/W	R/W

Table 19.59 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function																																										
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.																																										
5	UTSW	UART Transmission Start Wait 0: When UART buffer transmission is requested, transmission is started immediately. 1: When UART buffer transmission is requested, transmission is not started until reception of the stop bit is completed.																																										
4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.																																										
3 to 0	MDL[3:0]	UART Buffer Data Length Select <table border="0" style="margin-left: 20px;"> <tr> <td>b3</td> <td>b0</td> <td></td> </tr> <tr> <td>0 0 0 0</td> <td></td> <td>9 bytes</td> </tr> <tr> <td>0 0 0 1</td> <td></td> <td>1 bytes</td> </tr> <tr> <td>0 0 1 0</td> <td></td> <td>2 bytes</td> </tr> <tr> <td>0 0 1 1</td> <td></td> <td>3 bytes</td> </tr> <tr> <td>0 1 0 0</td> <td></td> <td>4 bytes</td> </tr> <tr> <td>0 1 0 1</td> <td></td> <td>5 bytes</td> </tr> <tr> <td>0 1 1 0</td> <td></td> <td>6 bytes</td> </tr> <tr> <td>0 1 1 1</td> <td></td> <td>7 bytes</td> </tr> <tr> <td>1 0 0 0</td> <td></td> <td>8 bytes</td> </tr> <tr> <td>1 0 0 1</td> <td></td> <td>9 bytes</td> </tr> <tr> <td>1 0 1 0</td> <td></td> <td>9 bytes</td> </tr> <tr> <td></td> <td></td> <td>:</td> </tr> <tr> <td>1 1 1 1</td> <td></td> <td>9 bytes</td> </tr> </table>	b3	b0		0 0 0 0		9 bytes	0 0 0 1		1 bytes	0 0 1 0		2 bytes	0 0 1 1		3 bytes	0 1 0 0		4 bytes	0 1 0 1		5 bytes	0 1 1 0		6 bytes	0 1 1 1		7 bytes	1 0 0 0		8 bytes	1 0 0 1		9 bytes	1 0 1 0		9 bytes			:	1 1 1 1		9 bytes
b3	b0																																											
0 0 0 0		9 bytes																																										
0 0 0 1		1 bytes																																										
0 0 1 0		2 bytes																																										
0 0 1 1		3 bytes																																										
0 1 0 0		4 bytes																																										
0 1 0 1		5 bytes																																										
0 1 1 0		6 bytes																																										
0 1 1 1		7 bytes																																										
1 0 0 0		8 bytes																																										
1 0 0 1		9 bytes																																										
1 0 1 0		9 bytes																																										
		:																																										
1 1 1 1		9 bytes																																										

UTSW Bit (UART Transmission Start Wait)

This bit controls the start timing of UART buffer transmission.

With 0 set, transmission is started as soon as the start of UART buffer transmission is requested.

With 1 set, transmission is started after the completion of the stop bit reception.

Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the RLN3nLBFC register.

This bit is enabled when the RTS bit in the RLN3nLTRC register is set to 1. In addition, writing a value to this bit is disabled when the RTS bit is 1 (UART buffer transmission started).

Set this bit to 1 only to switch from reception to transmission in half-duplex communication.

MDL[3:0] Bits (UART Buffer Data Length Select)

These bits specify the data length of the UART buffer.

Writing a value to these bits is disabled when the RTS bit in the RLN3nLTRC register is 1 (UART buffer transmission started).

19.3.4.13 RLN3nLIDB — UART ID Buffer Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	ID[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.60 RLN3nLIDB Register Contents

Bit Position	Bit Name	Function
7 to 0	ID[7:0]	ID value that is referred for the expansion bit data comparison is set

ID Bits (ID)

When the UEBE bit in the RLN3nLUOR1 register is set to 1 (expansion bit enabled), the UECD bit is set to 0 (expansion bit comparison enabled), and the UEBDCE bit is set to 1 (data comparison after expansion bit is detected), set the value to be compared with the received data. Write to the RLN3nLIDB register when the URS bit in the RLN3nLST register is 0 (receive operation is not in progress).

19.3.4.14 RLN3nLUDB0 — UART Data Buffer 0 Register

Value after reset: 00_H

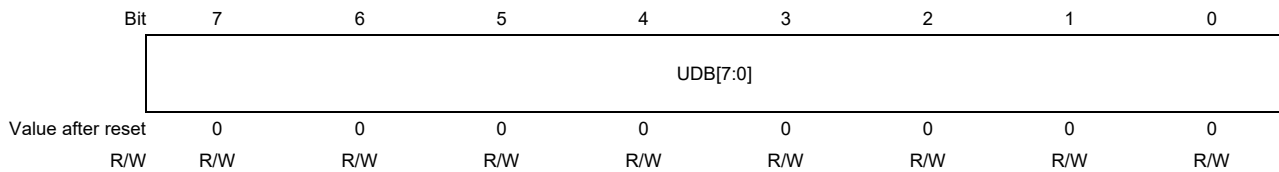


Table 19.61 RLN3nLUDB0 Register Contents

Bit Position	Bit Name	Function
7 to 0	UDB[7:0]	Value of UART transmission data

When transmitting 9-byte data from the UART buffer transmission (the RLN3nLDFC.MDL bit is 0_H or 9_H), set the first data to be transmitted.

Write to the RLN3nLUDB0 register when the RTS bit of the RLN3nLTRC register is 0 (UART buffer transmission stopped).

Table 19.62 shows the bit arrangement according to the settings for communication format.

For details about the UART buffer transmission, see **Section 19.5.4.1(2) UART Buffer Transmission**.

Table 19.62 Bit Arrangement of the RLN3nLUDB0 Register According to Each Communication Format

	RLN3nLUDB0							
	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7-bit; MSB first	—	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
8-bit; LSB first	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8-bit; MSB first	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7

19.3.4.15 RLN3nLDBRb — UART Data Buffer b Register (b = 1 to 8)

Value after reset: 00_H

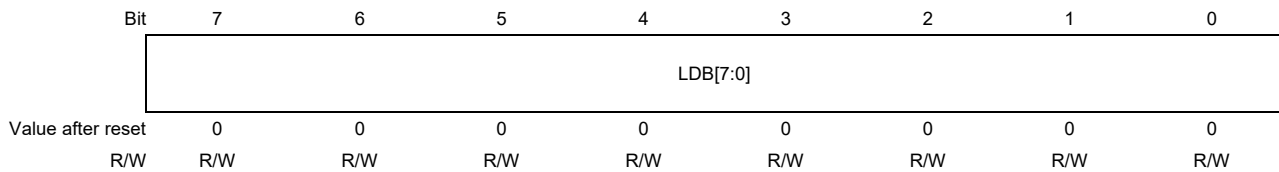


Table 19.63 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Value of UART transmission data

This register specifies the data transmitted from the UART buffer.

Write to these registers when the RTS bit of the RLN3nLTRC register is 0 (UART buffer transmission stopped).

Table 19.64 shows the bit arrangement according to the set communication format.

For details about the UART buffer transmission, see **Section 19.5.4.1(2) UART Buffer Transmission**.

Table 19.64 Bit Arrangement of the RLN3nLDBRb Register According to Each Communication Format

	RLN3nLDBRb							
	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7-bit; MSB first	—	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
8-bit; LSB first	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8-bit; MSB first	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7

19.3.4.16 RLN3nLUOER — UART Operation Enable Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	UROE	UTOE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 19.65 RLN3nLUOER Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	UROE	UART Reception Enable 0: Disables reception. 1: Enables reception.
0	UTOE	UART Transmission Enable 0: Disables transmission. 1: Enables transmission.

The RLN3nLUOER register is automatically cleared to 00_H upon transition to LIN reset mode.

In LIN reset mode, this register cannot be written to.

In LIN reset mode, the register retains 00_H.

UROE Bit (UART Reception Enable)

The UROE bit enables or disables reception.

With 0 set, reception is disabled.

With 1 set, reception is enabled.

Do not clear this bit during reception. If the communication is suspended during reception, set the OM0 bit in the RLN3nLCUC register to 0 (in LIN reset mode) to transition to the LIN reset mode. However, the transmit operation is also suspended at this time.

Do not set this bit to 1 when data transmission from the UART buffer is in progress.

UTOE Bit (UART Transmission Enable)

The UTOE bit enables or disables transmission.

With 0 set, transmission is disabled.

With 1 set, transmission is enabled.

Do not clear this bit during transmission. If the communication is suspended during transmission, set the OM0 bit in the RLN3nLCUC register to 0 (in LIN reset mode) to transition to the LIN reset mode. However, the receive operation is also suspended at this time.

19.3.4.17 RLN3nLUOR1 — UART Option Register 1

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	UECD	UTIGTS	UEBDCE	UEBDL	UEBE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 19.66 RLN3nLUOR1 Register Contents

Bit Position	Bit Name	Function
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	UECD	UART Expansion Bit Comparison Disable 0: Enables expansion bit comparison. 1: Disables expansion bit comparison.
3	UTIGTS	UART Transmission Interrupt Generation Timing Select 0: Transmission interrupt is generated at the start of transmission. 1: Transmission interrupt is generated at the completion of transmission.
2	UEBDCE	UART Expansion Bit Data Comparison Enable 0: Disables data comparison after an expansion bit is detected. 1: Enables data comparison after an expansion bit is detected.
1	UEBDL	UART Expansion Bit Detection Level Select 0: Selects expansion bit value 0 as the expansion bit detection level. 1: Selects expansion bit value 1 as the expansion bit detection level.
0	UEBE	UART Expansion Bit Enable 0: Disables expansion bit operation. 1: Enables expansion bit operation.

UECD Bit (Expansion Bit Comparison Disable)

The UECD bit enables or disables comparison between the received expansion bit and the UEBDL bit value when the UEBE bit is 1 (expansion bit operation is enabled).

With 0 set, comparison between the received expansion bit and the UEBDL bit value is enabled when the expansion bit is received.

With 1 set, comparison between the received expansion bit and the UEBDL bit value is disabled when the expansion bit is received.

Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (in LIN reset mode).

Do not set this bit to 1 when the UART buffer is used.

Do not set this bit to 1 when the UEBDCE bit is 1 (data comparison after expansion bit is detected).

UTIGTS Bit (Transmission Interrupt Generation Timing Select)

The UTIGTS bit sets the generation timing of the transmission interrupt.

With 0 set, the transmission interrupt is generated at the start of transmission.

With 1 set, the transmission interrupt is generated at the completion of transmission.

When transmission from the UART buffer is performed with 0 set, the transmission interrupt is generated only at the start of the transmission of the last data of the data length set with the MDL bits in the RLN3nLDFC register.

When transmission from the UART buffer is performed with 1 set, the transmission interrupt is generated only at the completion of the transmission of the last data of the data length set with the MDL bits in the RLN3nLDFC register.

UEBDCE Bit (Expansion Bit Data Comparison Enable)

After an expansion bit is detected, this bit enables or disables the comparison between the 8-bit receive data excluding the expansion bit and the value of the RLN3nLIDB register.

With 0 set, when the level selected by the UEBDL bit is detected as an expansion bit, the comparison between the received value in the RLN3nLURDR register and the value of the RLN3nLIDB register is disabled.

With 1 set, when the level selected by the UEBDL bit is detected as an expansion bit, the comparison between the received value in the RLN3nLURDR register and the value of the RLN3nLIDB register is enabled.

Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (in LIN reset mode).

Do not set this bit to 1 when the UEBE bit is 0 (expansion bit operation disabled).

Do not set this bit to 1 when the UECD bit is 1 (expansion bit comparison disabled).

Do not set this bit to 1 when the UART buffer is used.

UEBDL Bit (Expansion Bit Detection Level Select)

The UEBDL bit sets the level to be detected as the expansion bit when the UEBE bit is 1 (expansion bit operation is enabled) and the UECD bit is 0 (comparison of the expansion bit is enabled).

With 0 set, expansion bit value 0 is the level to be detected as the expansion bit.

With 1 set, expansion bit value 1 is the level to be detected as the expansion bit.

Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (in LIN reset mode).

Do not set this bit to 1 when the UART buffer is used.

UEBE Bit (Expansion Bit Enable Bit)

The UEBE bit enables or disables expansion bit operation.

With 0 set, expansion bit operation is disabled.

With 1 set, expansion bit operation is enabled.

Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (in LIN reset mode).

Do not set this bit to 1 when the UART buffer is used.

19.3.4.18 RLN3nLUTDR — UART Transmission Data Register

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UTD[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.67 RLN3nLUTDR Register Contents

Bit Position	Bit Name	Function
15 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	UTD[8:0]	Value of UART transmission data. Setting range: 000 _H to 1FF _H

The RLN3nLUTDR register sets the data to be transmitted from the transmit data register.

Writing data to this register with the UTOE bit in the RLN3nLUOER register set to 1 starts transmission.

This register can be accessed in 8 bits.

In 9-bit communication mode, do not attempt 8-bit access.

Do not write data to this register when data transmission from the UART buffer is in progress.

Also, do not write data to this register when a transmission request is being generated due to write access to the RLN3nLUWTDR register.

When transmitting data continuously, do not set another piece of transmission data in this register before the generation of transmission interrupt.

The table below shows the bit arrangement according to the set communication format.

Table 19.68 Bit Arrangement of the RLN3nLUTDR Register According to Each Communication Format

	RLN3nLUTDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	—	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
7-bit; MSB first	—	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
8-bit; LSB first	—	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8-bit; MSB first	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
9-bit; LSB first	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
9-bit; MSB first	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8

19.3.4.19 RLN3nLURDR — UART Reception Data Register

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	URD [8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.69 RLN3nLURDR Register Contents

Bit Position	Bit Name	Function
15 to 9	—	Reserved When read, the value after reset is returned.
8 to 0	URD [8:0]	Value of UART reception data. Setting range: 000 _H to 1FF _H

The RLN3nLURDR allows the reception data to be read from the receive data register.

When the UROE bit in the RLN3nLUOER register is 1, the reception data is stored in this register and can be read out.

This register is updated when the stop bit of the receive data is received.

This register is also updated when an error is caused by the parity or stop bit.

However, the value of this register is not updated upon occurrence of an overrun error when the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). The value of this register is updated upon occurrence of an overrun error when the OERE bit is 0 (overrun error detection disabled).

Read this register upon occurrence of a receive error (overrun error, framing error, parity error) when the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). If the next data is received without reading this register, an overrun error occurs.

This register can be accessed in 8 bits.

However, during expansion bit use (UEBE bit of the RLN3nLUOR1 register is 1 (expansion bit operation enabled), do not attempt 8-bit access.

The table below shows the bit arrangement according to the set communication format.

Table 19.70 Bit Arrangement of the RLN3nLURDR Register According to Each Communication Format

	RLN3nLURDR									
	b8	b7	b6	b5	b4	b3	b2	b1	b0	
7-bit; LSB first	—	—	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
7-bit; MSB first	—	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	
8-bit; LSB first	—	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
8-bit; MSB first	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	
9-bit; LSB first	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
9-bit; MSB first	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	

19.3.4.20 RLN3nLUWTDR — UART Wait Transmission Data Register

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UWTD[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.71 RLN3nLUWTDR Register Contents

Bit Position	Bit Name	Function
15 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	UWTD[8:0]	Sets the data to be transmitted from the wait transmit data register after waiting for the stop bit reception to be completed. Setting range: 000 _H to 1FF _H

The RLN3nLUWTDR register sets the data to be transmitted from the UART wait transmit data register.

Writing data to this register with the UTOE bit in the RLN3nLUOER register set to 1 starts transmission.

Use this register only to switch from reception to transmission in half-duplex communication.

The user should write to this register only while the stop bit is being received.

Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the RLN3nLBFC register.

When this register is read, the RLN3nLUTDR register value is actually read.

In 9-bit communication mode, do not attempt 8-bit access.

Do not write data to this register when data transmission from the UART buffer is in progress.

The table below shows the bit arrangement according to the set communication format.

Table 19.72 Bit Arrangement of the RLN3nLUWTDR Register According to Each Communication Format

	RLN3nLUWTDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	—	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
7-bit; MSB first	—	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
8-bit; LSB first	—	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8-bit; MSB first	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
9-bit; LSB first	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
9-bit; MSB first	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8

19.4 Interrupt Sources

The LIN/UART interface generates three types of interrupt requests.

- RLIN3n successful transmission interrupt
- RLIN3n successful reception interrupt
- RLIN3n status interrupt

Setting the LIOS bit in the RLIN3nLMD register to 1 allows to output the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, or RLIN3n status interrupt depending on the interrupt source.

Table 19.73 lists the sources for each interrupt.

Table 19.73 Interrupt Sources

		LIOS bit in RLIN3nLMD register is 1*1		
		RLIN3n Transmission Interrupt	RLIN3n Reception Interrupt	RLIN3n Status Interrupt
LIN mode	LIN master mode	<ul style="list-style-type: none"> • Successful frame transmission • Successful wake-up transmission • Successful header transmission 	<ul style="list-style-type: none"> • Successful response reception • Successful wake-up reception 	<ul style="list-style-type: none"> • Bit error • Physical bus error • Frame/response timeout error • Framing error • Checksum error • Response preparation error
	LIN slave mode	<ul style="list-style-type: none"> • Successful response transmission • Successful wake-up transmission 	<ul style="list-style-type: none"> • Successful response reception • Successful wake-up reception • Successful header reception 	<ul style="list-style-type: none"> • Bit error • Frame/response timeout error • Framing error • Sync field error • Checksum error • ID parity error • Response preparation error
UART mode		<ul style="list-style-type: none"> • Transmission start/successful transmission 	<ul style="list-style-type: none"> • Successful reception • Expansion bit mismatch 	<ul style="list-style-type: none"> • Bit error • Overrun error • Framing error • Expansion bit detection • ID match • Parity error

Note 1. The LIOS bit setting is valid in LIN Mode. In UART mode, setting the LIOS bit is not required.

In LIN mode, each interrupt request is output when the corresponding bit in the RLIN3nLIE register is 1 (interrupt is enabled) and the corresponding flag in the RLIN3nLST register is 1.

19.5 Operation

19.5.1 Modes

The LIN/UART interface provides the following four modes, depending upon the specific function to be performed:

- LIN reset mode
- LIN mode
 - LIN master mode
 - LIN slave mode [auto baud rate]
 - LIN slave mode [fixed baud rate]
- UART mode
- LIN Self-Test mode

Figure 19.2 shows mode transitions. **Table 19.74** describes mode transition conditions. **Table 19.75** lists operations available in each mode.

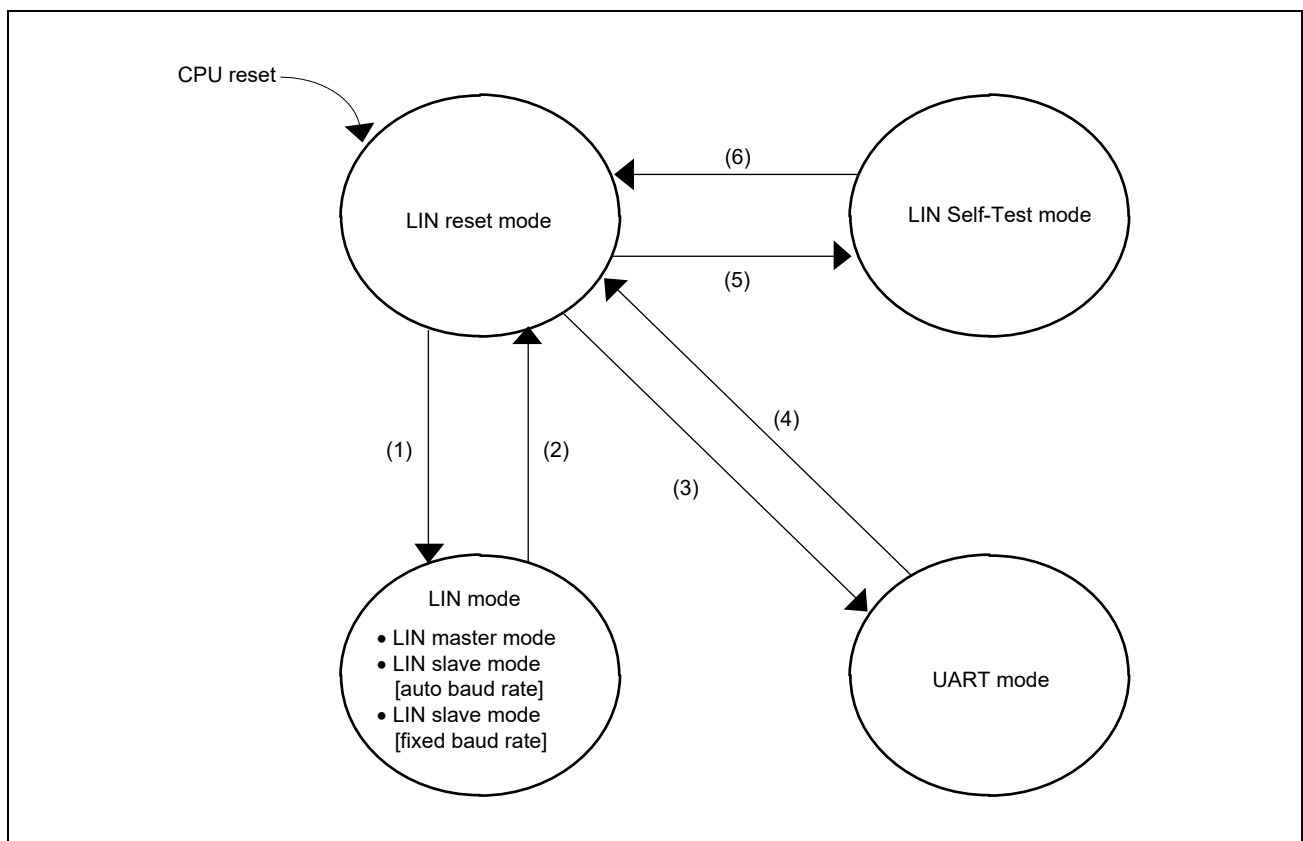


Figure 19.2 Mode Transitions

Table 19.74 Transition Condition of Each Mode

Mode Transition		Transition Condition
(1)	LIN reset mode → LIN mode <ul style="list-style-type: none"> • LIN master mode • LIN slave mode [auto baud rate] • LIN slave mode [fixed baud rate] 	<ul style="list-style-type: none"> • LMD bits in RLN3nLMD register = 00_B and OM1 and OM0 bits in RLN3nLCUC register = 01_B or 11_B • LMD bits in RLN3nLMD register = 10_B and OM1 and OM0 bits in RLN3nLCUC register = 01_B or 11_B • LMD bits in RLN3nLMD register = 11_B and OM1 and OM0 bits of RLN3nLCUC register = 01_B or 11_B
(2)	LIN mode → LIN reset mode	OM0 bit in RLN3nLCUC register = 0 _B
(3)	LIN reset mode → UART mode	LMD bits in RLN3nLMD register = 01 _B and OM0 bit in RLN3nLCUC register = 1 _B
(4)	UART mode → LIN reset mode	OM0 bit in RLN3nLCUC register = 0 _B
(5)	LIN reset mode → LIN Self-Test mode	See Section 19.5.5, LIN Self-Test Mode.
(6)	LIN Self-Test mode → LIN reset mode	See Section 19.5.5, LIN Self-Test Mode.

Table 19.75 Operations Available in Each Mode

LIN Mode		UART Mode	LIN Self-Test Mode
LIN Master Mode	LIN Slave Mode [auto baud rate] LIN Slave Mode [fixed baud rate]		
Header transmission	Header reception	UART transmission	Self test
Response transmission	Response transmission	UART reception	
Response reception	Response reception	Error detection	
Wake-up transmission	Wake-up transmission		
Wake-up reception	Wake-up reception		
Error detection	Error detection		

Whether mode has transitioned to LIN reset mode, the LIN mode, or the UART mode can be verified by reading the LMD bits in the RLN3nLMD register and the OMM0 bit in the RLN3nLMST register.

For a description of the LIN Self-Test mode, see **Section 19.5.5, LIN Self-Test Mode.**

19.5.2 LIN Reset Mode

Setting the OM0 bit in the RLN3nLCUC register to 0 (in LIN reset mode) causes a transition to LIN reset mode. The change to LIN reset mode can be verified by determining that the OMM0 bit in the RLN3nLMST register has been set to 0 (in LIN reset mode). In this mode, the LIN communication and the UART communication functions are halted.

From LIN reset mode, transitions to LIN mode, UART mode, and LIN Self-Test mode can be made.

When the mode changes to LIN reset mode, the following registers are initialized to their reset values, and as long as LIN reset mode is in effect, they retain their initial values.

- RLN3nLSTC register
- RLN3nLTRC register
- RLN3nLST register
- RLN3nLEST register
- RLN3nLUOER register

The following registers retain their previous values even when a transition to LIN reset mode is made:

- RLN3nLWBR register
- RLN3nLBRP0 register
- RLN3nLBRP1 register
- RLN3nLMD register
- RLN3nLBFC register
- RLN3nLSC register
- RLN3nLWUP register
- RLN3nLIE register
- RLN3nLEDE register
- RLN3nLDFC register
- RLN3nLIDB register
- RLN3nLCBR register
- RLN3nLUDB0 register
- RLN3nLDBRb register (b = 1 to 8)
- RLN3nLUOR1 register
- RLN3nLUTDR register
- RLN3nLURDR register
- RLN3nLUWTD R register

19.5.3 LIN Mode

LIN mode can operate in the following submodes:

LIN master mode, LIN slave mode [auto baud rate], and LIN slave mode [fixed baud rate].

In LIN master mode, the following operations can be performed: header transmission, response transmission, response reception, wake-up transmission, wake-up reception, and error detection. In LIN reset mode, setting the LMD bits in the RLN3nLMD register to 00_B (in LIN master mode) and the OM1 and OM0 bits in the RLN3nLCUC register to either 01_B or 11_B sets LIN master mode, turning the OMM1 and OMM0 bits in the RLN3nLMST register to either 01_B to 11_B.

In LIN slave mode [auto baud rate] and LIN slave mode [fixed baud rate], header reception, response transmission, response reception, wake-up transmission, wake-up reception, and error detection can be performed.

The LIN slave mode [auto baud rate] allows automatic detection of the break field and the sync field, and sets a baud rate based on the results of measurement of a sync field. The baud rate can be set to 1 kbps to 20 kbps.

Set the LPRS[2:0] bits in the RLN3nLWBR register so that the prescaler clock (with the frequency of the LIN communication clock source divided by the prescaler) becomes as follows according to the target baud rate.

[Target baud rate]	[Prescaler clock]
1 kbps to 20 kbps:	4 MHz* ¹
1 kbps to 2.4 kbps (excluding 2.4 kbps):	4 MHz
2.4 kbps to 20 kbps:	8 MHz to 12 MHz

Note 1. Use the clock with NSPB[3:0] bits in the RLN3nLWBR register set to "0011_B" (four samplings).

LIN slave mode [fixed baud rate] allows automatic detection of the break field, sync field, and ID field at a baud rate that is set in advance by the baud rate generator.

In LIN reset mode, setting the LMD bits in the RLN3nLMD register to 10_B (in LIN slave mode [auto baud rate]) and setting the OM1 and OM0 bits in the RLN3nLCUC register to 01_B or 11_B sets LIN slave mode [auto baud rate]; and setting the LMD bits in the RLN3nLMD register to 11_B (in LIN slave mode [fixed baud rate]), and setting the OM1 and OM0 bits in the RLN3nLCUC register to 01_B or 11_B sets LIN slave mode [fixed baud rate], turning the OMM1 and OMM0 bits in the RLN3nLMST register to 01_B or 11_B.

When changing a submode to another submode within LIN mode, a transition to LIN reset mode should first be made and change the LMD bits in the RLN3nLMD register.

The LIN mode provides the following two operation modes:

- LIN operation mode
- LIN wake-up mode

Figure 19.3 shows the transition of operation modes. **Table 19.76** describes the transition conditions of operation modes.

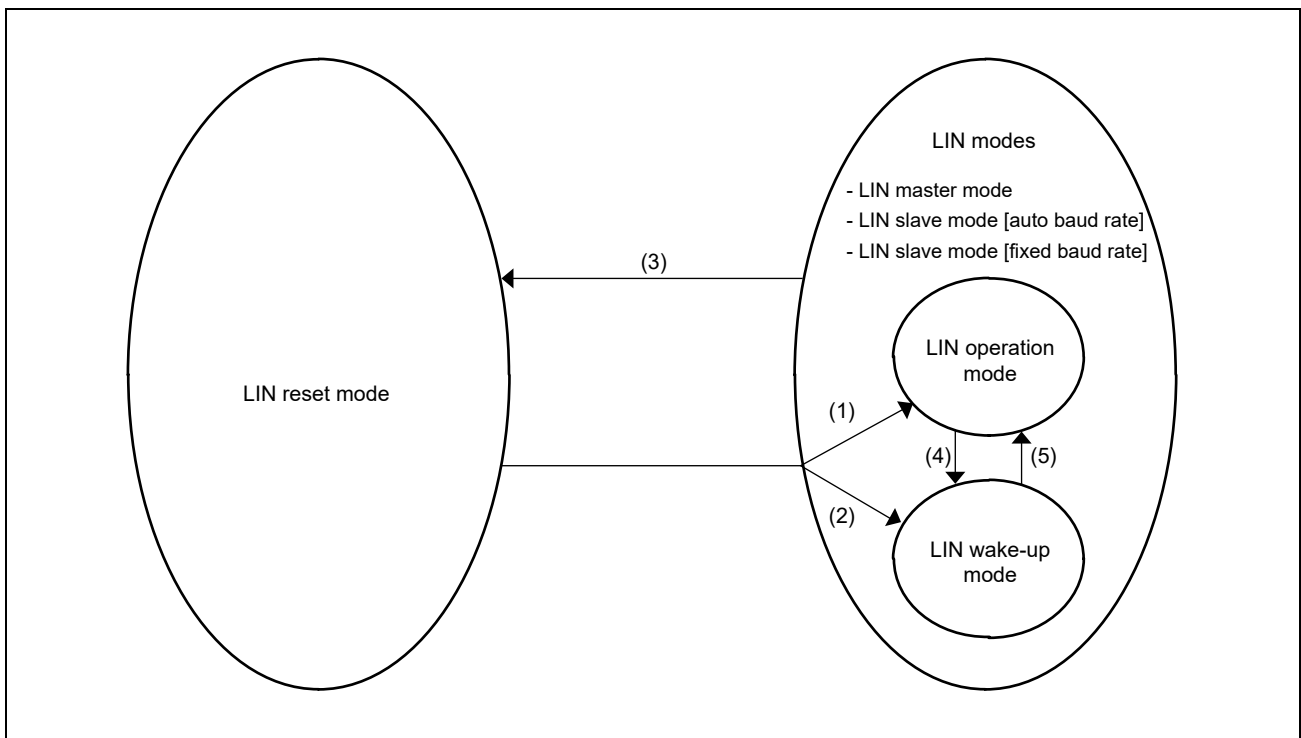


Figure 19.3 Transition of Operation Modes

Table 19.76 Transition Condition for Operation Mode

Operation Mode Transition		Transition Condition
(1)	LIN reset mode → LIN mode • LIN operation mode	LMD bits in RLN3nLMD register = 00 _B or 10 _B or 11 _B and OM1 and OM0 bits in RLN3nLCUC register = 11 _B
(2)	LIN reset mode → LIN mode • LIN wake-up mode	LMD bits in RLN3nLMD register = 00 _B or 10 _B or 11 _B and OM1 and OM0 bits in RLN3nLCUC register = 01 _B
(3)	LIN mode • LIN operation mode • LIN wake-up mode → LIN reset mode	OM0 bit in RLN3nLCUC register = 0 _B
(4)	LIN mode *1 • LIN operation mode → LIN mode • LIN wake-up mode	OM1 and OM0 bits in RLN3nLCUC register = 01 _B
(5)	LIN mode *1 • LIN wake-up mode → LIN mode • LIN operation mode	OM1 and OM0 bits in RLN3nLCUC register = 11 _B

Note 1. Transition between LIN operation mode and LIN wake-up mode cannot be made when communication is in progress (when the FTS bit in the RLN3nLTRC register is 1).

(1) LIN Operation Mode

In LIN operation mode, frame processing (header transmission, header reception, response transmission, response reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to 11_B changes the mode to LIN operation mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to 11_B. Communication settings should be performed after the OMM1 and OMM0 bits have become 11_B.

(2) LIN Wake-up Mode

In LIN wake-up mode, wake-up signal processing (wake-up transmission, wake-up reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to 01_B changes the mode to LIN wake-up mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to 01_B. Communication settings should be performed after the OMM1 and OMM0 bits have become 01_B.

19.5.3.1 LIN Master Mode

(1) Header Transmission

Figure 19.4 shows the operation of the LIN/UART interface (LIN master mode) in header transmission.

Table 19.77 describes the processing in header transmission.

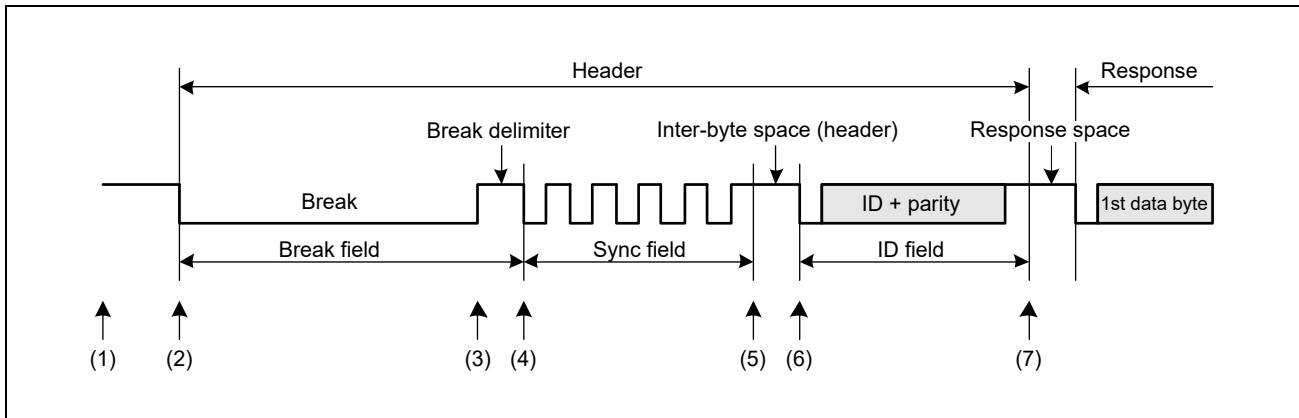


Figure 19.4 Operation in Header Transmission

Table 19.77 Processing in Header Transmission

Software Processing	LIN/UART Interface Processing
(1) <ul style="list-style-type: none"> • Sets a baud rate • Sets noise filter ON/OFF • Enables interrupt • Enables error detection • Sets frame configuration parameters • Changes the LIN/UART interface to the LIN master mode: LIN operation mode • Sets information on the frame to be transmitted (ID, parity, data length, response direction, checksum method, and transmission data) 	Waits for the setting of the FTS bit in the RLIN3nLTRC register by software (idle)
(2) Sets the FTS bit in the RLIN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started)	Transmits a break.
(3) Waits for an interrupt request	Transmits a break delimiter.
(4)	Transmits a sync field (55 _H).
(5)	Transmits an inter-byte space (header).
(6)	Transmits an ID field.
(7)	Sets a successful header transmission flag.

NOTE

For information about error detection conditions, see **Section 19.5.3.7, Error Status**.

(2) Response Transmission

Figure 19.5 shows the operation of the LIN/UART interface (LIN master mode) in response transmission. **Table 19.78** describes the processing in response transmission.

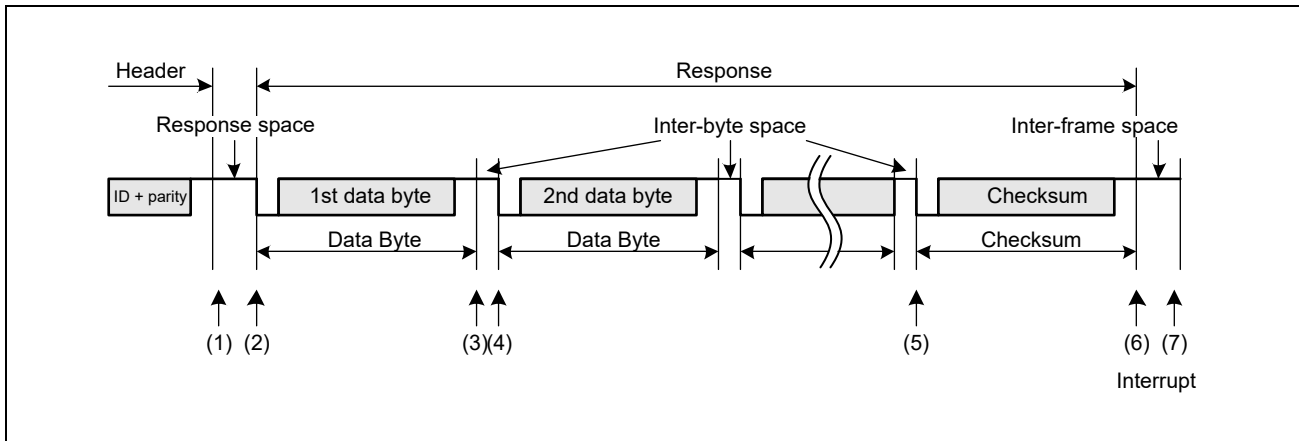


Figure 19.5 Operation in Response Transmission

Table 19.78 Processing in Response Transmission

Software Processing	LIN/UART Interface Processing
(1) [When in frame separate mode] <ul style="list-style-type: none"> • Sets the RTS bit in the RLN3nLTRC register to 1 (response transmission/reception started) [When not in frame separate mode] <ul style="list-style-type: none"> • Waits for an interrupt request 	[When in frame separate mode] <ul style="list-style-type: none"> • Waits for the setting of the RTS bit in the RLN3nLTRC register to 1 by software. • When the bit is set to 1, sends a response space. [When not in frame separate mode] <ul style="list-style-type: none"> • Sends a response space.
(2) Waits for an interrupt request	Transmits 1st data byte.
(3)	Transmits an inter-byte space.
(4)	<ul style="list-style-type: none"> • Transmits 2nd data byte. • Transmits an inter-byte space • Transmits 3rd data byte. • Transmits an inter-byte space (Repeats as many times as needed to reach the data length specified in bits RFDL[3:0] in the RLN3nLDFC register, and stops the transmission when the BER flag in the RLN3nLEST register is 1 (bit error detected). If an error occurs, the Checksum transmission in item (5) is not performed). : :
(5)	Transmits the checksum.
(6)	<ul style="list-style-type: none"> • Sets a successful frame/wake-up transmission flag. • Sets the FTS bit in the RLN3nLTRC register to 0 (frame transmission or wake-up transmission/reception stopped) [When in frame separate mode] <ul style="list-style-type: none"> • Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception is halted).
(7) <ul style="list-style-type: none"> • Processing after communication • Checks the RLN3nLST register, and clears flags. 	Idle

NOTE

For information about error detection conditions, see **Section 19.5.3.7, Error Status**.

(3) Response Reception

Figure 19.6 shows the operation of the LIN/UART interface (LIN master mode) in response reception. **Table 19.79** describes the processing in response reception.

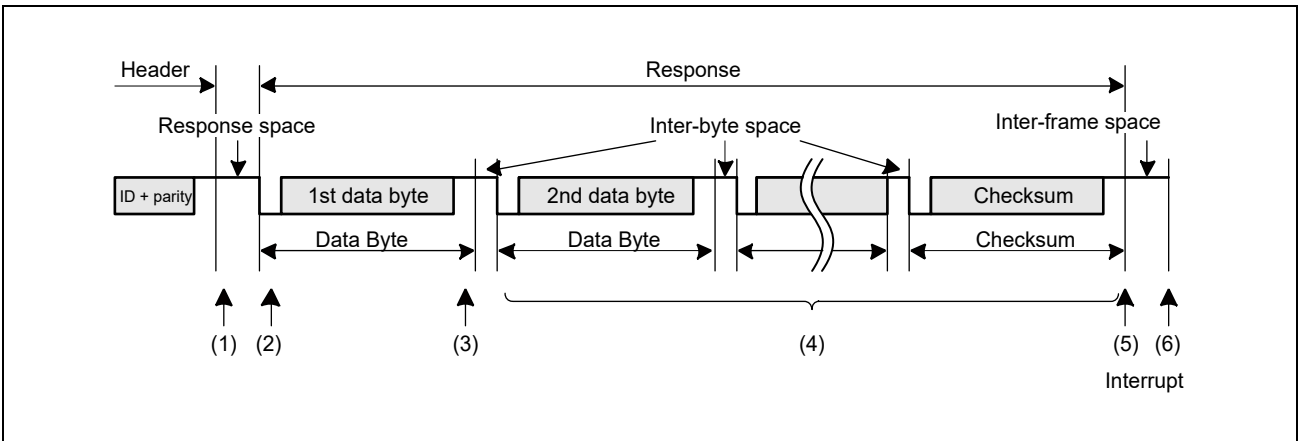


Figure 19.6 Operation in Response Reception

Table 19.79 Processing in Response Reception

Software Processing	LIN/UART Interface Processing
(1) <ul style="list-style-type: none"> When in frame separate mode Sets the RTS bit in the RLN3nLTRC register to 1 (response transmission/reception started) When in frame combined mode Waits for an interrupt request (no processing) 	<ul style="list-style-type: none"> When in frame separate mode Waits for the setting of the RTS bit in the RLN3nLTRC register to 1 by software. When the bit is set to 1, waits for detection of a start bit. When in frame combined mode Waits for detection of a start bit.
(2)	Receives 1st data byte when the start bit is detected.
(3)	Sets the successful 1st data byte reception flag.
(4)	<ul style="list-style-type: none"> Receives 2nd data byte when the start bit is detected. Receives 3rd data byte when the start bit is detected. <p>(Repeats the reception of data bytes as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register, and stops the reception when any bit in the RLN3nLEST register is 1 (bit error detected). If an error occurs, the checksum determination in item (5) is not performed).</p> <p style="text-align: center;">:</p> <p style="text-align: center;">:</p> <ul style="list-style-type: none"> Receives the checksum when the start bit is detected.
(5)	<ul style="list-style-type: none"> Determines the checksum. Sets the successful frame/wake-up reception flag. Sets the FTS bit in the RLN3nLTRC register to 0 (frame transmission or wake-up transmission/reception stopped).
(6) <ul style="list-style-type: none"> Processing after communication Reads the received data. Checks the RLN3nLST register, and clears flags. 	Idle

NOTE

For information about error detection conditions, see **Section 19.5.3.7, Error Status**.

19.5.3.2 LIN Slave Mode

(1) Header Reception

Figure 19.7 shows the operation of the LIN/UART interface (LIN slave mode) in header reception.

Table 19.80 describes the processing in header reception.

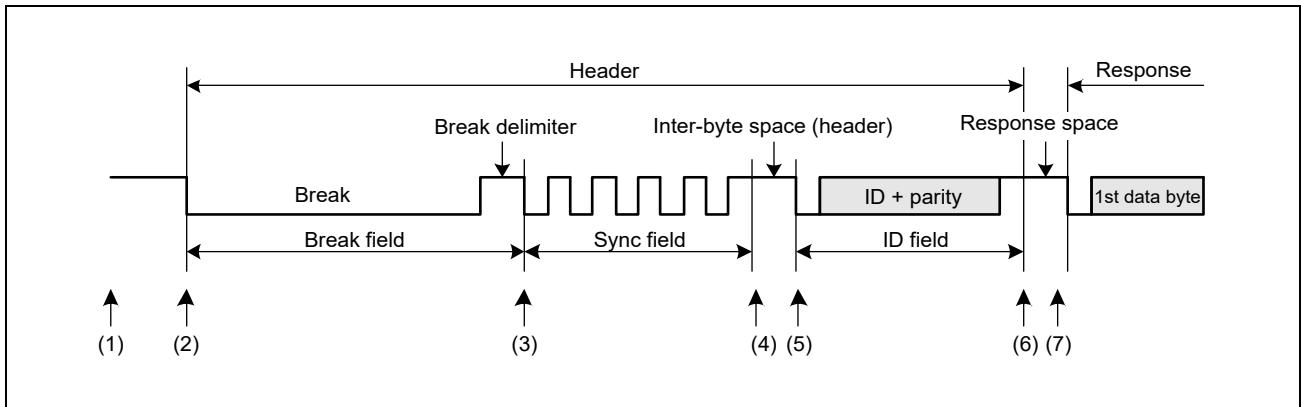


Figure 19.7 Operation in Header Reception

Table 19.80 Processing in Header Reception

Software Processing	LIN/UART Interface Processing
(1) <ul style="list-style-type: none"> • Sets a baud rate • Sets noise filter ON/OFF • Enables interrupt • Enables error detection • Sets frame configuration parameters • Changes the LIN/UART interface to the LIN slave mode: LIN operation mode • Sets the FTS bit in the RLIN3nLTRC register to 1 (header reception or wake-up transmission/reception started) 	Waits for the setting of the FTS bit in the RLIN3nLTRC register by software.
(2) Waits for an interrupt request.	Waits for detection of break field
(3)	Detects a break field. (LIN slave mode [fixed baud rate]). For details about the break field detection timing in the case of LIN slave mode [auto baud rate], see [Auto Baud Rate Correction Function].)
(4)	<ul style="list-style-type: none"> • Detects a sync field (55µs) • Set the baud rate generator (in the case of LIN slave mode [auto baud rate]) • Clears the no-response request bit (LNRR bit).
(5)	<ul style="list-style-type: none"> • Receives an ID field. • Checks an ID parity bit
(6)	Sets a header reception complete flag.
(7) <ul style="list-style-type: none"> • Checks the RLIN3nLST register, and clears flags. • Checks the RLIN3nLIDB register, and prepares a response. 	<ul style="list-style-type: none"> • Completes a header reception process. • Waits for a response request.

NOTE

The LIN/UART interface allows reception of break fields during frame transmission/reception. In that case, a framing error, bit error or other error may be detected at the stop bit position of the frame before the break field is received, and a status interrupt may occur as a result. However, reception of a new header (the following Sync field and ID field) continues regardless of whether an error occurred. For information about error detection conditions, see **Section 19.5.3.7, Error Status**.

[Auto Baud Rate Correction Function]

In LIN slave mode [auto baud rate], the system always measures the low-level widths that are received. If the first “Low level” width is 10 times (if the BLT bit of the RLN3nLBFC register is “0”) or 11 times (if the BLT bit of the RLN3nLBFC register is “1”) or greater than the bit width calculated from the average of the starting 2 bits (the period of the consecutive falling edges from the beginning of the sync field) of the sync field, the system concludes that the detection of break field was successful, the system verifies that the data in the sync field is 55_H. If the data in the sync field is indeed 55_H and the system judges that sync field reception was successful, the system automatically sets the baud rate correction result to the RLN3nLBRP01 register.

If data is received up to the ID field without error, a successful header reception interrupt is generated at the stop bit position.

On the other hand, if the data in the sync field is not 55_H and the system judges that sync field reception failed, the system sets the sync field error flag and an error interrupt is generated. In that case, baud rate correction is not performed and the LIN/UART interface waits for the detection of the next break field (low level).

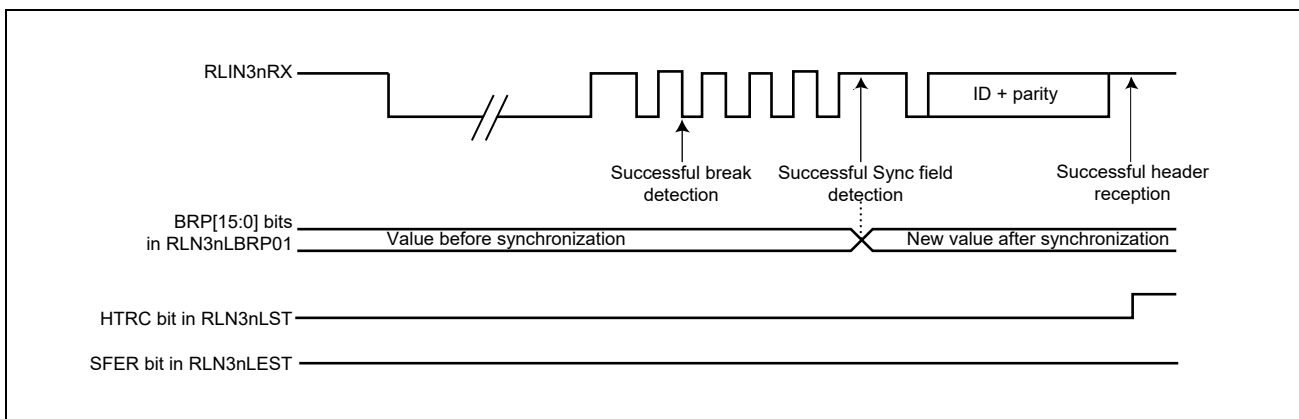


Figure 19.8 Header Reception in LIN Slave Mode [Auto Baud Rate] (in Normal Operation)

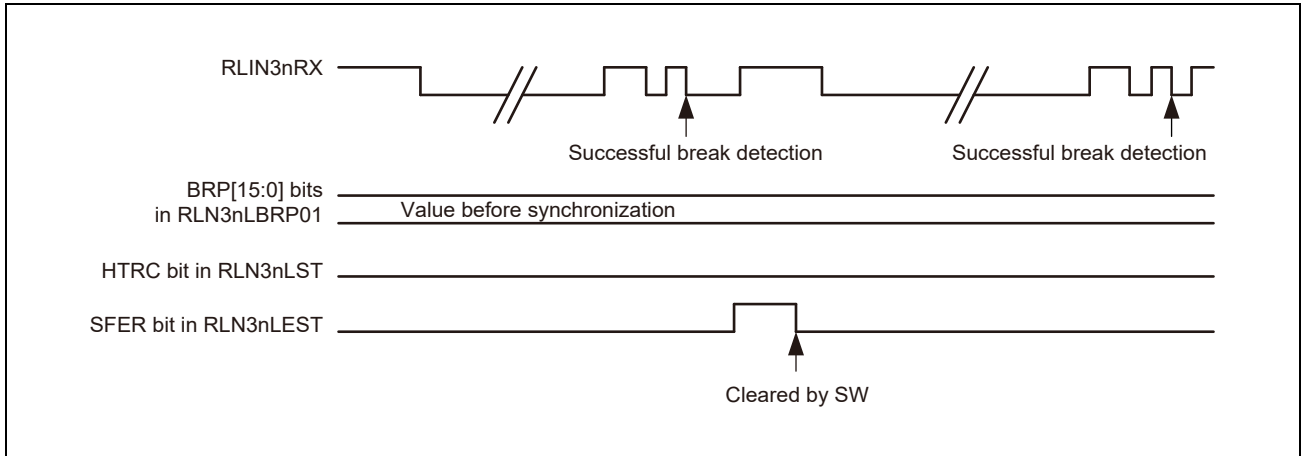


Figure 19.9 Header Reception in LIN Slave Mode [Auto Baud Rate] (Sync Field Error)

(2) Response Transmission

Figure 19.10 shows the operation of the LIN/UART interface (LIN slave mode) in response transmission. **Table 19.81** describes the processing in response transmission.

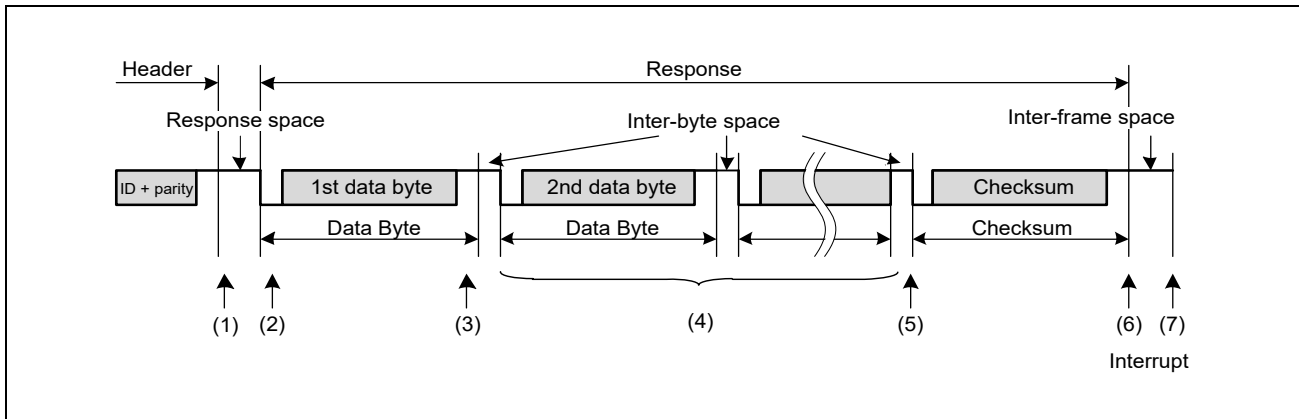


Figure 19.10 Operation in Response Transmission

Table 19.81 Processing in Response Transmission

Software Processing	LIN/UART Interface Processing
(1) <ul style="list-style-type: none"> • Sets the RLN3nLDFC register. • Sets the RLN3nLDBRb registers. (b = 1 to 8) • Sets the RTS bit in the RLN3nLTRC register to 1 (response transmission/reception started) 	<ul style="list-style-type: none"> • Waits for the setting of the RTS or LNRR bit of the RLN3nLTRC register by software • Transmits the response space after the RTS bit of the RLN3nLTRC register is set to 1
(2) Waits for an interrupt request.	Transmits 1st data byte.
(3)	Transmits the inter-byte space.
(4)	<ul style="list-style-type: none"> • Transmits 2nd data byte. • Transmits an inter-byte space • Transmits 3rd data byte. • Transmits an inter-byte space (Repeats as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register, and stops the transmission when the BER bit in the RLN3nLEST register is 1 (bit error detected). If an error occurs, the checksum transmission in item (5) is not performed).
(5)	Transmits the checksum.
(6)	<ul style="list-style-type: none"> • Sets the successful response/wake-up transmission flag. • Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception stopped). [In frame separate mode] <ul style="list-style-type: none"> • Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception stopped).
(7) <ul style="list-style-type: none"> • Processing after communication Checks the RLN3nLST register, and clears flags. 	<ul style="list-style-type: none"> • Completes the response transmission process. • Waits for a new break.

NOTE

- For information about error detection, see **Section 19.5.3.7, Error Status**.
 - The LIN/UART interface allows reception of break fields during frame transmission/reception. In that case, a framing error, bit error or other error may be detected at the stop bit position of the frame before the break field is received, and a status interrupt may occur as a result. However, reception of a new header (the following Sync field and ID field) continues regardless of whether an error occurred.
-

(3) Response Reception

Figure 19.11 shows the operation of the LIN/UART interface (LIN slave mode) in response reception. **Table 19.82** describes the processing in response reception.

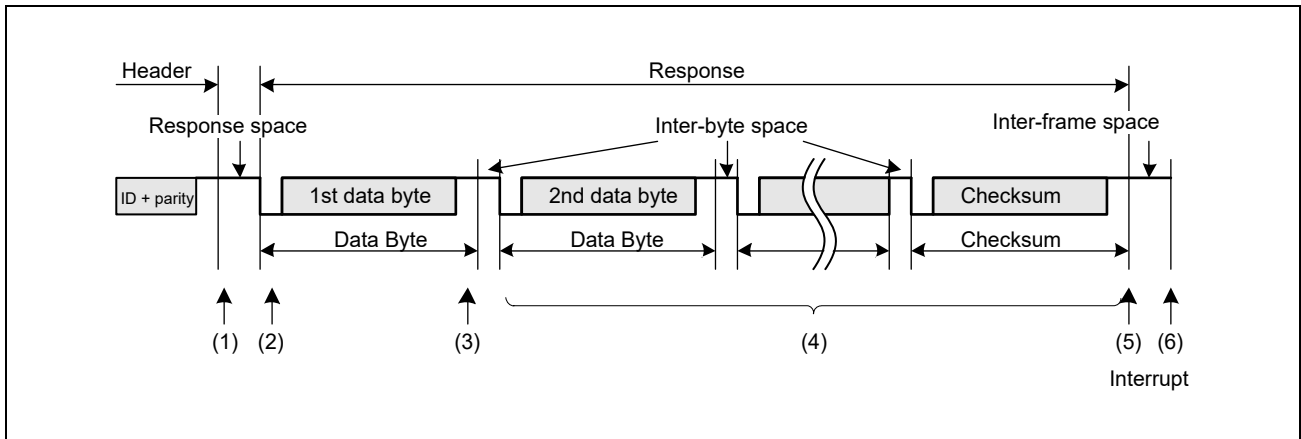


Figure 19.11 Operation in Response Reception

Table 19.82 Processing in Response Reception

Software Processing	LIN/UART Interface Processing
(1) <ul style="list-style-type: none"> • Sets the RLN3nLDFC register. • Sets the response transmission/reception start bit (RTS bit) to 1. 	<ul style="list-style-type: none"> • Waits for the setting by software of the response transmission/reception start bit (RTS bit) or the no-response request bit (LNRR bit). • Waits for detection of the start bit.
(2) Waits for an interrupt request.	Receives 1st data byte when the start bit is detected.
(3)	Sets the successful data 1 reception flag.
(4)	<ul style="list-style-type: none"> • Receives 2nd data byte when the start bit is detected. • Receives 3rd data byte when the start bit is detected. (Repeats as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register, and stops the transmission when any bit in the RLN3nLEST register is 1 (bit error detected). If an error occurs, the checksum determination in item (5) is not performed). : : <ul style="list-style-type: none"> • Receives the checksum when the start bit is detected.
(5)	<ul style="list-style-type: none"> • Determines the checksum. • Sets a successful response/wake-up reception flag or error flag. • Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception stopped).
(6) <ul style="list-style-type: none"> • Processing after communication • Reads the received data. • Checks the RLN3nLST register, and clears flags. 	<ul style="list-style-type: none"> • Completes the response process. • Waits for a new break.

NOTE

- For information about error detection conditions, see **Section 19.5.3.7, Error Status**.
 - The LIN/UART interface allows reception of break fields during frame transmission/reception. In that case, a framing error, bit error or other error may be detected at the stop bit position of the frame before the break field is received, and a status interrupt may occur as a result. However, reception of a new header (the following Sync field and ID field) continues regardless of whether an error occurred.
-

(4) No-response Request

Figure 19.12 shows the operation of the LIN/UART interface (LIN slave mode) when no response is requested. **Table 19.83** shows the processing that occurs when no response is requested.

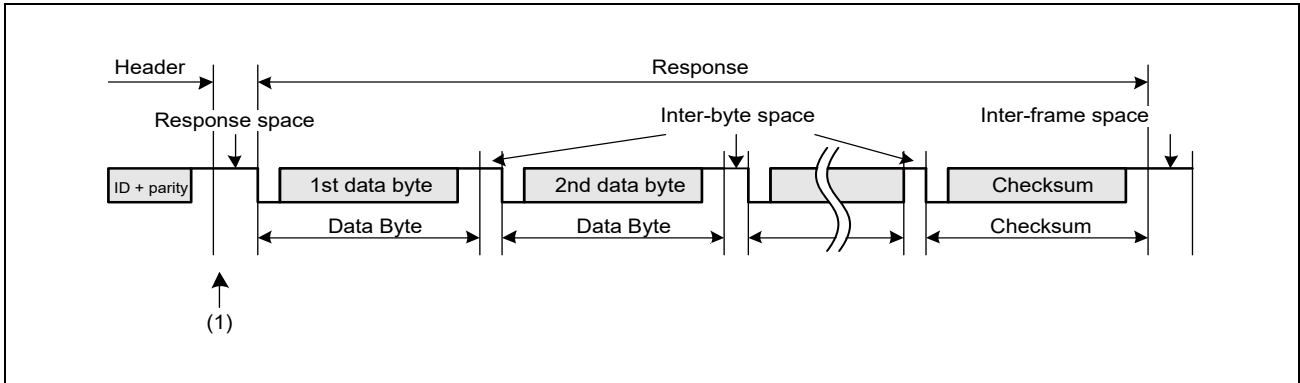


Figure 19.12 Operation when No Response is Requested

Table 19.83 Processing when No Response is Requested

Software Processing	LIN/UART Interface Processing
(1) <ul style="list-style-type: none"> • Sets the no-response request bit (LNRR bit) to 1. 	<ul style="list-style-type: none"> • Waits for setting of the no-response request bit (LNRR bit) by software • Completes the frame reception process • Waits for a new break

19.5.3.3 Data Transmission/Reception

(1) Data Transmission

One bit of data is transmitted per 1 Tbit.

The data that is transmitted returns to the reception data input pin via the LIN transceiver. The received data and the transmitted data are compared bit by bit, and the results are stored in the BER flag in the RLN3nLEST register (see **Section 19.5.3.7, Error Status.**).

In LIN master mode and LIN slave mode [fixed baud rate], 1 Tbit is generated to be $16f_{LIN}$, and thus the sampling point for received data are at the 13th clock cycle (81.25% position).

In LIN slave mode [auto baud rate], if 1 Tbit is generated to be $4f_{LIN}$, the sampling point for received data is at the third clock cycle (75% position). If 1 Tbit is generated to be $8f_{LIN}$, the sampling point for received data is at the 7th clock cycle (87.5% position).

Figure 19.13 shows an example of data transmission timing.

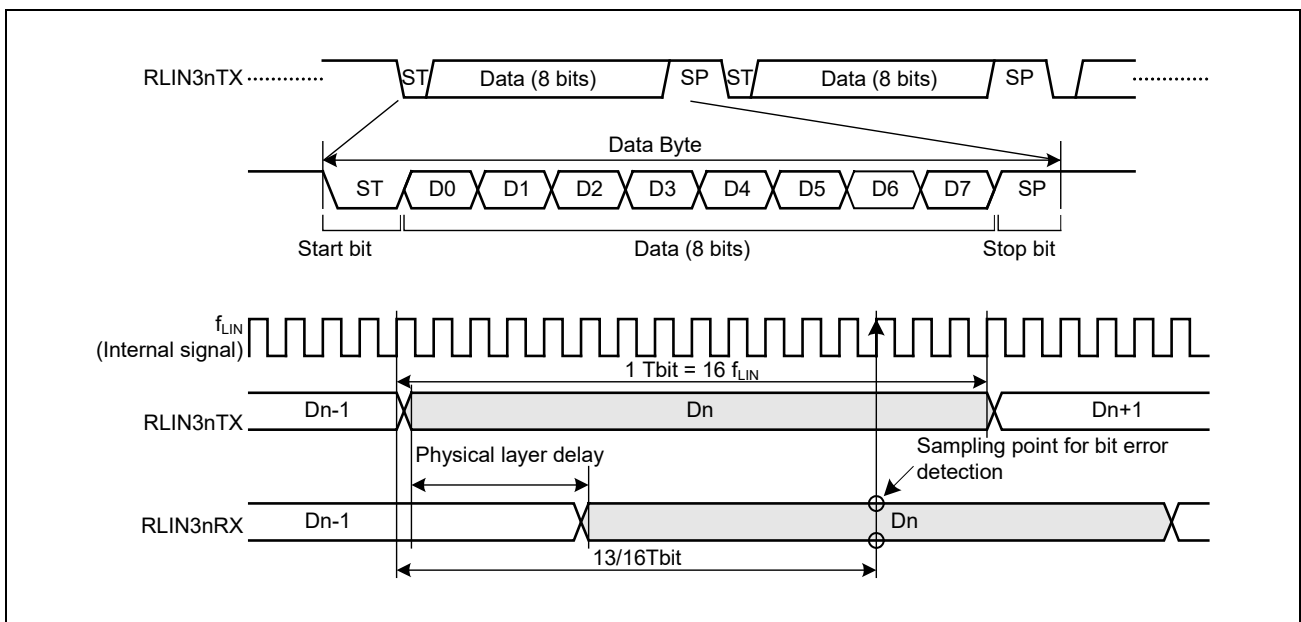


Figure 19.13 Example of Data Transmission Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate])

(2) Data Reception

Data reception is performed by using the synchronized RLIN3nRX signal (an internal signal) that is the input from the RLIN3nRX pin synchronized with prescaler clock.

The data byte is synchronized at the falling edge of the start bit for the synchronized RLIN3nRX signal. After the falling edge is detected, sampling is performed again 0.5 Tbit later, and the falling edge is recognized as a start bit if the synchronized RLIN3nRX signal is low level. The falling edge is not recognized as a start bit if the RLIN3nRX signal after the reset is de-asserted is fixed to low level or if a high level is detected on re-sampling.

After the start bit is detected, the system samples 1 bit per Tbit.

The LIN/UART interface has a noise filter function for reception data. If the LRDNFS bit in the RLIN3nLMD register is 0, the LIN/UART interface uses a noise filter, and the value determined by a 3-sampling majority rule on prescaler clocks is used as the sampling value. If the LRDNFS bit in the RLIN3nLMD register is 1, the LIN/UART interface does not use a noise filter, and the value of the synchronized RLIN3nRX value at the sampling position is used as the sampling value.

Figure 19.14 shows an example of data reception timing.

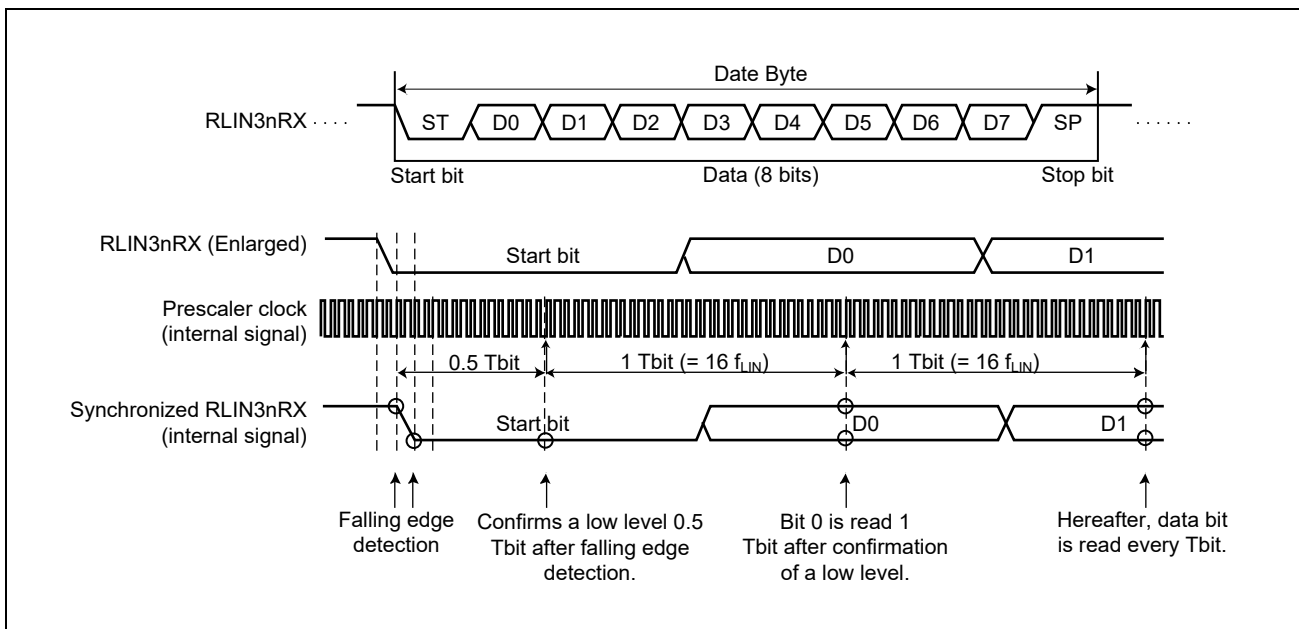


Figure 19.14 Example of Data Reception Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate])

19.5.3.4 Transmission/Reception Data Buffering

This section explains the buffer processing that takes place when the LIN/UART interface sends or receives data continuously.

(1) Transmission of LIN Frames

For an 8-byte transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR8 are sequentially transmitted to data areas 1 to 8 of the LIN frame. In the case of a 4-bytes transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR4 are transmitted to data areas 1 to 4 of the LIN frame, but the contents of registers RLN3nLDBR5 to RLN3nLDBR8 are not transmitted. The transmitted checksum data is stored in the RLN3nLCBR register.

Figure 19.15 shows the LIN transmission processing and the required buffers.

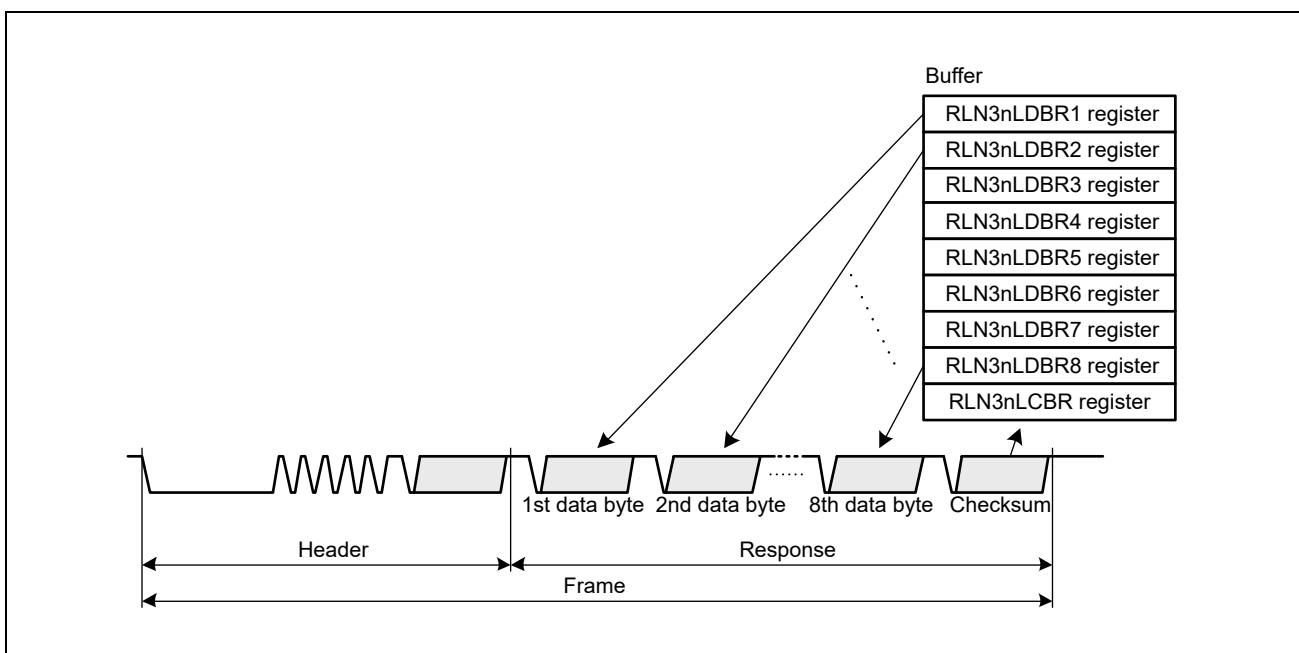


Figure 19.15 LIN Transmission Processing and Required Buffer

[Frame separate mode]

Setting the FSM bit in the RLN3nLDFC register to 1 turns on the frame separate mode.

In frame separate mode, a header and a response are transmitted when prompted by separate transmission start requests.

When the transmission of a header is finished, the HTRC flag in the RLN3nLST register is set to 1 (successful header transmission).

Use frame separate mode when transmitting or receiving response data of 9 bytes or more in LIN master mode.

(2) Reception of LIN Frames

For an 8-byte reception, the contents of data areas 1 to 8 of the LIN frame are stored in registers RLN3nLDBR1 to RLN3nLDBR8, respectively, upon reception of a stop bit. In the case of a 4-byte reception, the contents of data areas 1 to 4 of the LIN frame are stored in registers RLN3nLDBR1 to RLN3nLDBR4, respectively; no data is stored in registers RLN3nLDBR5 to RLN3nLDBR8. Also, the received checksum data is stored in the RLN3nLCBR register.

Figure 19.16 shows the LIN reception processing and the required buffers.

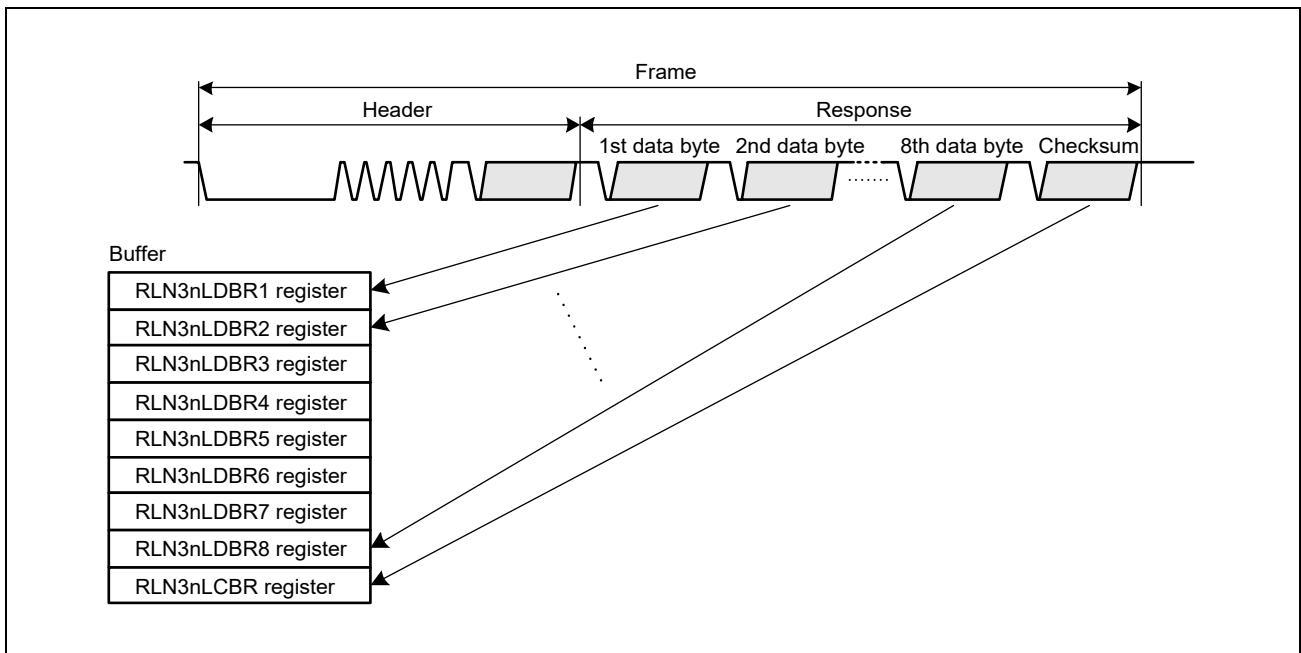


Figure 19.16 LIN Reception Processing and Required Buffer

[Reception of 1st Data Byte]

When the reception of the first byte of data is finished, the DIRC flag in the RLN3nLST register is set to 1 (successful data 1 reception).

(3) Multi-Byte Response Transmission/Reception Function

Normally in LIN communications, a response is 9 bytes or less including a checksum field; however, responses of 10 bytes or more can also be transmitted and received.

In such a case, the bit error, framing error, response preparation error detection, and auto checksum functions are enabled.

If the data length is longer than 8 bytes, the LSS bit in RLN3nLDFC register should be set to 1 (indicating that the next data group to be transmitted or received is not the last data group) in the first data group (variable from 0 to 8 bytes) before transmitting or receiving the data group. After the transmission or reception, the user should determine whether the next data group is the last data group.

If it is the last data group, the LSS bit in the RLN3nLDFC register should be set to 0 (indicating that the next data group to be transmitted or received is the last data group), and a checksum should be appended to the last data group.

By changing the RFDL bit in RLN3nLDFC register settings when the RTS bit in RLN3nLTRC register is 0, the user can change the data length for each data group.

When performing multi-byte response transmission/reception in LIN master mode, set the FSM bit in RLN3nLDFC register to 1 (frame separate mode).

NOTE

In LIN slave mode, the LIN/UART interface can detect a new break field during the transmission or reception of a response.

19.5.3.5 Wake-up Transmission/Reception

The wake-up transmission/reception can be used in LIN wake-up mode.

(1) Wake-up Transmission

In LIN wake-up mode, setting the RFT bit in the RLN3nLDFC register to 1 (in LIN master mode: response transmission), or setting the RCDS bit in the RLN3nLDFC register to 1 (in LIN slave mode: response transmission), and then the FTS bit in the RLN3nLTRC register to 1 (frame transmission, header reception or wake-up transmission/reception started) causes a wake-up signal to be output from the output pin. The low width of the wake-up signal should be set using the WUTL[3:0] bits in the RLN3nLWUP register.

However, if the LWBR0 bit of the RLN3nLWBR register in LIN master mode is 1 (LIN2.x), the LIN system clock (f_{LIN}) becomes low level width at f_a regardless of the setting of the LCKS bit of the RLN3nLMD register. By setting the WUTL[3:0] bits of the RLN3nLWUP register to 0100_B (5 Tbits), 260 μs low width can be output in LIN wake-up mode regardless of the setting of the LCKS bit of the RLN3nLMD register.

If a wake-up low-level width is output without any bit error, the FTC flag in the RLN3nLST register is set to 1 (successful frame response or wake-up transmission); when the FTCIE bit in the RLN3nLIE register is 1 (successful frame response/wakeup transmission interrupt enabled), an interrupt request for RLIN3n transmission is generated.

If RLN3nLEDE.BERE is set and a bit error is detected, wake-up transmission is canceled and the BER flag in the RLN3nLEST register is set to 1 (bit error detection).

When RLN3nLEDE.PBERE is set in LIN master mode, set RLN3nLEST.PBER flag to 1 (physical bus error detection) at the same time of a bit error.

Figure 19.17 shows the wake-up transmission timing.

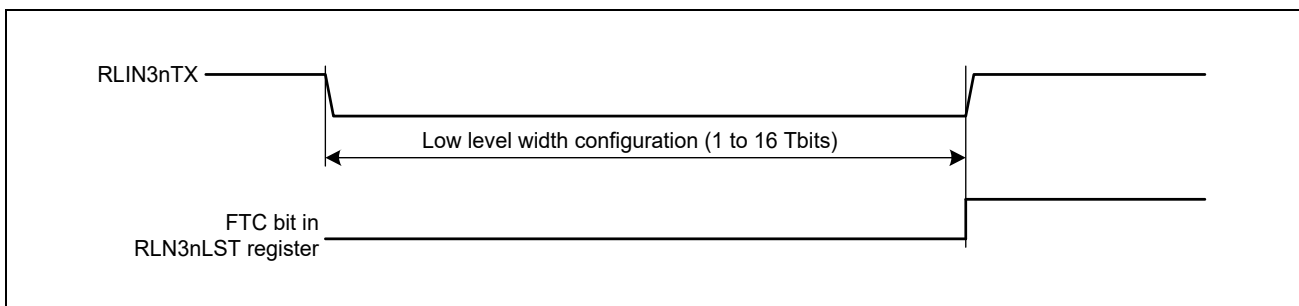


Figure 19.17 Wake-up Transmission Timing

(2) Wake-up Reception

To detect a wake-up signal, use the input signal low-level width count function. The input signal low level width count function measures the low level width of the input signal to the RLIN3nRX pin, using the same sampling point as data reception. This allows the 2.5-Tbit or longer low-level width of the input signal of fLIN to be measured.

In LIN master mode, by setting the LWBR0 bit in the RLN3nLWBR register, operation can be executed without changing the baud rate generator setting when switching between LIN operation mode and LIN wake-up mode.

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 sets the LIN system clock (fLIN) to fa regardless of the setting of the LCKS bit in the RLN3nLMD register. (The LCKS bit is not changed). By setting the baud rate to 19200bps while fa is selected, an input signal with a low-level width of 130 us or longer can be measured in LIN wake-up mode regardless of the setting of the LCKS bit in the RLN3nLMD register.

When using the wake-up reception function, in LIN wake-up mode set the RFT bit in the RLN3nLDFC register to 0 (LIN master mode: response reception), or the RCDS bit in the RLN3nLDFC register to 0 (in LIN slave mode: response reception), and then set the FTS bit in the RLN3nLTRC register to 1 (frame transmission (header reception) or wake-up transmission/reception started).

When the low level width to be measured is reached, the FRC flag in the RLN3nLST register is set to 1 (successful frame response/wake-up reception). If the FRCIE bit in the RLN3nLIE register is 1 (successful frame response or wake-up reception interrupt enabled), an interrupt request for successful RLIN3n reception is generated.

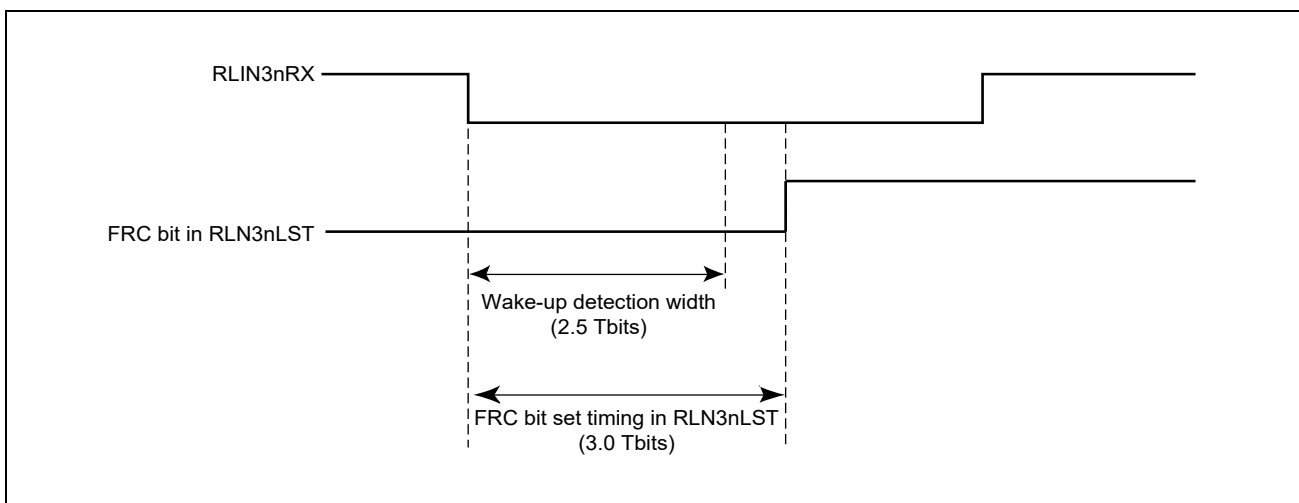


Figure 19.18 Input Signal Low Level Count Function

(3) Wakeup Collision

If the master node and the slave node transmit wakeup signals simultaneously, a collision will occur on the LIN bus, though a collision of wakeup signals is not detected by the LIN/UART interface.

19.5.3.6 Status

In LIN mode operation, the LIN/UART interface can detect seven types of statuses.

The four statuses, successful frame/wake-up transmission, successful frame/wake-up reception, error detection, and successful header transmission/reception, can generate interrupt requests.

Table 19.84 shows the types of statuses available in LIN master mode. **Table 19.85** lists the types of statuses available in LIN slave mode [auto baud rate] and in LIN slave mode [fixed baud rate].

Table 19.84 Types of Statuses in LIN Master Mode

Status	Status Set Condition	Status Clear Condition	Operation Mode Capable of Status Detection	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN-reset-mode, if actually the LIN/UART interface is cleared from LIN reset mode.	After the OM0 bit in the RLN3nLCUC register is set to LIN reset mode, if actually the LIN/UART interface enters LIN reset mode.	All modes	OMM0 bit in RLN3nLMST register	—
Operation mode	After the OM1 bit in the RLN3nLCUC register is set to LIN operation mode, if actually the LIN/UART interface enters LIN operation mode.	After the OM1 bit in the RLN3nLCUC register is set to LIN wake-up mode, if actually the LIN/UART interface enters LIN wake-up mode.	<ul style="list-style-type: none"> • LIN operation mode • LIN wake-up mode 	OMM1 bit in RLN3nLMST register	—
Frame/wake-up transmission end	When a frame (header transmission + response transmission), a wake-up signal, or a data group is transmitted successfully.	<ul style="list-style-type: none"> • When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode 	<ul style="list-style-type: none"> • LIN operation mode • LIN wake-up mode 	FTC flag in RLN3nLST register	✓
Frame/wake-up reception end	When a frame (header transmission + response reception), a wake-up signal, or a data group is received successfully.	<ul style="list-style-type: none"> • When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode 	<ul style="list-style-type: none"> • LIN operation mode • LIN wake-up mode 	FRC flag in RLN3nLST register	✓
Error detection	If any of the RPER flag, CSER flag, FER flag, FTER flag, PBER flag, and BER flag in the RLN3nLEST register is set to 1 (error detected).	<ul style="list-style-type: none"> • When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software*1 • After transition to LIN reset mode 	<ul style="list-style-type: none"> • LIN operation mode • LIN wake-up mode 	ERR flag in RLN3nLST register	✓
1st data byte reception end	The RFT bit in the RLN3nLDFC register is 0 (reception) and the first byte of the response field or the first byte of each data group is received successfully.*2	<ul style="list-style-type: none"> • When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode 	LIN operation mode	D1RC flag in RLN3nLST register	—
Header transmission end	When a header field is transmitted successfully.	<ul style="list-style-type: none"> • When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode 	LIN operation mode	HTRC flag in RLN3nLST register	✓

Note 1. In LIN wake-up mode or LIN operation mode, the ERR flag in the RLN3nLST register is cleared to 0 by writing 0 to the RPER flag, CSER flag, FER flag, FTER flag, PBER flag, or BER flag in the RLN3nLEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN3nLDFC register are 0000₈ (0-byte + checksum).

Table 19.85 Types of Statuses in LIN Slave Mode

Status	Status Set Condition	Status Clear Condition	Operation Mode Capable of Status Detection	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN-reset-mode, if actually the LIN/UART interface is cleared from LIN reset mode.	After the OM0 bit of the RLN3nLCUC register is set to LIN reset mode, if actually the LIN/UART interface enters LIN reset mode.	All modes	OMM0 bit in RLN3nLMST register	—
Operation mode	After the OM1 bit in the RLN3nLCUC register is set to LIN operation mode, if actually the LIN/UART interface enters LIN operation mode.	<ul style="list-style-type: none"> After the OM1 bit in the RLN3nLCUC register is set to LIN wake-up mode, if actually the LIN/UART interface enters LIN wake-up mode. 	<ul style="list-style-type: none"> LIN operation mode LIN wake-up mode 	OMM1 bit in RLN3nLMST register	—
Frame/wake-up transmission end	When a response field, a wake-up signal, or a data group is transmitted successfully.	<ul style="list-style-type: none"> When cleared by software Transition to LIN reset mode 	<ul style="list-style-type: none"> LIN operation mode LIN wake-up mode 	FTC flag in RLN3nLST register	✓
Frame/wake-up reception end	When a response field, a wake-up signal, or a data group is received successfully.	<ul style="list-style-type: none"> When cleared by software Transition to LIN reset mode 	<ul style="list-style-type: none"> LIN operation mode LIN wake-up mode 	FRC flag in RLN3nLST register	✓
Error detection	If any of the RPER flag, IPER flag, CSER flag, SFER flag, FER flag, TER flag, and BER flag in the RLN3nLEST register is set to 1 (error detected).	<ul style="list-style-type: none"> When cleared by software*1 Transition to LIN reset mode 	<ul style="list-style-type: none"> LIN operation mode LIN wake-up mode 	ERR flag in RLN3nLST register	✓
1st data byte reception end	The RCDS bit in the RLN3nLDFC register is 0 (reception) and the first byte of the response field or the first byte for each data group is received successfully.*2	<ul style="list-style-type: none"> When cleared by software Transition to LIN reset mode 	LIN operation mode	D1RC flag in RLN3nLST register	—
Header reception end	When a header field is received successfully.	<ul style="list-style-type: none"> When cleared by software Transition to LIN reset mode 	LIN operation mode	HTRC flag in RLN3nLST register	✓

Note 1. In LIN wake-up mode or LIN operation mode, the ERR flag in the RLN3nLST register is cleared to 0 by writing 0 to the RPER flag, IPER flag, CSER flag, SFER flag, FER flag, TER flag, or BER flag in the RLN3nLEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN3nLDFC register are 0000_B (0-byte + checksum).

19.5.3.7 Error Status

(1) LIN Master Mode

(a) Types of Error Statuses

The LIN/UART interface can detect six types of error statuses in LIN master mode. The condition of these error statuses can be verified by checking the corresponding bits in the RLIN3nLEST register.

All error statuses are interrupt factors.

Table 19.86 shows the types of error statuses.

Table 19.86 Types of Error Statuses in LIN Master Mode

Status	Error Detection Condition	Operation Mode Capable of Error Detection	Communication	Enable/Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data on the LIN bus monitored by the receive pin do not match <small>*1,*2</small>	<ul style="list-style-type: none"> LIN operation mode LIN wake-up mode 	Cancel	Enabled	BER flag in RLIN3nLEST register
Physical bus error	<ul style="list-style-type: none"> LIN bus detected a high level when sending a break LIN bus detected a low level when sending a break delimiter LIN bus detected a high level when sending a wake-up 	<ul style="list-style-type: none"> LIN operation mode LIN wake-up mode 	Cancel	Enabled	PBER flag in RLIN3nLEST register
Timeout error	A frame or response transmission/reception is not completed within a given time ^{*3}	LIN operation mode	Cancel	Enabled	FTER flag in RLIN3nLEST register
Framing error	In response field reception, a stop bit of each data byte is low level	LIN operation mode	Cancel	Enabled	FER flag in RLIN3nLEST register
Checksum error	In response field reception, the result of checksum indicates an error	LIN operation mode	—	Disabled	CSEF flag in RLIN3nLEST register
Response preparation error	<p>One of the following conditions occurs in frame separate mode during a multi-byte response reception:</p> <ul style="list-style-type: none"> After header transmission is complete, the first byte of receive data is received before a response transmission/reception request is set. After the previous data group reception is complete, the first byte of receive data is received before a transmission/reception request for the next data group is set. 	LIN operation mode	Cancel	Disabled	RPER flag in RLIN3nLEST register

Note 1. If a bit error is detected, processing is stopped after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is suspended immediately after the bit which had the error is sent. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is canceled after the error-causing bit is sent.

Note 2. In a multi-byte response transmission, bit errors are also detected between data groups.

Note 3. The timeout time depends on the response field data length (the RFDL [3:0] bits in the RLIN3nLDFC register) and the checksum selection (the CSM bit in the RLIN3nLDFC register), and can be calculated from the following formula. When the FSM bit in the RLIN3nLDFC registers is 1 (frame separation mode), the timeout time is that for eight bytes until the RTS bit of the RLIN3nLTRC register is set. Once the RTS bit is set, the timeout time is re-set to the time based on the response field data length (the RFDL[3:0] bits in the RLIN3nLDFC register).

[Frame timeout]

When classic is selected (when the CSM bit in RLIN3nLDFC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]

When enhanced is selected (when the CSM bit in RLIN3nLDFC is 1): Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]

The aforementioned timeout time is a time longer than the TFRAME_MAX of LIN Specification Package Revision 1.3 when classic is selected, or the TFRAME_MAX of LIN Specification Package Revision 2.x when enhanced is selected.

[Response timeout]

$$\text{Timeout time} = (\text{number of data bytes} + 1) \times 14 \text{ [Tbit]}$$

When an error is detected, the timeout error detection function stops.

The error status is cleared when the next communication is started (when the FTS bit in the RLIN3nLTRC register is set), by software, or at a transition to LIN reset mode.

(b) Target Time Area for LIN Error Detection

Figure 19.19 shows the time domain in which the LIN/UART interface in LIN master mode performs monitoring for error detection.

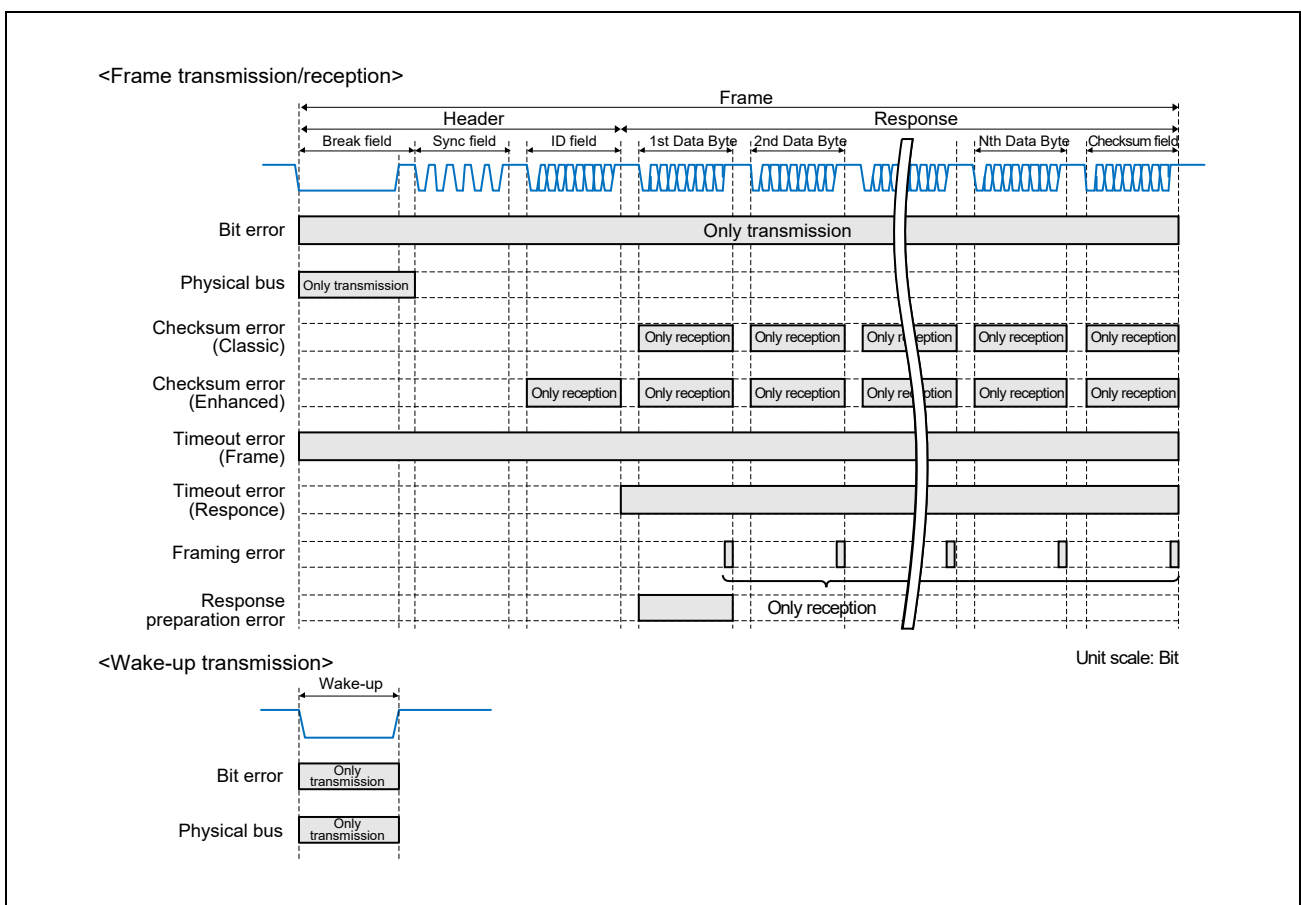


Figure 19.19 Target Time Area for LIN Error Detection (LIN Master Mode)

(2) LIN Slave Mode

(a) Types of Error Statuses

The LIN/UART interface can detect seven types of error statuses in LIN slave mode [auto baud rate] or in LIN slave mode [fixed baud rate]. These error statuses can be verified by checking the corresponding bits in the RLN3nLEST register.

Table 19.87 shows the types of error statuses.

Table 19.87 Types of Error Statuses in LIN Slave Mode

Status	Error Detection Condition	Operation Mode Capable of Error Detection	Communication	Enable/Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data on the LIN bus monitored by the receive pin do not match ^{*1*2}	<ul style="list-style-type: none"> • LIN operation mode • LIN wake-up mode 	Cancel	Enabled	BER flag in RLN3nLEST register
Timeout error	A frame or response transmission/reception is not completed within a given time ^{*3}	LIN operation mode	Cancel	Enabled	TER flag in RLN3nLEST register
Framing error	In frame reception, a stop bit of each data byte is low level	LIN operation mode	Cancel	Enabled	FER flag in RLN3nLEST register
Sync field error	If the width of the break low level is greater than the width set by the LBLT bit in the RLN3nLDFC register and the sync field is not 55 _H	LIN operation mode	Cancel	Enabled ^{*4}	SFER flag in RLN3nLEST register
Checksum error	In response field reception, the result of checksum indicates an error	LIN operation mode	— ^{*5}	Disabled	CSEF flag in RLN3nLEST register
ID parity error	If the received ID parity bit does not match the value that is automatically calculated by the LIN/UART interface	LIN operation mode	Cancel	Enabled	IPER flag in RLN3nLEST register
Response preparation error	<ul style="list-style-type: none"> • After the reception of a header, response preparation is not completed in time before the first byte of reception data is received • In multi-byte response reception, the reception preparation for the next data group is not completed in time before the first byte of the next data group reception data is received. 	LIN operation mode	Cancel	Disabled	RPER flag in RLN3nLEST register

Note 1. If a bit error is detected, processing is stopped after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is suspended immediately after the bit which had the error is sent. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is canceled after the error-causing bit is sent.

Note 2. In a multi-byte response transmission, bit errors are also detected between data groups.

Note 3. The timeout time depends on the response field data length (the RFDL [3:0] bits in the RLN3nLDFC register) and the checksum selection (the LCS bit in the RLN3nLDFC register), and this can be calculated according to the following formula. The timeout period until the RTS or LNRR bit of the RLN3nLTRC register is set is 8 data bytes. When the RTS bit is set, the timeout time is reset to the time based on the response field data length (RFDL[3:0] bits of the RLN3nLDFC register). When the LNRR bit is set, the timeout function stops.

[Frame timeout]

When classic is selected (when the LCS bit in RLN3nLDFC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]

When enhance is selected (when the LCS bit in RLN3nLDFC is 1): Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]

The aforementioned timeout time is a time longer than the TFRAME_MAX of LIN Specification Package Revision 1.3 when classic is selected, or the TFRAME_MAX of LIN Specification Package Revision 2.x when enhanced is selected.

[Response timeout]

Timeout time = (number of data bytes + 1) × 14 [Tbit]

When an error is detected, timeout error detection function stops.

Note 4. Only reflection of the result to the SFER flag can be enabled/disabled. Error detection cannot be enabled/disabled.

Note 5. Checksum judgment is performed upon completion of response frame reception. In case of an error, the successful reception flag is not set to 1.

The error status is cleared by software or at a transition to LIN reset mode.

(b) Target Time Area for LIN Error Detection

Figure 19.20 shows the time domain in which the LIN/UART interface in slave mode performs monitoring for error detection.

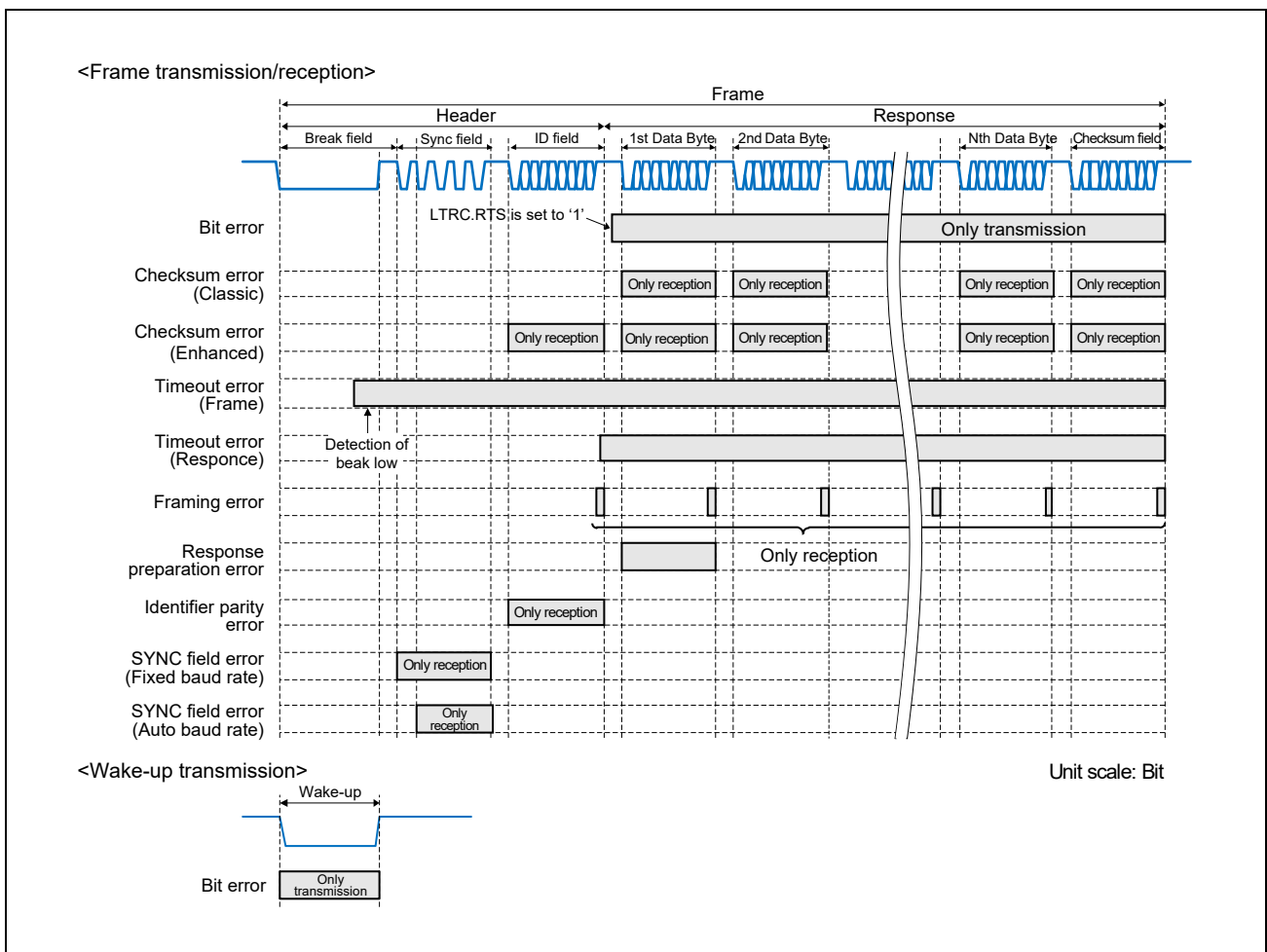


Figure 19.20 Target Time Area for LIN Error Detection (LIN Slave Mode)

19.5.4 UART Mode

In LIN reset mode, setting the LMD bits in the RLN3nLMD register to 01_B (UART mode) and the OM0 bit in the RLN3nLCUC register to 1 changes the mode to UART mode, turning the OMM0 bit in the RLN3nLMST register to 1.

19.5.4.1 Transmission

Figure 19.21 shows LIN/UART interface (in UART mode) transmission operations;

Table 19.88 shows LIN/UART interface (in UART mode) transmission processing.

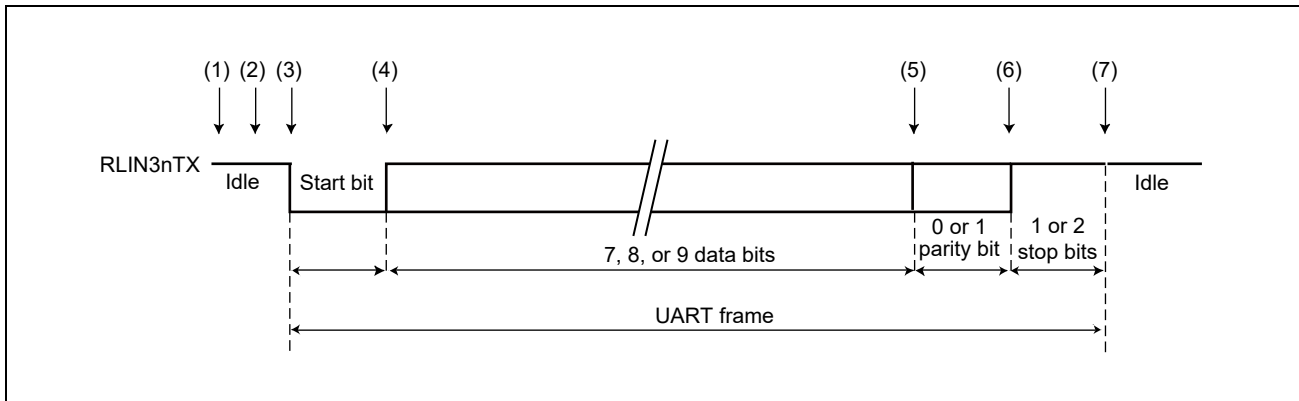


Figure 19.21 LIN/UART Interface (in UART Mode) Transmission Operation

Table 19.88 LIN/UART Interface (UART Mode) Transmission Processing (1/2)

Software Processing	LIN/UART Interface Processing
(1) <ul style="list-style-type: none"> • Sets a baud rate. • Sets noise filter ON/OFF. • Enables error detection • Sets data format. • Sets an interrupt generation timing. • Clears the LIN/UART interface from LIN reset mode. • Sets the transmit enable bit (UTOE bit) to 1. 	<ul style="list-style-type: none"> • Waits for a transmission trigger (RLN3nLUTDR register) by software.
(2) <ul style="list-style-type: none"> • Sets the transmit data to the UART transmit data register (RLN3nLUTDR) or UART wait transmit data register (RLN3nLUWTDNR). 	<ul style="list-style-type: none"> • Sets the transmit status flag.
(3) <ul style="list-style-type: none"> • Waits an interrupt request. 	<ul style="list-style-type: none"> • Transmits a start bit (for switching between transmission and reception in half duplex communication, transmits a start bit after receiving 1 stop bit. For details about this function, see Section 19.5.4.1(4) Transmission Start Wait Function.) • [When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] • Outputs a transmission interrupt.
(4) [When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)]	<ul style="list-style-type: none"> • Transmits the data set in the UART (for wait) transmit data register.
(5) <ul style="list-style-type: none"> • When transmitting data continuously, sets another piece of transmission data in the UART transmit data register (RLN3nLUTDR register), waits for the generation of an interrupt request. 	<ul style="list-style-type: none"> • Transmits a parity bit when parity is used.
(6)	<ul style="list-style-type: none"> • Transmits 1 or 2 stop bits.

Table 19.88 LIN/UART Interface (UART Mode) Transmission Processing (2/2)

Software Processing	LIN/UART Interface Processing
<p>(7) [When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)]</p> <ul style="list-style-type: none"> • If another item of transmission data is set, goes to step (3). <p>[When the UTIGTS bit is 1 (a transmission interrupt is generated upon end of transmission)]</p> <ul style="list-style-type: none"> • When transmitting data continuously, goes to step (2). 	<p>[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)]</p> <ul style="list-style-type: none"> • If another piece of transmission data is set, goes to step (3). • If another piece of transmission data is not set, clears the transmit status flag. <p>[When the UTIGTS bit is 1 (a transmission interrupt is generated upon end of transmission)]</p> <ul style="list-style-type: none"> • Generates RLIN3n transmission interrupt request. • Clears the transmission status flag.

(1) Continuous Transmission

The LIN/UART interface (in UART mode) can transmit multiple sets of data continuously by using the RLIN3nLUTDR register. **Figure 19.22** shows an operation example where the transmission interrupt generation timing is the start of transmission and an operation example where the transmission interrupt generation timing is the end of transmission.

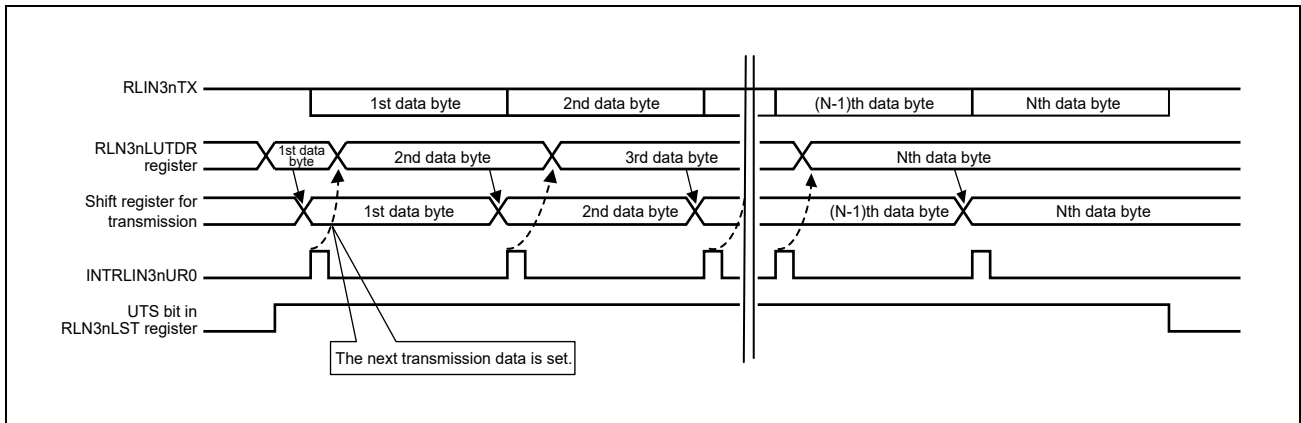


Figure 19.22 Operation Example of LIN/UART Interface (UART Mode) Continuous Transmission

An interrupt can be generated at the end of a transmission by changing the UTIGTS bit in the RLIN3nLUOR1 register from 0 to 1 after the start of transmission of final data, provided that the transmission interrupt generation timing is the start of transmission and the end of transmission of final data needs to be known.

(2) UART Buffer Transmission

The LIN/UART interface (in UART mode) has a maximum of nine bytes of UART buffers, and thus it is capable of performing continuous transmissions through the use of UART buffers.

Figure 19.23 shows the UART buffer transmission operation of the LIN/UART interface (in UART mode).

Table 19.89 describes the UART buffer data transmission processing.

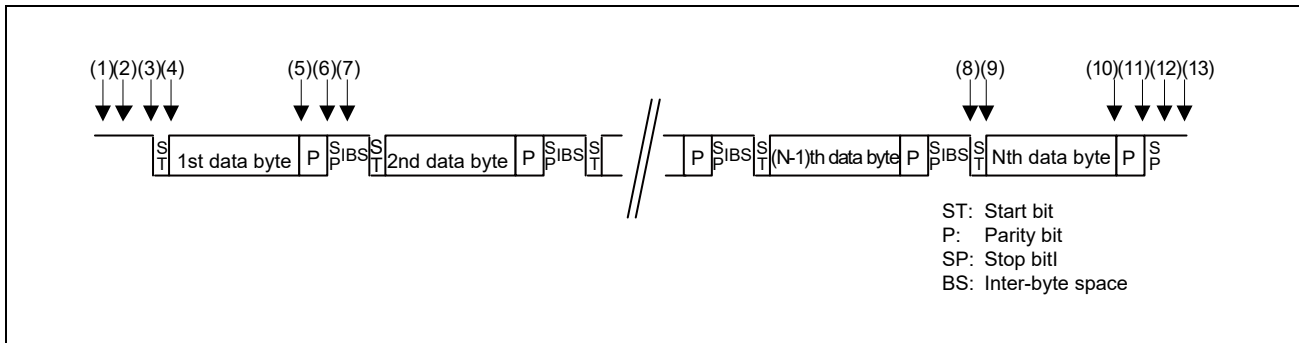


Figure 19.23 UART Buffer Transmission of LIN/UART Interface (in UART Mode)

Table 19.89 UART Buffer Transmission Processing of LIN/UART Interface (in UART Mode) (1/2)

Software Processing	LIN/UART Interface Processing
(1) <ul style="list-style-type: none"> • Sets a baud rate • Sets noise filter ON/OFF • Enables error detection • Sets data format • Sets an interrupt generation timing to the end of transmission. • Clears the LIN/UART interface from LIN reset mode. • Sets the transmit enable bit (UTOE bit) to 1 	<ul style="list-style-type: none"> • Waits for a transmission trigger (RTS bit) by software
(2) <ul style="list-style-type: none"> • Sets the UART buffer data length and whether the system must wait for the start of transmission. • Sets the transmission data in the UART data buffer 0 register (RLN3nLUDB0) and the LIN data buffer b register (RLN3nLDBRb). (b = 1 to 8) • Sets the UART buffer data transmission start bit (RTS). 	<ul style="list-style-type: none"> • Sets the transmit status flag.
(3) Waits for an interrupt request.	Transmits a start bit. (When switching from reception to transmission during half-duplex communication, transmits the start bit upon completion of the stop bit for reception. For details about this function, see Section 19.5.4.1(4) Transmission Start Wait Function.)
(4)	Transmits the data set in the UART data buffer 0 register (RLN3nLUDB0) and the LIN/UART data buffer b register (RLN3nLDBRb).
(5)	Transmits a parity bit when parity is used.
(6)	Transmits 1 or 2 stop bits (When the number of data set in UART buffer data length select bits is 1, proceeds to (12).)

Table 19.89 UART Buffer Transmission Processing of LIN/UART Interface (in UART Mode) (2/2)

Software Processing	LIN/UART Interface Processing
(7) Waits for an interrupt request.	Transmits an inter-byte space (idle). Repeats steps (3) to (7) until frame count –1 that was set in the UART buffer data length select bits is reached.
(8)	Transmits a start bit.
(9)	Transmits the data set in the LIN/UART data buffer b register (RLN3nLDBRb).
(10)	Transmits a parity bit when parity is used.
(11)	Transmits 1 or 2 stop bits.
(12)	<ul style="list-style-type: none"> • Sets the successful buffer transmission flag. • Clears the UART Buffer Transmission Start bit (RTS). • A transmission interrupt request signal. • Clears the transmission status flag.
(13) <ul style="list-style-type: none"> • Checks the RLN3nLST register, and clears flags • In the case of continuous data transmission, goes to step (2). 	

(a) UART Buffer Transmission

For a 9-byte transmission, the contents stored in the RLN3nLUDB0 and RLN3nLDBR1 to RLN3nLDBR8 registers are transmitted to data areas 1 to 9. The RLN3nLUDB0 register is used only if 9-byte transmission is set. In other cases, the RLN3nLDBR1 to RLN3nLDBR8 registers are selected depending upon the length of data involved. For a 4-byte transmission, the contents stored in the RLN3nLDBR1 to RLN3nLDBR4 registers are transmitted to data areas 1 to 4, but the contents of the RLN3nLDBR5 to RLN3nLDBR8 registers are not transmitted. An RLIN3n transmission interrupt is generated after the number of data specified in the MDL [3:0] bits of the RLN3nLDFC register is transmitted. The spaces between transmission data items can be set in the IBS bit in the RLN3nLSC register.

Figure 19.24 shows a UART buffer and the transmission processing.

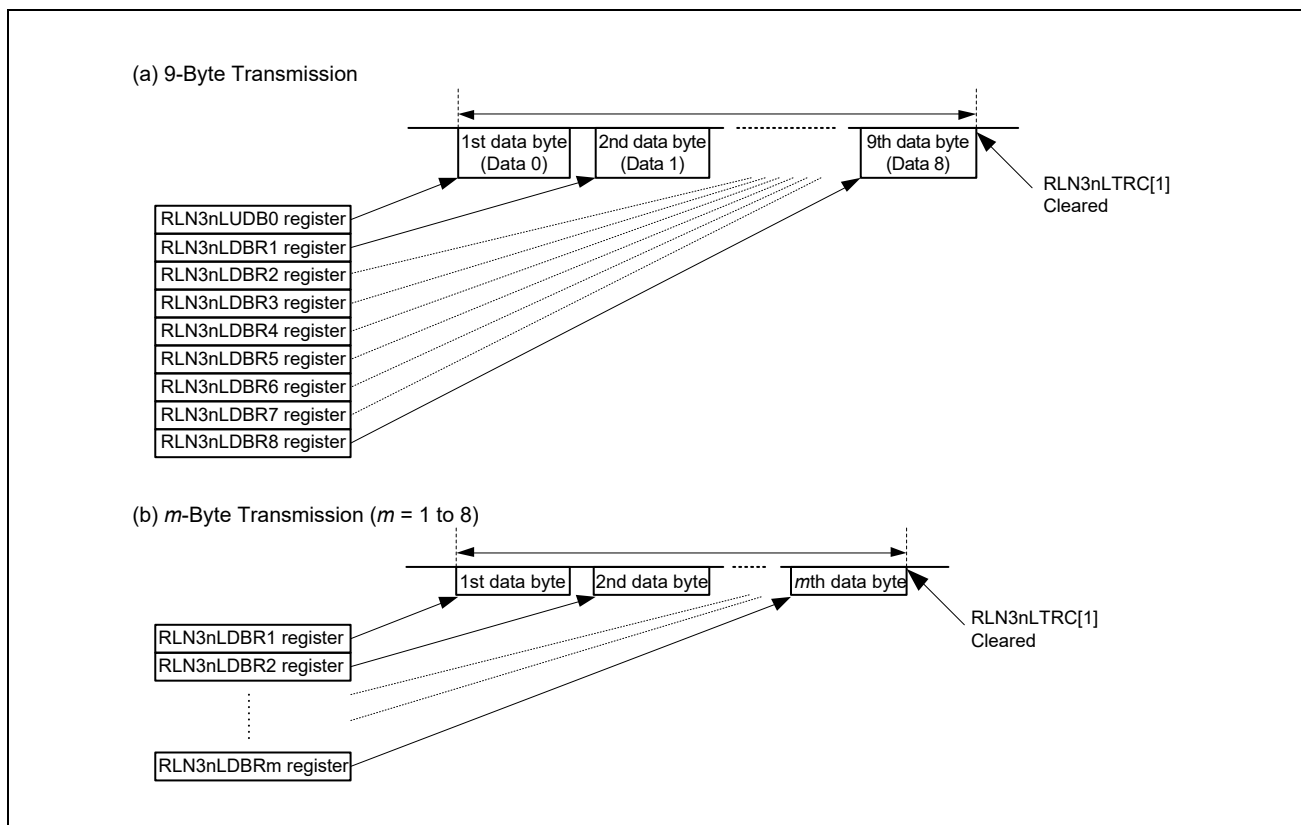


Figure 19.24 UART Buffer and Transmission Processing

(3) Data Transmission

One bit of data is transmitted per Tbit.

In half-duplex communication, if the BERE bit in the RLN3nLEDE register is 1 (bit error detection enabled), the transmission data and the input pin level are compared bit by bit during data transmission, and the results are stored in the BER flag in the RLN3nLEST register (see **Section 19.5.4.5, Error Status**). The timing at which the input pin is sampled during data transmission can vary depending upon the settings of the LPRS[2:0] and NSPB[3:0] bits in the RLN3nLWBR register.

The bit error detection timing in UART mode is shown in **Table 19.90**.

Table 19.90 Error Detection Timing in UART Mode

Sampling Count Per Bit	Bit Error Detection Timing
6 samples	3rd clock cycle + 1 prescaler clock
7 samples	4th clock cycle + 1 prescaler clock
8 samples	4th clock cycle + 1 prescaler clock
9 samples	5th clock cycle + 1 prescaler clock
10 samples	5th clock cycle + 1 prescaler clock
11 samples	6th clock cycle + 1 prescaler clock
12 samples	6th clock cycle + 1 prescaler clock
13 samples	7th clock cycle + 1 prescaler clock
14 samples	7th clock cycle + 1 prescaler clock
15 samples	8th clock cycle + 1 prescaler clock
16 samples	8th clock cycle + 1 prescaler clock

Example of Data Transmission Timing (when 1 Tbit = 16 Sampling) is shown in **Figure 19.25**.

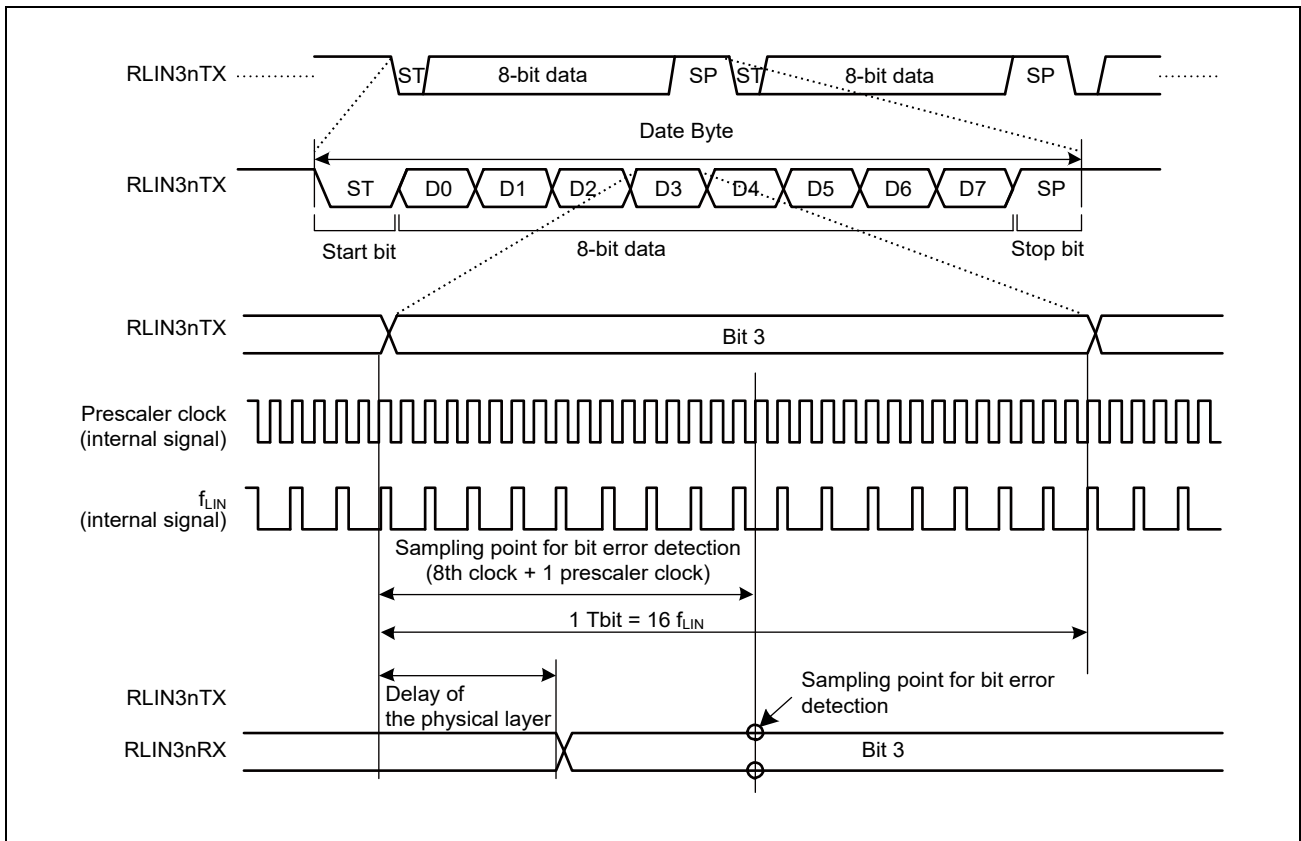


Figure 19.25 Example of Data Transmission Timing (When 1 Tbit = 16 Sampling)

(4) Transmission Start Wait Function

For performing half-duplex communication, the LIN/UART interface (in UART mode) has the function of securing the reception stop bit length when switching from reception to transmission.

If it is desired to delay the start of transmission until the stop bits for the reception are completed, set data in the RLIN3nLUWTDR register, which is used only for the wait function, instead of setting transmission data in the RLIN3nLUTDR register as a start-of-transmission request. When transmitting from the UART buffer, set 1 (UART buffer transmission started) in the RTS bit in the RLIN3nLTRC register with 1 set in the UTSW bit in the RLIN3nLDFC register. In such a case, the LIN/UART interface delays the start of transmission until the stop bits of reception data are completed.

It should be noted that even if the UART stop bit length selection bit (USBLS) in RLIN3nBLFC register is 1 (stop bits = 2 bits), delay is made only for 1 bits.

Figure 19.26 shows the operation of transmission wait function.

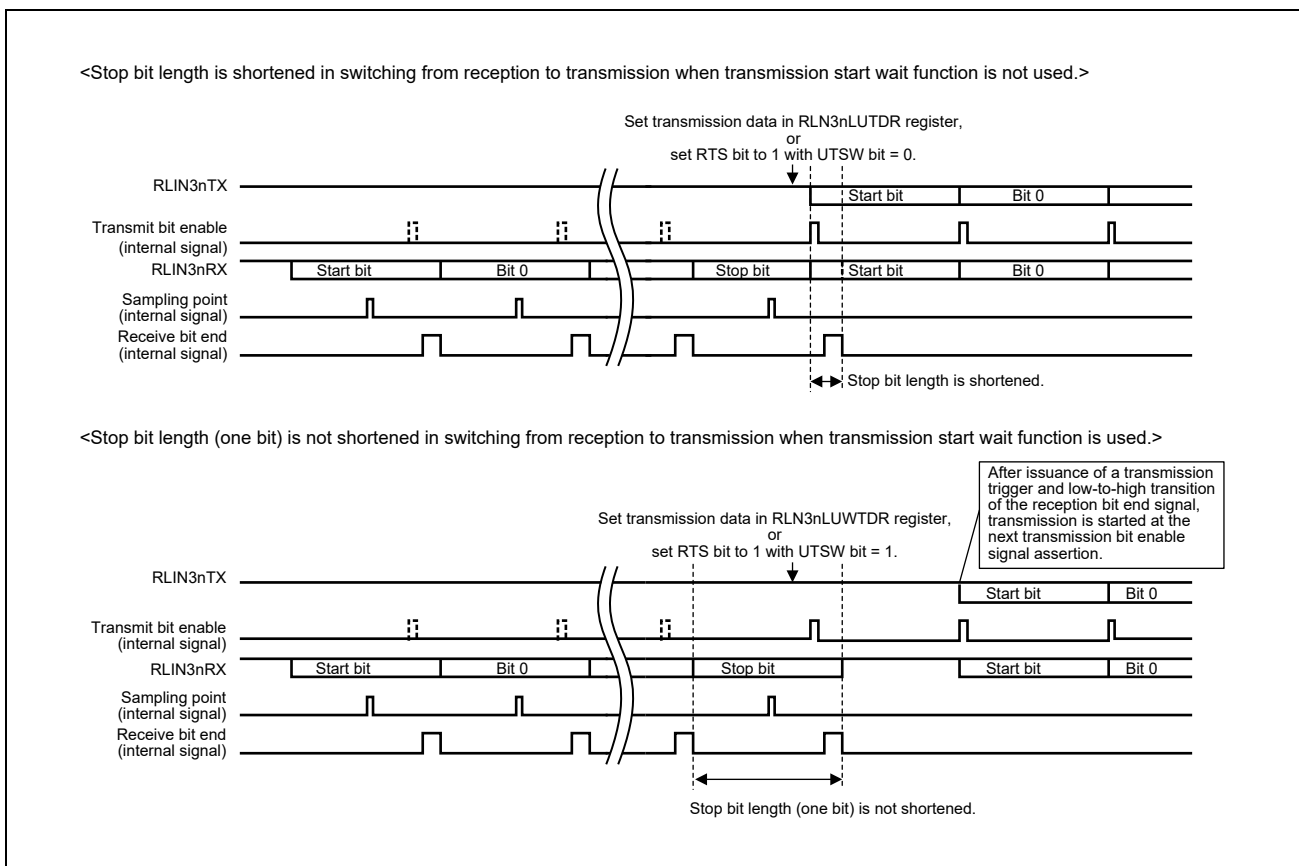


Figure 19.26 Case When Transmit Data Is Set While Stop Bits Are Being Received

19.5.4.2 Reception

Figure 19.27 shows the LIN/UART interface (in UART mode) reception operation. **Table 19.91** shows the LIN/UART interface (in UART mode) reception processing.

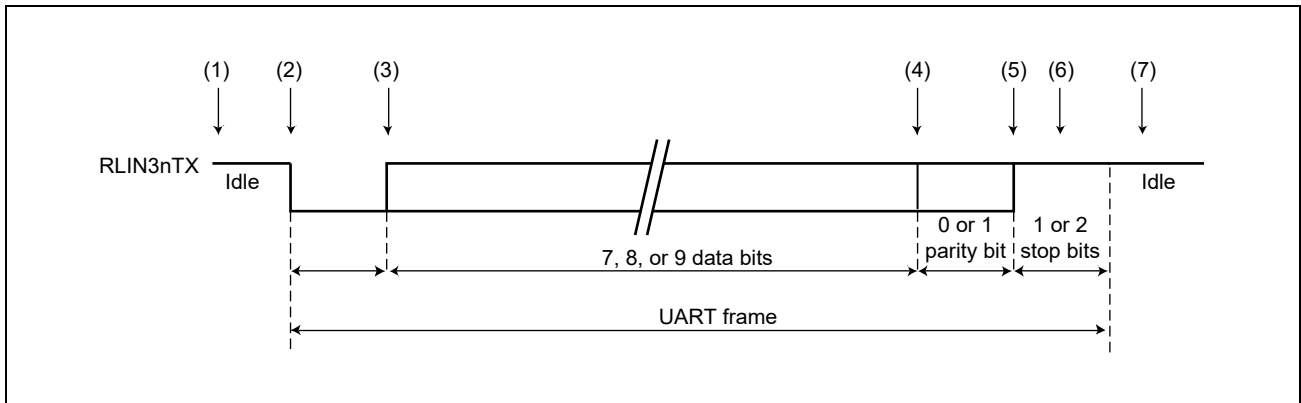


Figure 19.27 LIN/UART Interface (in UART Mode) Reception Operation

Table 19.91 LIN/UART Interface (in UART Mode) Reception Processing

Software Processing	LIN/UART Interface Processing
(1) <ul style="list-style-type: none"> • Sets a baud rate. • Sets noise filter ON/OFF. • Enables error detection. • Sets data format. • Clears the LIN/UART interface from LIN reset mode. • Sets the receive enable bit (UROE bit) to 1. 	<ul style="list-style-type: none"> • Waits for the reception to be enabled by software. • Waits for detection of a start bit.
(2) Waits for an interrupt request.	<ul style="list-style-type: none"> • Waits for a falling edge from the reception pin, and detects a start bit. • Sets the reception status flag.
(3)	Receives data.
(4)	Receives a parity bit when parity is used.
(5)	Receives only 1 stop bit.
(6)	<ul style="list-style-type: none"> • Generates a successful RLIN3n reception interrupt request. • Clears the reception status flag.
(7) Checks the RLIN3nLST register, and clears flags	Waits for a falling edge from the reception pin.

(1) Data Reception

Data reception is performed by using the synchronized RLIN3nRX (an internal signal) that is the input from the RLIN3nRX pin synchronized with the prescaler clock.

The Data Byte is synchronized at the falling edge of the start bit for the synchronized RLIN3nRX signal. After the falling edge is detected, resampling is performed 0.5 Tbits later when the number of sampling per 1 Tbit is even and $\{(the\ number\ of\ sampling + 1) / 2\} / (the\ number\ of\ sampling)$ Tbits later when the number is odd. If the synchronized RLIN3nRX signal is low level, the bit is recognized as a start bit. The bit is not recognized as a start bit if the RLIN3nRX signal after the reset is de-asserted is fixed to low level or if a high level is detected during the resampling.

After the start bit is detected, 1 bit is sampled per Tbit.

However, when the BERE bit in the RLN3nLEDE register is 1, the sampling point is the same as the bit error detection timing.

The LIN/UART interface has a noise filtering function for received data. If the LRDNFS bit in the RLN3nLMD register is 0, the noise filter is used. For a sampling value, the value determined by a 3-samplings majority rule by the prescaler clock is used. If the LRDNFS bit in the RLN3nLMD register is 1, the noise filter is not used. In this case, for a sampling value, the synchronized RLIN3nRX value at the sampling position is used as is.

Figure 19.28 shows an example of data reception timing.

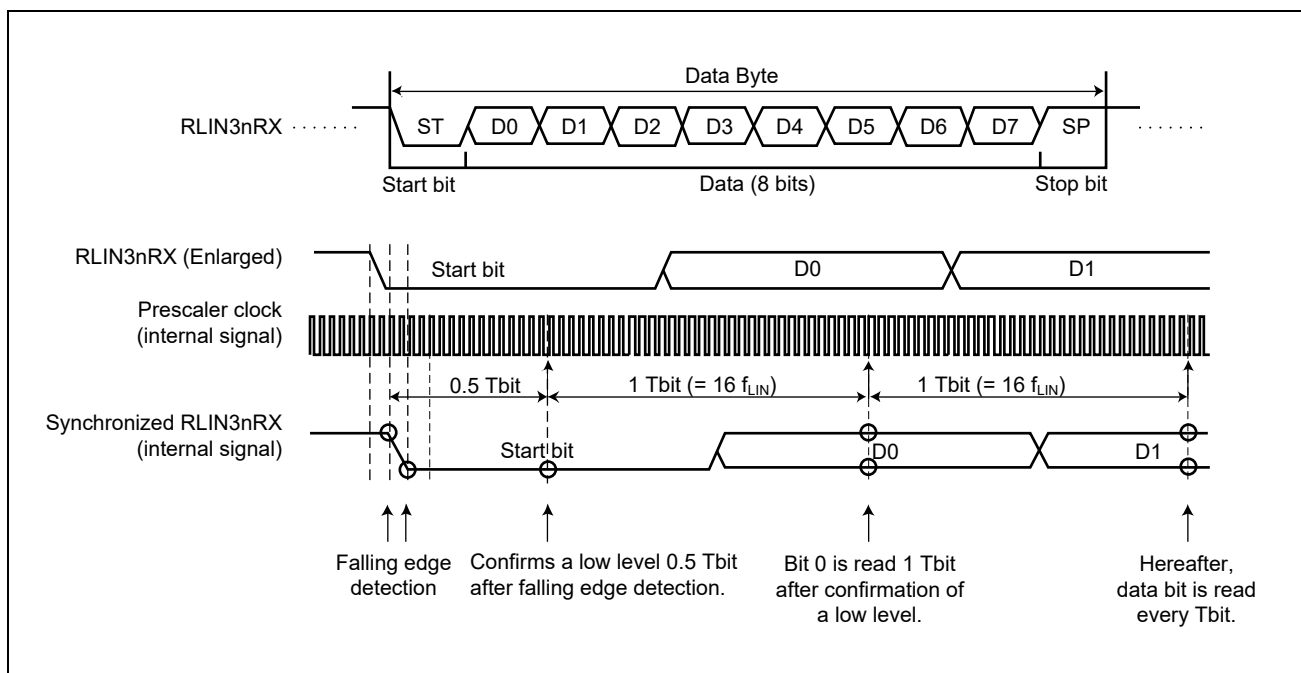


Figure 19.28 Example of Data Reception Timing (When Sampling Count is 16 in 1 Tbit)

19.5.4.3 Expansion Bits

The LIN/UART interface (in UART mode) can transmit and receive 9-bit long data by setting the UEBE bit in the RLN3nLUOR1 register to 1.

(1) Expansion Bit Transmission

The LIN/UART interface (in UART mode) can transmit 9-bit long data when the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1 and by writing the 9-bit data to either the UART transmission data register (RLN3nLUTDR) or the UART wait transmission data register (RLN3nLUWTDR).

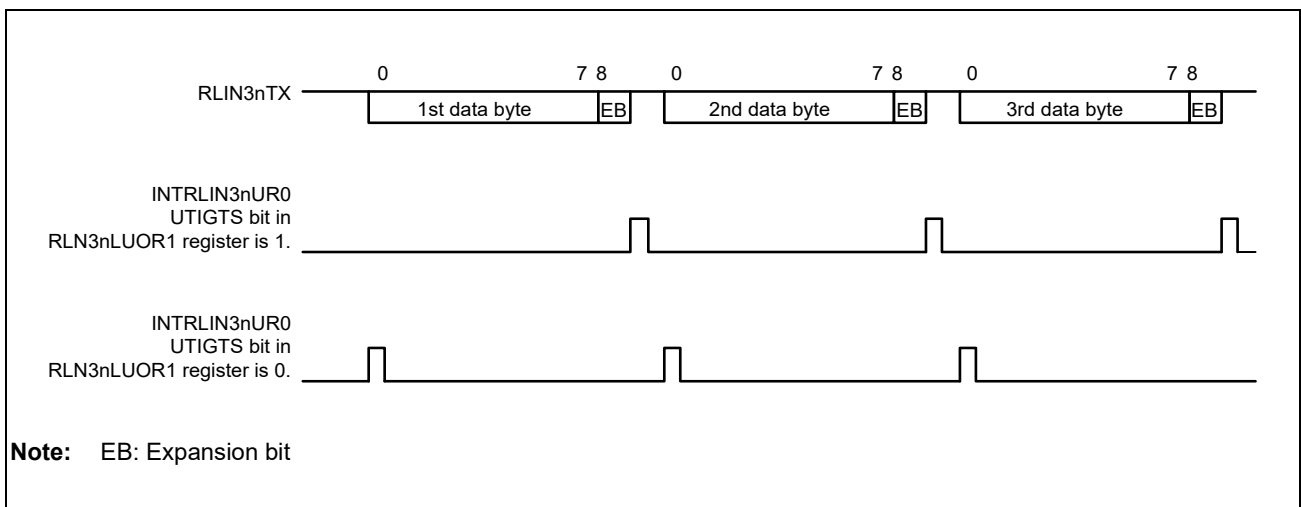


Figure 19.29 Transmission Example When Expansion Bit is Enabled (LSB First)

(2) Expansion Bit Reception

With the LIN/UART interface (in UART mode), 9-bit data can always be received without requiring a comparison of expansion bits, provided that the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1, the expansion bit comparison disable bit (UECD) is 1, and the expansion bit data comparison enable bit (UEBDCE) is 0. Irrespective of the particular setting of the expansion bit detection level selection bit (UEBDL) in the UART option register 1 (RLN3nLUOR1), a successful RLIN3n reception interrupt is generated when 9-bit data is received.

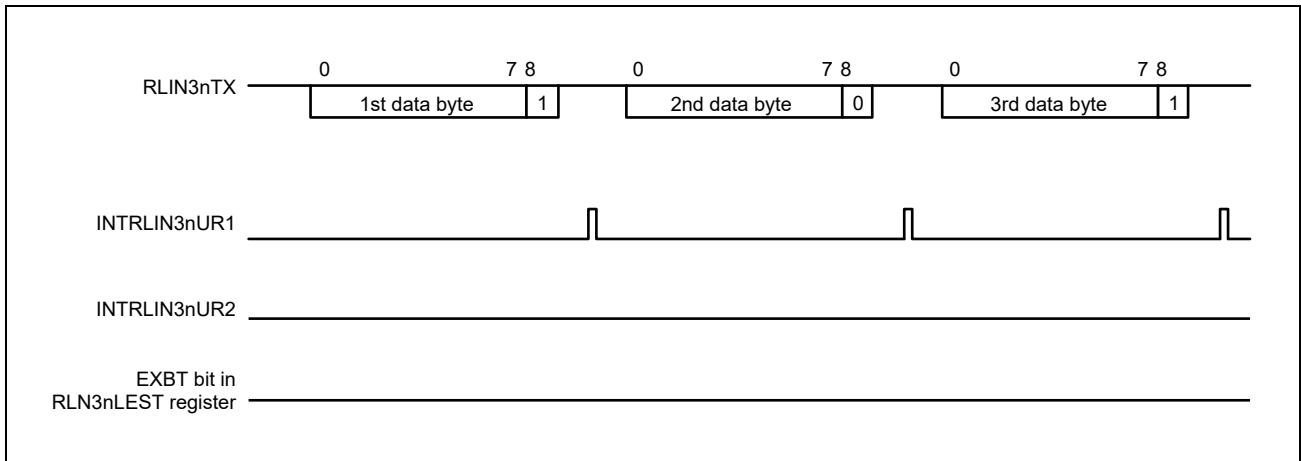


Figure 19.30 Expansion Bit Reception Example (LSB First)

(3) Expansion Bit Reception (with Expansion Bit Comparison)

The LIN/UART interface (in UART mode) can compare received expansion bits and the UEBDL bits when the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1, the expansion bit comparison disable bit (UECD) is 0 and the expansion bit/data comparison enable bit (UEBDCE) is 0.

If the level that was set in the expansion bit detection level selection bit (UEBDL) is detected, an RLIN3n status interrupt request is generated upon completion of data reception, and the expansion bit detection flag (EXBT) in the LIN error status register (RLN3nLEST) is set. If the reversed value of an expansion bit detection level is detected, successful RLIN3n reception interrupt request is generated. In either case, the received data is stored in the UART reception data register (RLN3nLURDR), unless there was an overrun error.

Figure 19.31 shows an example when the expansion bit detection level selection bit (UEBDL) is set to 0.

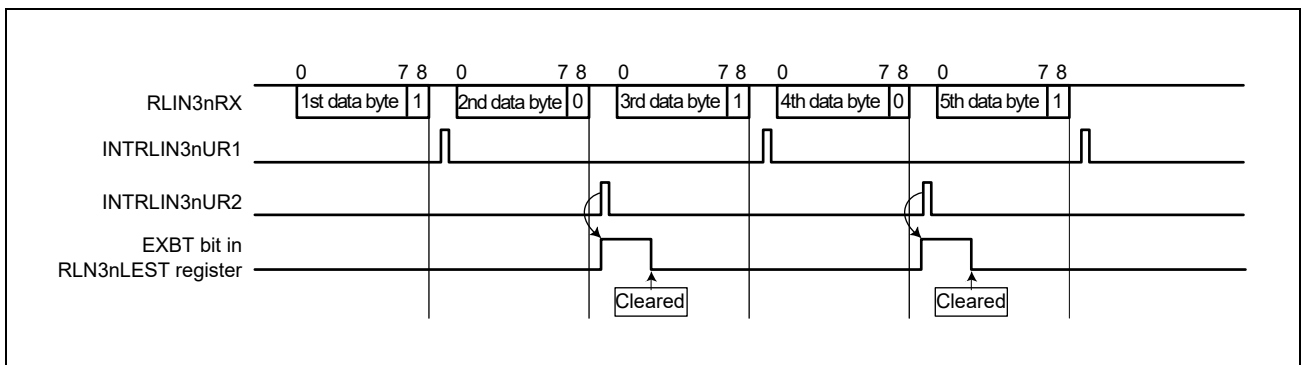


Figure 19.31 Expansion Bit Reception Example (with Expansion Bit Comparison) (LSB First, UEBDL = 0)

NOTE

- If a reception error (parity error, framing error, or overrun error) occurs in received data 0, 2, or 4 (if a reversed value of an expansion bit detection level is detected), an RLIN3n status interrupt is generated, and the error flag is updated. In this case, a successful RLIN3n reception interrupt is not generated.
- If a reception error (parity error, framing error, or overrun error) occurs in received data 1 or 3 (if an expansion bit detection level is detected), an RLIN3n status interrupt is generated, and the error flag is updated. If the overrun error occurs, the expansion bit detection flag (EXBT) is also set.

(4) Expansion Bit Reception (with Data Comparison)

If the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1, the expansion bit comparison disable bit (UECD) is 0 and the expansion bit/data comparison enable bit (UEBDCE) is 1, and if the level that was set by the expansion bit detection level selection bit (UEBDL) is detected, the LIN/UART interface (in UART mode) compares the 8 bits, excluding the expansion bit in the received data, with the a pre-set RLN3nLIDB register value.

If the result of the comparison is a match, the LIN/UART interface performs the following operations:

- Generates an RLIN3n status interrupt
- Sets an expansion bit detection flag (EXBT)
- Sets an ID match flag (IDMT)
- Stores the received data in the UART reception data register (RLN3nLURDR)

Even when the result of the comparison is a match, successful RLIN3n reception interrupt is not generated.

If the result of the comparison is not a match, no successful RLIN3n reception interrupt or RLIN3n status interrupt is generated, and the EXBT and IDMT flags are not set to 1. The received data is not stored in the UART reception data register (RLN3nLURDR).

When changing the UEBDCE bit to 0, make the change before the reception of another set of data is finished.

Figure 19.32 shows an example when the expansion bit detection level selection bit (UEBDL) is set to 0.

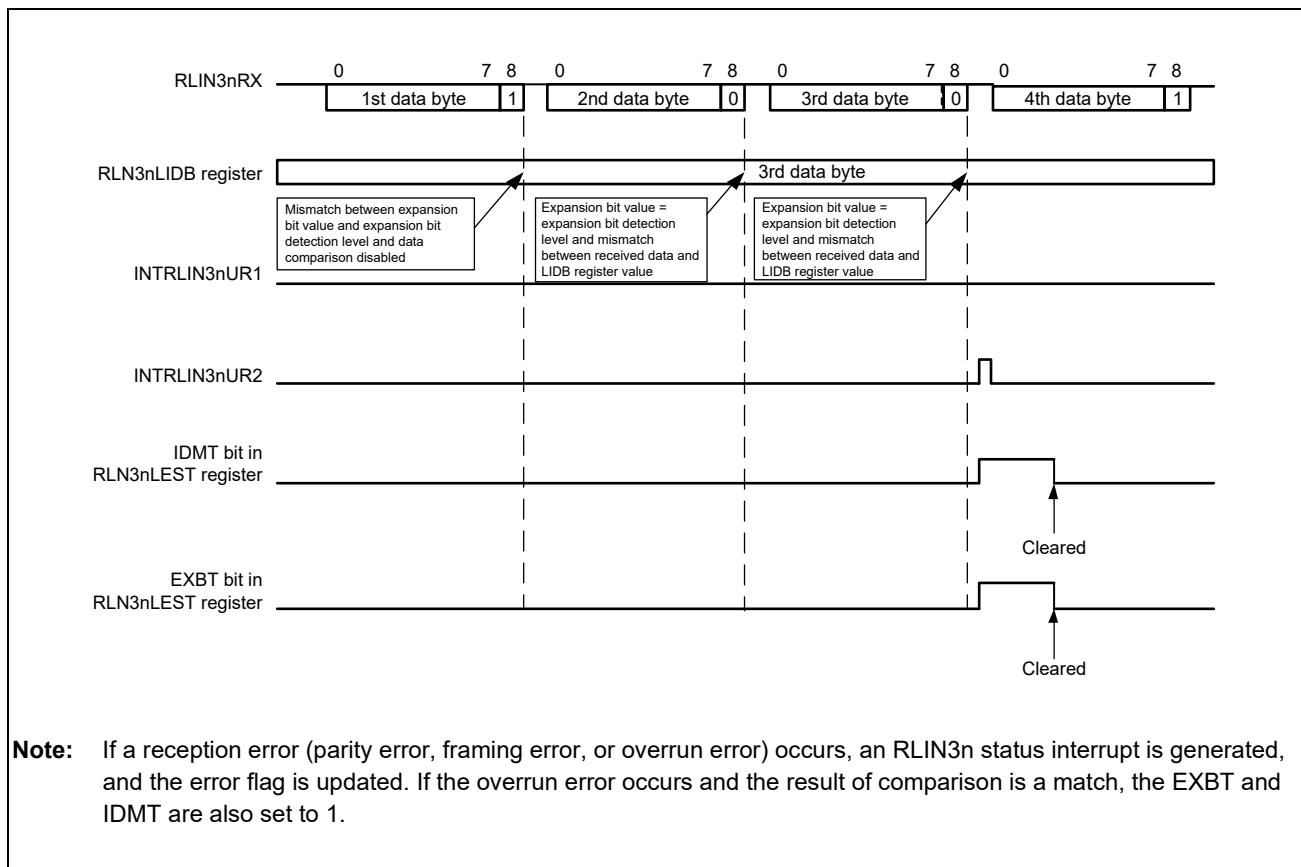


Figure 19.32 Expansion Bit Reception Example (with Data Comparison) (LSB First, UEBDL = 0)

19.5.4.4 Status

In UART mode, the LIN/UART interface can detect five types of statuses.

Two statuses, successful UART buffer transmission and error detection, can generate interrupt requests.

Table 19.92 shows the types of statuses available in UART mode.

Table 19.92 Types of Statuses in UART Mode

Status	Status Set Condition	Status Clear Condition	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN-reset-mode, if actually the LIN/UART interface is cleared from LIN reset mode.	After the OM0 bit in the RLN3nLCUC register is set to LIN reset mode, if actually the LIN/UART interface enters LIN reset mode.	OMM0 bit in RLN3nLMST register	—
Successful UART buffer transmission	<ul style="list-style-type: none"> When the transmission is finished of data equal to the length set in the MDL bits in the RLN3nLDFC register. When the UTIGTS bit in the RLN3nLUOR1 register is 0 (transmission interrupt request is generated upon start of transmission), the transmission of the last data of the data length set by the MDL bit in the RLN3nLDFC register is started. When the UTIGTS bit in the RLN3nLUOR1 register is 1 (transmission interrupt request is generated upon end of transmission), the transmission of the data length set by the MDL bit in the RLN3nLDFC register is ended. 	<ul style="list-style-type: none"> When cleared by software Transition to LIN reset mode 	FTC flag in RLN3nLST register	✓
Error detection	If any of the UPER flag, IDMT flag, EXBT flag, FER flag, OER flag, and BER flag in the RLN3nLEST register is set to 1 (error detected).	<ul style="list-style-type: none"> When cleared by software*1 Transition to LIN reset mode 	ERR flag in RLN3nLST register	✓
Transmission status	<ul style="list-style-type: none"> When data is written to the RLN3nLUTDR or RLN3nLUWTDRC register. When 1 is written to the RTS bit in the RLN3nLTRC register. 	<ul style="list-style-type: none"> The transmission of the data set in the RLN3nLUTDR or RLN3nLUWTDRC register is complete, but another transmission data item is not set The transmission of the data in the UART buffer is complete, and the RTS bit in the RLN3nLTRC register is cleared Transition to LIN reset mode 	UTS flag in RLN3nLST register	—
Reception status	<ul style="list-style-type: none"> When a start bit is detected. 	<ul style="list-style-type: none"> When a sampling point for stop bits is detected Transition to LIN reset mode 	URS flag in RLN3nLST register	—

Note: Writing a 0 to the UPER, IDMT, EXBT, FER, OER, and BER flags in the RLN3nLEST register when the LIN reset mode is being canceled sets the ERR flag in the RLN3nLST register to 0.

19.5.4.5 Error Status

Types of Error Statuses

In UART mode, the LIN/UART interface can detect four types of errors and two types of statuses. The condition of these statuses can be verified by using the corresponding bits in the RLN3nLEST register.

Table 19.93 shows available status types.

Table 19.93 Types of Statuses in UART Mode

Status	Error Detection Condition	Communication	Enable/Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data monitored on the receive pin do not match*1	Continues until the transmission of the set transmission data is finished.	Enabled	BER flag in RLN3nLEST register
Overrun error	After received data is stored in the RLN3nLURDR register, another data item is received before the data is read. (In this case, no data is stored in the RLN3nLURDR register).	— (Reception is finished by the time this error is detected)	Enabled	OER flag in RLN3nLEST register
Framing error	When the first stop bit is low level in the reception processing.	— (Reception is finished by the time this error is detected)	Enabled	FER flag in RLN3nLEST register
Parity error	The received parity value fails to match the parity value calculated from the received data	Continues until the data reception is finished.	Disabled*2	UPER flag in RLN3nLEST register
Expansion bit detection	The value of the received expansion bit matches the value of the UEBDL bit in the RLN3nLUOR1 register.	—	Enabled	EXBT flag in RLN3nLEST register
ID match detection	The value of the received expansion bit matches the value of the UEBDL bit in the RLN3nLUOR1 register and the 8-bit receive data excluding the expansion bit matches the value of the RLN3nLIDB register.	—	Enabled	IDMT flag in RLN3nLEST register

Note 1. In the case of transmission from the UART buffer, bit errors are detected even in the space between UART frames (inter-byte space).

Note 2. Setting the UPS[1:0] bits in the RLN3nLBFC register to 10_B (0 parity) disables the checking of parity bit values. In this case, no parity error is generated.

The error status is cleared by software or at a transition to LIN reset mode.

19.5.5 LIN Self-Test Mode

The LIN/UART interface provides a LIN Self-Test mode.

When the LIN/UART interface enters the LIN Self-Test mode, RLIN3nTX and RLIN3nRX are disconnected from external pins and RLIN3nTX and RLIN3nRX are connected to the LIN/UART interface internally. Therefore, the frame transmitted from RLIN3nTX is looped back to RLIN3nRX. The LIN Self-Test mode can perform tests exclusively in LIN mode.

The Self-Test can be performed in the following four modes:

- LIN master Self-Test mode (transmission): Header transmission and response transmission
- LIN master Self-Test mode (reception): Header transmission and response reception
- LIN slave Self-Test mode (transmission): Header reception and response transmission
- LIN slave Self-Test mode (reception): Header reception and response reception

In LIN Self-Test mode, operation is performed at the fastest baud rate, regardless of the setting of the baud rate generator.

Regardless of the setting of the baud rate related registers, the baud rate setting is the LIN communication clock source/16 [bps]. (The NSPB bits in the RLN3nLWBR register should be set to 0000_B or 1111_B.)

(The LPRS bits in the RLN3nLWBR register should be set to 000_B.)

In addition, in LIN Self-Test mode, the following functions are not supported.

- LIN wake-up mode
- Frame separate mode
- Multi-byte response transmission/reception
- LIN slave mode (Auto baud rate)
- Frame/response timeout error

Do not use these functions.

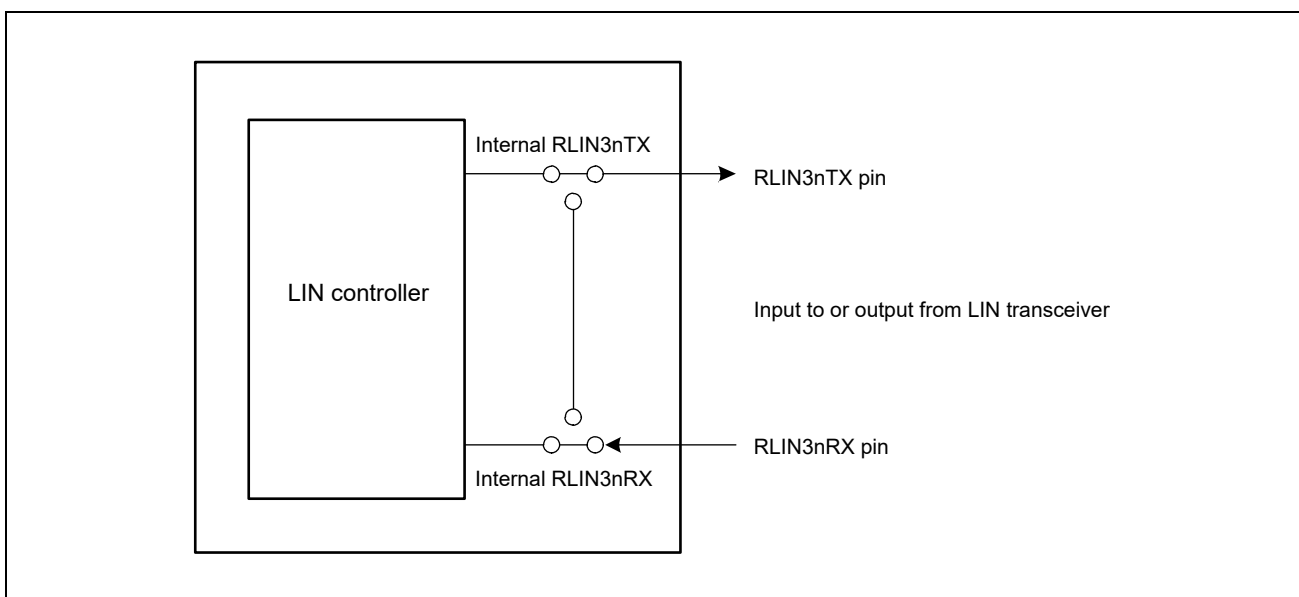


Figure 19.33 Connection in LIN Reset Mode, LIN Mode, and UART Mode

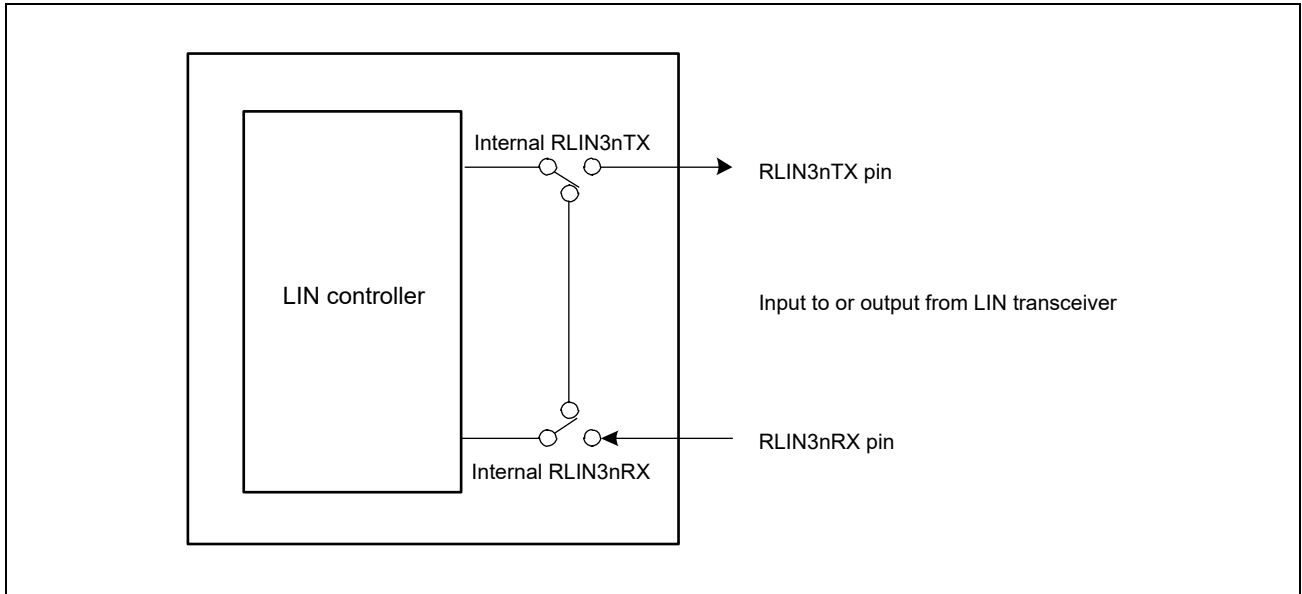


Figure 19.34 Connection in LIN Self-Test Mode

19.5.5.1 Transitioning to LIN Self-Test Mode

Writing to the RLN3nLSTC register makes a transition to the LIN Self-Test mode.

The LSTM bit in the RLN3nLSTC register set to 1 indicates that the mode has transitioned to the LIN Self-Test mode.

To transition to LIN Self-Test mode, be sure to execute a specific sequence. In that sequence, information must be written three times consecutively to the LIN Self-Test control register as shown below:

- Transition to LIN reset mode
 - Set the OM0 bit in the RLN3nLCUC register to 0 (in LIN reset mode).
 - Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (in LIN reset mode).
- Select a LIN mode
 - LMD bits in RLN3nLMD = 00_B (in LIN master mode) or 11_B (in LIN slave mode [fixed baud rate])
- 1st write: RLN3nLSTC register = 1010 0111_B (A7_H)
- 2nd write: RLN3nLSTC register = 0101 1000_B (58_H)
- 3rd write: RLN3nLSTC register = 0000 0001_B (01_H)
- Verify the transition to LIN Self-Test mode
 - Read the LSTM bit in the RLN3nLSTC register; verify that it is 1 (in LIN Self-Test mode).

If the key of the first write (A7_H) is written twice by mistake, the transition to LIN Self-Test mode is canceled. The above sequence should be retried from the step of first write. In addition, if a write to another LIN-related register is performed during transition to LIN Self-Test mode (three consecutive write operations to the RLN3nLSTC register), the transition is also canceled.

19.5.5.2 Transmission in LIN Master Self-Test Mode

To execute a Self-Test on LIN master transmission, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.

RLN3nLWBR register = 0000 000_x_B

RLN3nLBRP0 register = xxxx xxxx_B*¹

RLN3nLBRP1 register = xxxx xxxx_B*¹

RLN3nLMD register = 00xx xx00_B*¹

- Set the interrupt enable and error enable related registers.

RLN3nLIE register = 0000 xxxx_B*²

RLN3nLEDE register = x000 x0xx_B

- Set the break field and space related registers.

RLN3nLBFC register = 00xx xxxx_B

RLN3nLSC register = 00xx 0xxx_B

- Release from the LIN reset mode.

Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.

- Set the transmit frame related registers.

RLN3nLDFC register = 00x1 xxxx_B

RLN3nLIDB register = xxxx xxxx_B

RLN3nLDRB1 to RLN3nLDRB8 registers = xxxx xxxx_B

- Header transmission → response transmission started

Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started).

The LIN master Self-Test mode (transmission) is executed. In this mode, interrupt is generated, and status and error status are also updated. The checksum is automatically calculated by the LIN/UART interface. To suspend the LIN master Self-Test mode (transmission) being executed, write 0 (in LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.

- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is stored as a reversed value because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the transmission fails to complete due to an error, the corresponding error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Arbitrary value can be specified.

Note 1. The following register settings are not reflected to the operation of the LIN Self-Test mode. The RLN3nLBRP0 register, the RLN3nLBRP1 register and the LCKS bit in the RLN3nLMD register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 6, Interrupts**.

Note 3. When the successful header transmission interrupt and the successful frame transmission interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not completed before the generation of the successful frame transmission interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header transmission interrupt enabled).
The time required from the setting of the successful header transmission flag to the setting of the successful

frame/wake-up transmission flag is calculated by the following formula.

$$10 \times (\text{number of data bytes} + 1) \text{ [Tbit]}$$

$$1 \text{ Tbit} = 1/\text{frequency of LIN communication clock source} \times 16$$

19.5.5.3 Reception in LIN Master Self-Test Mode

To execute a Self-Test on LIN master reception, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.

RLN3nLWBR register = 0000 000_xB

RLN3nLBRP0 register = xxxx xxxx_B*¹

RLN3nLBRP1 register = xxxx xxxx_B*¹

RLN3nLMD register = 00xx xx00_B*¹

- Set the interrupt enable and error enable related registers.

RLN3nLIE register = 0000 xxxx_B*²

RLN3nLEDE register = x000 x0xx_B

- Set the break field and space related registers.

RLN3nLBFC register = 00xx xxxx_B

RLN3nLSC register = 00xx 0xxx_B*¹

- Release from the LIN reset mode.

Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.

- Set the reception frame related registers.

RLN3nLDFC register = 00x0 xxxx_B

RLN3nLIDB register = xxxx xxxx_B

RLN3nLDRB1 to RLN3nLDRB8 registers = xxxx xxxx_B

RLN3nLCBR register = xxxx xxxx_B

Since the checksum value to be transmitted is not automatically calculated, users must calculate it and set it to the RLN3nLCBR register. A checksum test can be performed by setting an incorrect checksum value here.

- Header transmission → response reception started

Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started).

The LIN master Self-Test mode (reception) is executed. In this mode, interrupt is generated, and status and error status are also updated. To suspend the LIN master Self-Test mode (reception) being executed, write 0 (in LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.

- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is stored as a reversed value because the set value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the reception fails to complete due to an error, the corresponding error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Don't care

Note 1. The following register settings are not reflected to the operation of the LIN Self-Test mode. The RLN3nLBRP0 register, the RLN3nLBRP1 register, the LCKS bit in the RLN3nLMD register, and the IBS bit in the RLN3nLSC register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 6, Interrupts**.

Note 3. When the successful header transmission interrupt and the successful frame reception interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not

completed before the generation of the successful frame reception interrupt, the SHIE bit in the RLIN3nLIE register should not be set to 1 (successful header transmission interrupt enabled).

Note 4. The time required from the setting of the successful header transmission flag to the setting of the successful frame/wake-up reception flag is calculated by the following formula.

$$10 \times (\text{number of data bytes} + 1) [\text{Tbit}]$$

$$1 \text{ Tbit} = 1/\text{LIN communication clock source} \times 16$$

19.5.5.4 Transmission in LIN Slave Self-Test Mode

To execute a Self-Test on LIN slave transmission, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.

RLN3nLWBR register = 0000 0000_B

RLN3nLBRP0 register = xxxx xxxx_B*¹

RLN3nLBRP1 register = xxxx xxxx_B*¹

RLN3nLMD register = 00xx 0011_B

- Set the interrupt enable and error enable related registers.

RLN3nLIE register = 0000 xxxx_B*²

RLN3nLEDE register = xx0x x00x_B

- Set the break field and space related registers.

RLN3nLBFC register = 0000 000x_B*³

RLN3nLSC register = 00xx 0001_B

- Release from the LIN reset mode.

Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.

- Set the transmit frame related registers.

RLN3nLDFC register = 00x1 xxxx_B

RLN3nLIDB register = xxxx xxxx_B

RLN3nLDBR1 to RLN3nLDBR8 registers = xxxx xxxx_B

- Header reception → response transmission started

Set the FTS bit in the RLN3nLTRC register to 1 (The header reception and the response transmission are executed in this order, without manipulating the RTS bit in the RLN3nLTRC register.).

The LIN slave Self-Test mode (transmission) is executed. In this mode, interrupt is generated, and status and error status are also updated.

The checksum is automatically calculated by the LIN/UART interface. To suspend the LIN slave Self-Test mode (transmission) being executed, write 0 (in LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.

- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is stored as a reversed value because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the transmission fails to complete due to an error, the corresponding error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Arbitrary value can be specified

Note 1. The following register settings are not reflected to the operation of the LIN Self-Test mode. The RLN3nLBRP0 register, and the RLN3nLBRP1 register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 6, Interrupts**.

Note 3. According to the setting of this register, 9.5-Tbit or 10.5-Tbit width break is output from the internal RLIN3nTX.

Note 4. When the successful header reception interrupt and the successful response transmission interrupt are used in the same interrupt processing, if the software processing of the successful header reception interrupt is not completed before the generation of the successful response transmission interrupt, the SHIE bit in the

RLN3nLIE register should not be set to 1 (successful header reception interrupt enabled).

The time required from the setting of the successful header reception flag to the setting of the successful response/wake-up transmission flag is calculated by using the following formula.

$$10 \times (\text{number of data bytes} + 1) \text{ [Tbit]}$$

$$1 \text{ Tbit} = 1/\text{frequency of LIN communication clock source} \times 16$$

19.5.5.5 Reception in LIN Slave Self-Test Mode

To execute a Self-Test on LIN slave reception, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.

RLN3nLWBR register = 0000 0000_B

RLN3nLBRP0 register = xxxx xxxx_B*¹

RLN3nLBRP1 register = xxxx xxxx_B*¹

RLN3nLMD register = 00xx 0011_B

- Set the interrupt enable and error enable related registers.

RLN3nLIE register = 0000 xxxx_B*²

RLN3nLEDE register = xx0x x00x_B

- Set the break field and space related registers.

RLN3nLBFC register = 0000 000x_B*³

RLN3nLSC register = 00xx 0001_B*¹

- Release from the LIN reset mode.

Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.

- Set the reception frame related registers.

RLN3nLDFC register = 00x0 xxxx_B

RLN3nLIDB register = xxxx xxxx_B

RLN3nLDBR1 to RLN3nLDBR8 registers = xxxx xxxx_B

RLN3nCBR register = xxxx xxxx_B

Since the checksum value to be transmitted is not automatically calculated, users must calculate it and set it to the RLN3nLCBR register. A checksum test can be performed by setting an incorrect checksum value here.

- Header reception → response reception started

Set the FTS bit in the RLN3nLTRC register to 1 (header reception or wake-up transmission/reception started).

(Without any setting of the RTS bit in the RLN3nLTRC register, the header reception and the response reception are executed in this order.)

The LIN slave Self-Test mode (reception) is executed. In this mode, interrupt is generated, and status and error status are also updated. To suspend the LIN slave Self-Test mode (reception) being executed, write 0 (in LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.

- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBR_b (b = 1 to 8), and RLN3nLCBR registers (the data is stored as a reversed value because the set value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the reception fails to complete due to an error, the corresponding error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Arbitrary value can be specified

Note 1. The following register settings are not reflected to the operation of the LIN Self-Test mode. The RLN3nLBRP0 register, the RLN3nLBRP1 register, and the IBS bit in the RLN3nLSC register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 6, Interrupts**.

Note 3. According to the setting of this register, 9.5-Tbit or 10.5-Tbit width break is output from the internal RLIN3nTX.

Note 4. When the successful header reception interrupt and the successful response reception interrupt are used in the same interrupt processing, if the software processing of the successful header reception interrupt is not completed before the generation of the successful response reception interrupt, the SHIE bit in the RLIN3nLIE register should not be set to 1 (successful header reception interrupt enabled).

The time required from the setting of the successful header reception flag to the setting of the successful response/wake-up reception flag is calculated by the following formula.

$$10 \times (\text{number of data bytes} + 1) \text{ [Tbit]}$$

$$1 \text{ Tbit} = 1/\text{frequency of LIN communication clock source} \times 16$$

19.5.5.6 Terminating LIN Self-Test Mode

To terminate LIN Self-Test mode, perform the procedure below:

- Write 0 (in LIN reset mode) to the OM0 bit in the RLN3nLCUC register.

If the OMM1 and OMM0 bits in the RLN3nLMST register are not 11_B, write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register. After confirming that the OMM1 and OMM0 bits in the RLN3nLMST register are set to 11_B, transition to LIN reset mode.

- Verify the cancelation of LIN Self-Test mode.

Read the LSTM bit in the RLN3nLSTC register; confirm that it is 0 (not in LIN Self-Test mode).

- Verify the transition to LIN reset mode.

Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (in LIN reset mode).

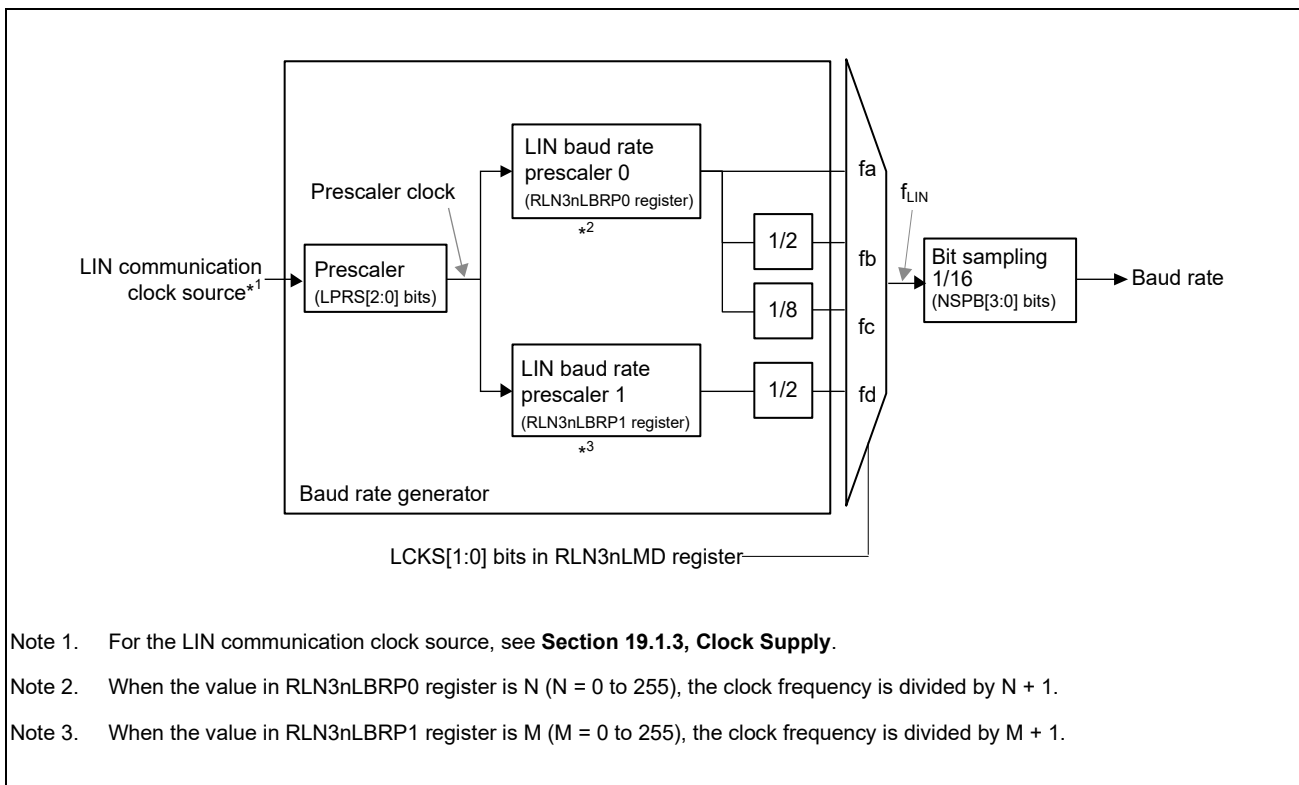
19.5.6 Baud Rate Generator

The prescaler clock is obtained by frequency-dividing the LIN communication clock source by the prescaler, and the LIN system clock (fLIN) is obtained by frequency-dividing the prescaler clock by the baud rate generator. The clock obtained by frequency-dividing the LIN system clock (fLIN) by the number of samplings is the baud rate. The reciprocal of this baud rate is called the bit time (Tbit).

The LIN/UART interface has two types of baud rate generators. The baud rate generator to be used is switched according to the mode.

19.5.6.1 LIN Master Mode

Figure 19.35 shows a block diagram of baud rate generation in LIN master mode.



Note 1. For the LIN communication clock source, see Section 19.1.3, Clock Supply.

Note 2. When the value in RLN3nLBRP0 register is N (N = 0 to 255), the clock frequency is divided by N + 1.

Note 3. When the value in RLN3nLBRP1 register is M (M = 0 to 255), the clock frequency is divided by M + 1.

Figure 19.35 Block Diagram of Baud Rate Generation in LIN Master Mode

By setting the RLN3nLBRP0 register so that fa is 307200 Hz (= 19200 × 16), the resulting system clock frequencies are fa = 19200 × 16, fb = 9600 × 16, and fc = 2400 × 16. These system clock frequencies are divided by 16 in the bit timing generator, enabling bit rates of 19200 bps, 9600 bps and 2400 bps, to be generated. Also, by setting the RLN3nLBRP1 register so that fd is 166672 Hz (= 10417 × 16), the resulting system clock frequency is fd = 10417 × 16. This system clock frequency is divided by 16 in the bit timing generator, enabling 10417 bps to be generated.

Table 19.94 shows examples of baud rate (19200, 9600, 2400, and 10417 bps) generation for each LIN communication clock source frequency, and also the corresponding errors.

Table 19.94 Examples of Baud Rate (19200, 10417, 9600, and 2400 bps) Generation in LIN Master Mode

LIN communication clock source	Prescaler	Baud rate generator 0 (MN+1) frequency-divided	Baud rate generator 1 (MN+1) frequency-divided	System Clock	Baud rate	Error
CLK_LSB	1/1	130	—	fa	19230.77	+0.16%
		—	120	fd	10416.67	-0.003%
		130	—	fb	9615.38	+0.16%
		130	—	fc	2403.85	+0.16%

NOTE

Bit sampling count is 16 sampling (RLIN3nLWBR.NSPB[3:0] = 0000_B).

19.5.6.2 LIN Slave Mode

Figure 19.36 shows a block diagram of baud rate generation in LIN slave mode.

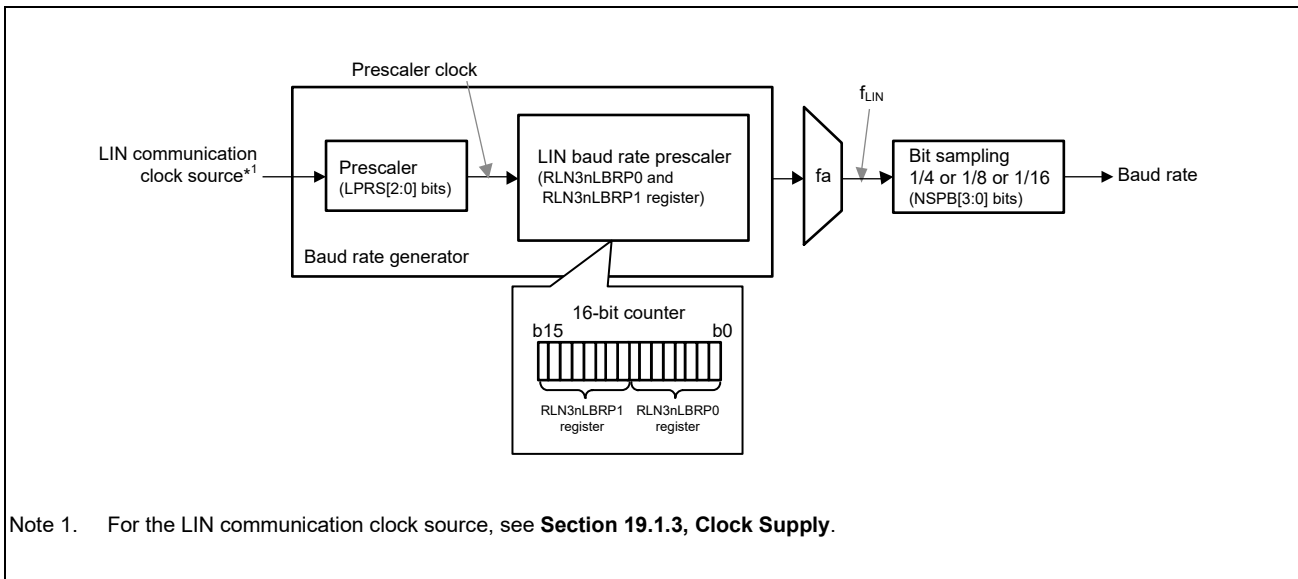


Figure 19.36 Block Diagram of Baud Rate Generation in LIN Slave Mode

Table 19.95 shows examples of baud rate (19200, 10417, 9600, and 2400 bps) generation for each peripheral function clock frequency, and also the corresponding errors.

Table 19.95 Examples of Baud Rate (19200, 10417, 9600, and 2400 bps) Generation in LIN Slave Mode [Fix Baud Rate]

LIN communication clock source	Prescaler	Baud rate generator 01 (MN+1) frequency-divided	Baud rate	Error
CLK_LSB	1/1	130	19230.77	+0.16%
		—	10416.67	-0.003%
		130	9615.38	+0.16%
		130	2403.85	+0.16%

NOTE

The bit sampling count is 16 sampling (RLN3nLWBR.NSPB[3:0] = 0000_B).

19.5.6.3 UART Mode

Figure 19.37 shows a block diagram of baud rate generation in UART mode.

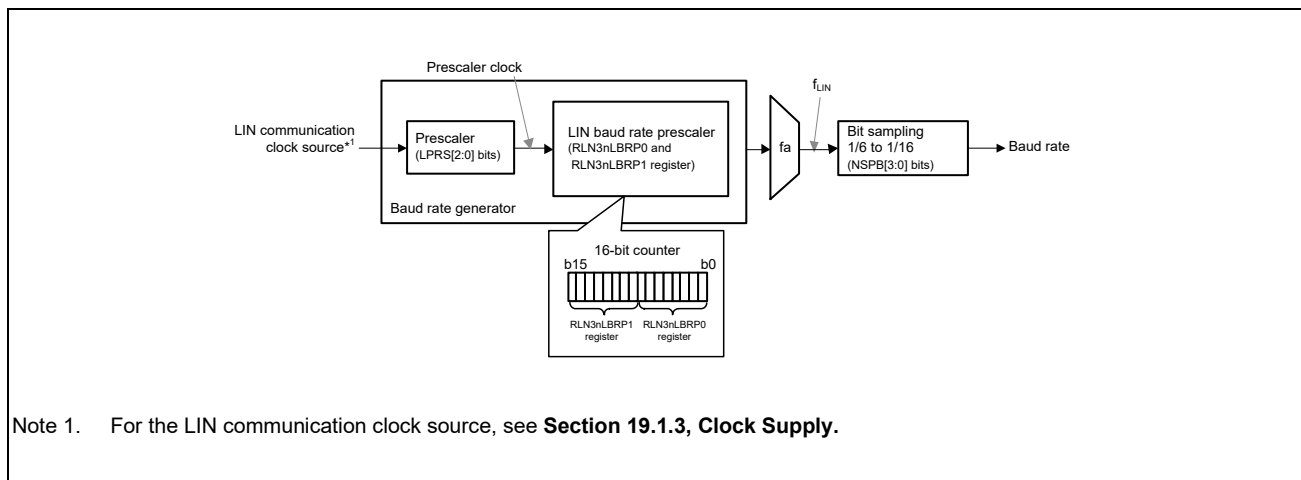


Figure 19.37 Block Diagram of Baud Rate Generation in UART Mode

UART baud rate is calculated with the following formula:

<UART baud rate>

$$= \{ \text{LIN communication clock source frequency} \} \times (\text{RLN3nLWBR.LPRS}[2:0] \text{ selection clock}) \div (\text{RLN3nLBRP0} + 1) \div \{ \text{RLN3nLWBR.NSPB}[3:0] \text{ selection count} \} \text{ [bps]}$$

Table 19.96 lists the examples of baud rate (6600000, 38400, 31250, 19200, 9600, 4800, 2400 and 1200 bps) generation for each LIN communication clock source frequency, and also the corresponding errors.

Table 19.96 UART Baud Rate Setting Examples (when LIN communication clock source = 40 MHz)

UART Baud Rate (Target Baud Rate)	Prescaler	Baud rate generator 01 (MN+1) frequency-divided	Baud rate	Error
1200 bps	1/2	1042	1199.62	-0.03%
2400 bps	1/2	521	2399.23	-0.03%
4800 bps	1/2	260	4807.69	+0.16%
9600 bps	1/2	130	9615.38	+0.16%
19200 bps	1/2	65	19230.77	+0.16%
31250 bps	1/2	40	31250.00	0.00%
38400 bps	1/2	33	37878.79	-1.36%
6600000 bps	1/1	0	6666666.67	-1.36%

NOTE

The bit sampling count for each baud rate is written below.

6.6Mbps: 6 sampling (RLN3nLWBR.NSPB[3:0] = 0101_B)

other than 6.6Mbps: 16 sampling (RLN3nLWBR.NSPB[3:0] = 1111_B)

19.5.7 Noise Filter

The LIN/UART interface has a noise filter for reducing erroneous receiving of data due to noise. By setting the LRDNFS bit in the RLIN3nLMD register to 0 (use the noise filter), the noise filter is enabled. The noise filter samples the level of the synchronized RLIN3nRX with the prescaler clock, and outputs the sampling value determined by a 3-sampling majority rule. The value of each bit of the receive data is determined based on the noise filter output.

Figure 19.38 shows the configuration of the noise filter, **Figure 19.39** shows an example of a noise filter circuit, and **Figure 19.40** shows the determination of the received data when the noise filter is used.

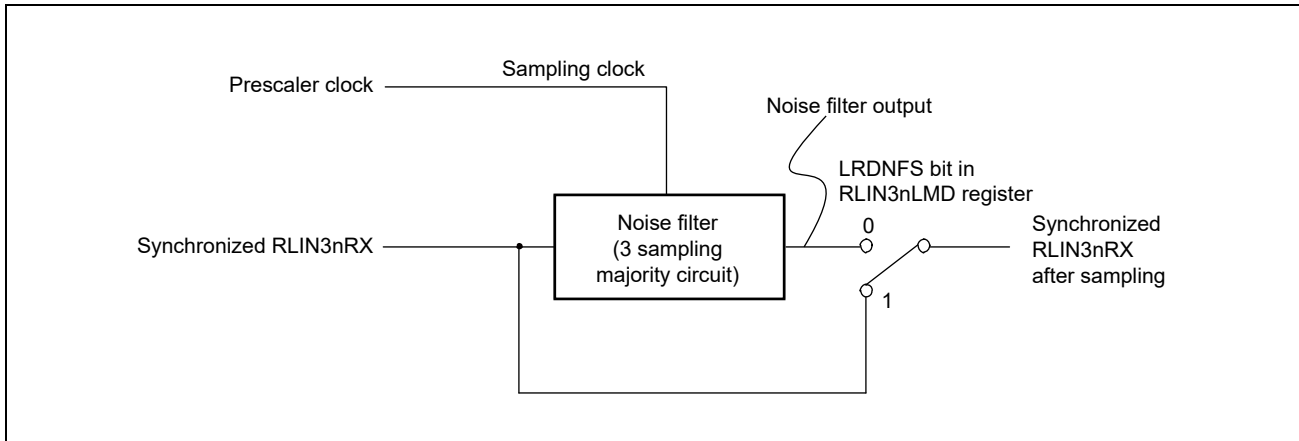


Figure 19.38 Configuration of Noise Filter

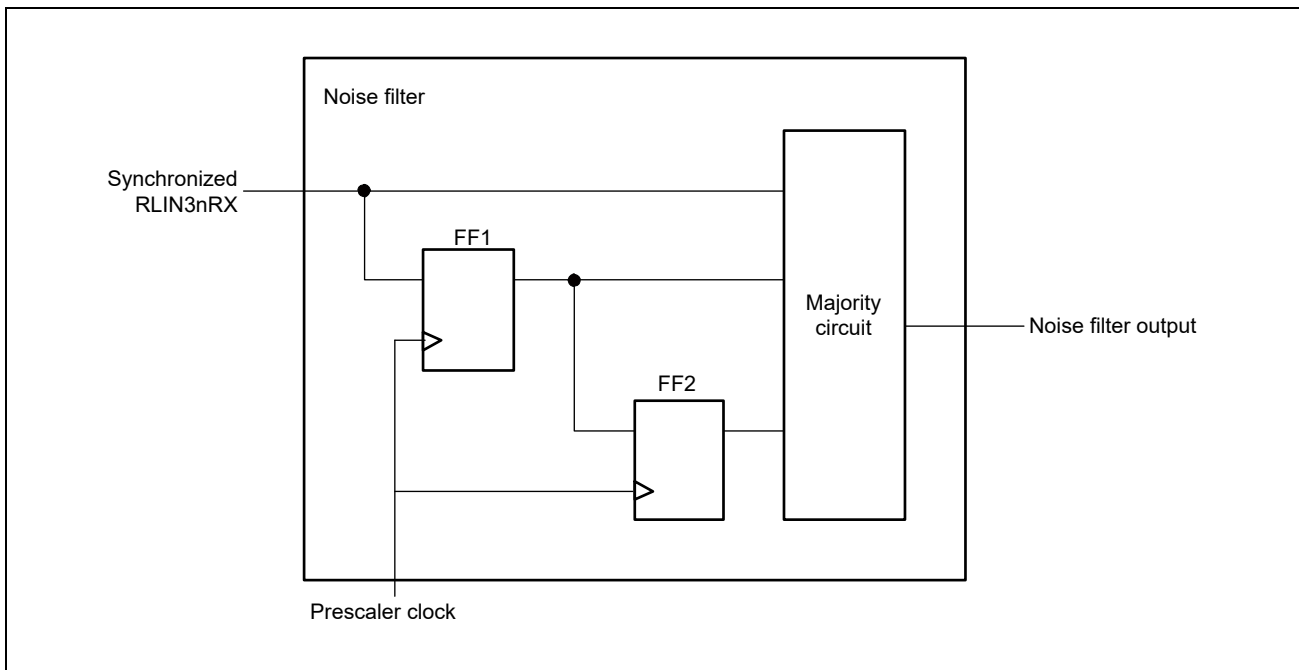


Figure 19.39 Example of Noise Filter Circuit

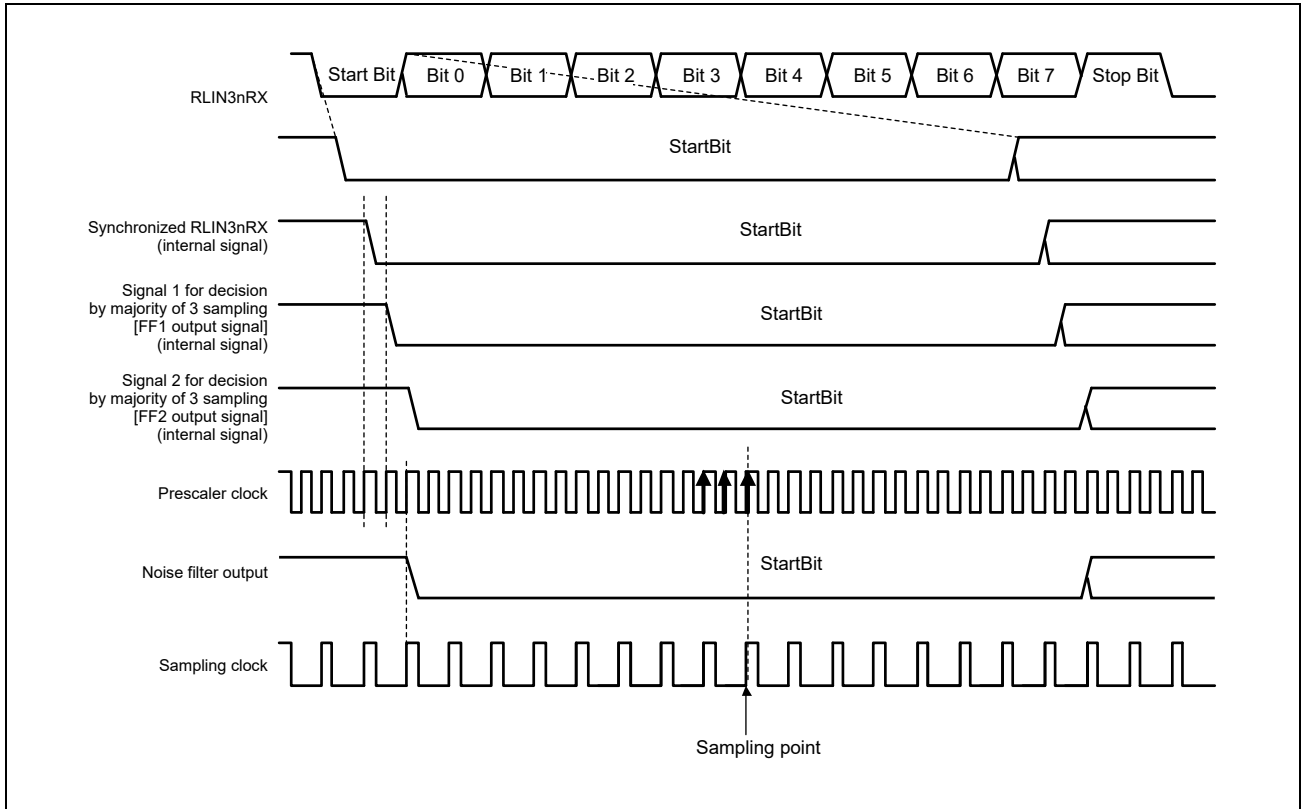


Figure 19.40 Determination of Received Data when Noise Filter is Used

Section 20 CANFD Interface (RS-CANFD)

20.1 Features of RS-CANFD

20.1.1 Units and Channels

This microcontroller has the following number of RS-CANFD units.

Table 20.1 Number of Units (for E2x-FCC1)

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Units	1	1
Name	RSCFDn (n = 0)	

Table 20.2 Number of Units (for E2M)

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Units	1	1
Name	RSCFDn (n = 0)	

The individual products have the CANFD interface channels listed below.

Table 20.3 Unit Configurations and Channels (for E2x-FCC1)

Unit Name	Channel Name	RH850/E2x-FCC1	
		373 Pins	292 Pins
RSCFDn	CAN0	✓	✓
	CAN1	✓	✓
	CAN2	✓	✓
	CAN3	✓	✓
	CAN4	✓	✓

Table 20.4 Unit Configurations and Channels (for E2M)

Unit Name	Channel Name	RH850/E2M	
		373 Pins	292 Pins
RSCFDn	CAN0	✓	✓
	CAN1	✓	✓
	CAN2	✓	✓
	CAN3	✓	✓
	CAN4	✓	✓

The RS-CANFD has two interface modes (classical CAN mode and CAN FD mode) and uses different registers for each mode. There are two types of register names RSCANnXXX and RSCFDnCFDXXX (XXX: arbitrary) depending on interface modes. When explaining specifications common to two registers, register names are described as RSCFDn(CFD)XXX.

Table 20.5 Indices

Index	Meaning
n	Throughout this section, the individual RS-CANFD units are generically indicated by the index "n".
m	Throughout this section, the individual channels of RS-CANFD units are generically indicated by the index "m".
j	The individual registers associated with receive rule table are generically indicated by the index "j". j = 0 to 15.
k	The individual transmit/receive FIFO buffers are generically indicated by the index "k". k = 0 to [channel m × 3 + 2];
x	The individual receive FIFO buffers are generically identified by the index "x". x = 0 to 7.
d	Data field registers of transmit/receive FIFO buffers and receive FIFO buffers are identified by "d". classical CAN mode: d = 0 or 1, CAN FD mode: d = 0 to 15.
q	The individual receive buffers are generically indicated by the index "q". q = 0 to [channel m × 16 + 15].
p	The individual transmit buffers are generically indicated by the index "p". p = 0 to [channel m × 16 + 15].
b	Data field registers of receive buffers and transmit buffers are identified by "b". classical CAN mode: b = 0 or 1, CAN FD mode: b = 0 to 4.
r	The individual RAM tests for CAN are generically indicated by the index "r". r = 0 to 63.
y	The registers not covered above are indicated by the letter "y".

Note: When writing a value to bits in the register that are outside of the index range on the product, write the value after reset to these bits.

The following table lists the values of indices for individual products.

Table 20.6 Indices for Individual Products

Indices for Individual Products	
E2x-FCC1	E2M
n = 0	n = 0
m = 0 to 4	m = 0 to 4
j = 0 to 15	j = 0 to 15
k = 0 to 14	k = 0 to 14
x = 0 to 7	x = 0 to 7
d = 0, 1* ¹ d = 0 to 15* ²	d = 0, 1* ¹ d = 0 to 15* ²
q = 0 to 79	q = 0 to 79
p = 0 to 79	p = 0 to 79
b = 0, 1* ¹ b = 0 to 4* ²	b = 0, 1* ¹ b = 0 to 4* ²
r = 0 to 63	r = 0 to 63
y = 0 to 2	y = 0 to 2

Note 1. In classical CAN mode

Note 2. In CAN FD mode

20.1.2 Register Base Address

The base addresses of registers in RSCFDn are described in the list below.

All Access to RSCFDn register addresses are given as address offsets to the base address.

Table 20.7 Register Base Address

Base Address Name	Base Address	Bus Group
<RSCFD0_base>	FFD0 0000 _H	Peripheral Group 3

20.1.3 Clock Supply

The clocks listed below are supplied to the RSCFDn.

Table 20.8 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
RSCFDn	clk_xincan	CLK_MOSC
	clkc	CLK_LSB
	pclk* ¹	CLK_HSB

Note 1. pclk must satisfy the following conditions so that no event of the CAN engine clock is overlooked.

- $pclk \geq 2 \times clkc$
- $pclk \geq 2 \times clk_xincan$

20.1.4 Interrupt Requests

The RSCFDn interrupt requests are listed in the following table.

Table 20.9 Interrupt Requests (for E2x-FCC1)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number
RSCFD0			
INTRCANGERR0	INTRCANGERR	RSCFD0 global error interrupt	379
INTRCANGRECC0	INTRCANGRECC	RSCFD0 receive FIFO interrupt	380
INTRCAN0ERR	INTRCANmERR (m = 0)	CAN0 error interrupt	381
INTRCAN0REC	INTRCANmREC (m = 0)	CAN0 transmit/receive FIFO receive completion interrupt	382
INTRCAN0TRX	INTRCANmTRX (m = 0)	CAN0 transmit interrupt	383
INTRCAN1ERR	INTRCANmERR (m = 1)	CAN1 error interrupt	384
INTRCAN1REC	INTRCANmREC (m = 1)	CAN1 transmit/receive FIFO receive completion interrupt	385
INTRCAN1TRX	INTRCANmTRX (m = 1)	CAN1 transmit interrupt	386
INTRCAN2ERR	INTRCANmERR (m = 2)	CAN2 error interrupt	387
INTRCAN2REC	INTRCANmREC (m = 2)	CAN2 transmit/receive FIFO receive completion interrupt	388
INTRCAN2TRX	INTRCANmTRX (m = 2)	CAN2 transmit interrupt	389
INTRCAN3ERR	INTRCANmERR (m = 3)	CAN3 error interrupt	390
INTRCAN3REC	INTRCANmREC (m = 3)	CAN3 transmit/receive FIFO receive completion interrupt	391
INTRCAN3TRX	INTRCANmTRX (m = 3)	CAN3 transmit interrupt	392
INTRCAN4ERR	INTRCANmERR (m = 4)	CAN4 error interrupt	393
INTRCAN4REC	INTRCANmREC (m = 4)	CAN4 transmit/receive FIFO receive completion interrupt	394
INTRCAN4TRX	INTRCANmTRX (m = 4)	CAN4 transmit interrupt	395

Table 20.10 Interrupt Requests (for E2M)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number
RSCFD0			
INTRCANGERR0	INTRCANGERR	RSCFD0 global error interrupt	379
INTRCANGRECC0	INTRCANGRECC	RSCFD0 receive FIFO interrupt	380
INTRCAN0ERR	INTRCANmERR (m = 0)	CAN0 error interrupt	381
INTRCAN0REC	INTRCANmREC (m = 0)	CAN0 transmit/receive FIFO receive completion interrupt	382
INTRCAN0TRX	INTRCANmTRX (m = 0)	CAN0 transmit interrupt	383
INTRCAN1ERR	INTRCANmERR (m = 1)	CAN1 error interrupt	384
INTRCAN1REC	INTRCANmREC (m = 1)	CAN1 transmit/receive FIFO receive completion interrupt	385
INTRCAN1TRX	INTRCANmTRX (m = 1)	CAN1 transmit interrupt	386
INTRCAN2ERR	INTRCANmERR (m = 2)	CAN2 error interrupt	387
INTRCAN2REC	INTRCANmREC (m = 2)	CAN2 transmit/receive FIFO receive completion interrupt	388
INTRCAN2TRX	INTRCANmTRX (m = 2)	CAN2 transmit interrupt	389
INTRCAN3ERR	INTRCANmERR (m = 3)	CAN3 error interrupt	390
INTRCAN3REC	INTRCANmREC (m = 3)	CAN3 transmit/receive FIFO receive completion interrupt	391
INTRCAN3TRX	INTRCANmTRX (m = 3)	CAN3 transmit interrupt	392
INTRCAN4ERR	INTRCANmERR (m = 4)	CAN4 error interrupt	393
INTRCAN4REC	INTRCANmREC (m = 4)	CAN4 transmit/receive FIFO receive completion interrupt	394
INTRCAN4TRX	INTRCANmTRX (m = 4)	CAN4 transmit interrupt	395

Table 20.11 DMA Trigger Numbers (for E2x-FCC1)

Interrupt Symbol Name	Unit Interrupt Signal	Description	sDMA Trigger Number	DTS Trigger Number
INTRCANRFDREQ7	can_rf_dmareq[7]	Receive FIFO DMA request 7	group0-131	group1-39
INTRCANRFDREQ6	can_rf_dmareq[6]	Receive FIFO DMA request 6	group0-130	group1-38
INTRCANRFDREQ5	can_rf_dmareq[5]	Receive FIFO DMA request 5	group0-129	group1-37
INTRCANRFDREQ4	can_rf_dmareq[4]	Receive FIFO DMA request 4	group0-128	group1-36
INTRCANRFDREQ3	can_rf_dmareq[3]	Receive FIFO DMA request 3	group0-127	group1-35
INTRCANRFDREQ2	can_rf_dmareq[2]	Receive FIFO DMA request 2	group0-126	group1-34
INTRCANRFDREQ1	can_rf_dmareq[1]	Receive FIFO DMA request 1	group0-125	group1-33
INTRCANRFDREQ0	can_rf_dmareq[0]	Receive FIFO DMA request 0	group0-124	group1-32
INTRCANCDFREQ4	can_cf_dmareq[4]	Common FIFO DMA request 4	group0-136	group1-44
INTRCANCDFREQ3	can_cf_dmareq[3]	Common FIFO DMA request 3	group0-135	group1-43
INTRCANCDFREQ2	can_cf_dmareq[2]	Common FIFO DMA request 2	group0-134	group1-42
INTRCANCDFREQ1	can_cf_dmareq[1]	Common FIFO DMA request 1	group0-133	group1-41
INTRCANCDFREQ0	can_cf_dmareq[0]	Common FIFO DMA request 0	group0-132	group1-40

Table 20.12 DMA Trigger Numbers (for E2M)

Interrupt Symbol Name	Unit Interrupt Signal	Description	sDMA Trigger Number	DTS Trigger Number
INTRCANRFDREQ7	can_rf_dmareq[7]	Receive FIFO DMA request 7	group0-131	group1-39
INTRCANRFDREQ6	can_rf_dmareq[6]	Receive FIFO DMA request 6	group0-130	group1-38
INTRCANRFDREQ5	can_rf_dmareq[5]	Receive FIFO DMA request 5	group0-129	group1-37
INTRCANRFDREQ4	can_rf_dmareq[4]	Receive FIFO DMA request 4	group0-128	group1-36
INTRCANRFDREQ3	can_rf_dmareq[3]	Receive FIFO DMA request 3	group0-127	group1-35
INTRCANRFDREQ2	can_rf_dmareq[2]	Receive FIFO DMA request 2	group0-126	group1-34
INTRCANRFDREQ1	can_rf_dmareq[1]	Receive FIFO DMA request 1	group0-125	group1-33
INTRCANRFDREQ0	can_rf_dmareq[0]	Receive FIFO DMA request 0	group0-124	group1-32
INTRCANCDFREQ4	can_cf_dmareq[4]	Common FIFO DMA request 4	group0-136	group1-44
INTRCANCDFREQ3	can_cf_dmareq[3]	Common FIFO DMA request 3	group0-135	group1-43
INTRCANCDFREQ2	can_cf_dmareq[2]	Common FIFO DMA request 2	group0-134	group1-42
INTRCANCDFREQ1	can_cf_dmareq[1]	Common FIFO DMA request 1	group0-133	group1-41
INTRCANCDFREQ0	can_cf_dmareq[0]	Common FIFO DMA request 0	group0-132	group1-40

20.1.5 Reset Sources

RSCFDn reset sources are listed in the following table. RSCFDn is initialized by these reset sources.

Table 20.13 Reset sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
RSCFDn	All registers	✓	✓	✓	✓	—	—

20.1.6 External Input/Output Signals

External input/output signals of RSCFDn are listed below.

Table 20.14 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
RSCFD0		
CANmRX	CANm receive data input	CRXm
CANmTX	CANm transmit data output	CTXm

CAUTION

This AC specification is only applied to the specific pin groups. For details, refer to Appendix file "*Limited_conditions_for_AC_specification.xlsx*".

20.2 Overview

20.2.1 Functional Outline

Table 20.15 shows the RS-CANFD module specifications. **Figure 20.1** shows the RS-CANFD module block diagram.

Table 20.15 RS-CANFD Module Specifications (1/3)

Item	Specification
Protocol	ISO11898-1 compliant Using CAN FD frames is selectable by switching interface modes.
Communication speed	<p>Classical CAN mode:</p> <ul style="list-style-type: none"> Up to 1 Mbps $\text{Communication speed (CANm bit time clock)} = \frac{1}{\text{CANm bit time}}$ $\text{CANm bit time} = \text{CANmTq} \times \text{Tq count per bit}$ $\text{CANmTq} = \frac{(\text{BRP}[9:0] \text{ bits in the RSCANnCmCFG register} + 1)}{f_{\text{CAN}}}$ <p>f_{CAN}: Frequency of CAN clock (selected by the DCS bit in the RSCANnGCFG register)</p> <p>CAN FD mode:</p> <ul style="list-style-type: none"> Nominal bit rate: up to 1 Mbps; data bit rate: up to 8 Mbps $\text{Transmission rate (CANm nominal bit time clock)} = \frac{1}{\text{CANm nominal bit time}}$ $\text{Transmission rate (CANm data bit time clock)} = \frac{1}{\text{CANm data bit time}}$ <p>CANm nominal bit time = CANmTq(N) × Tq count per nominal bit CANm data bit time = CANmTq(D) × Tq count per data a bit</p> $\text{CANmTq(N)} = \frac{(\text{NBRP}[9:0] \text{ bits in the RSCFDnCFDCmNCFG register} + 1)}{f_{\text{CAN}}}$ $\text{CANmTq(D)} = \frac{(\text{DBRP}[7:0] \text{ bits in the RSCFDnCFDCmDCFG register} + 1)}{f_{\text{CAN}}}$ <p>f_{CAN}: Frequency of CAN clock (selected by the DCS bit in the RSCFDnCFDGCFCFG register)</p> <p>m = see Table 20.6 Tq: Time quantum</p>
Buffer	<p>400 buffers in total</p> <ul style="list-style-type: none"> Individual buffers: 80 buffers (16 buffers × 5 channels) Transmit buffer: 16 buffers per channel Transmit queue: Single queue per channel (shared with the transmit buffer; up to 16 buffers allocatable) Shared buffers: 320 buffers for all channels Receive buffer: 0 to 80 buffers for all channels Receive FIFO buffer: 8 FIFO buffers for all channels (up to 128 buffers allocatable to each) Transmit/receive FIFO buffer: 3 FIFO buffers per channel (up to 128 buffers allocatable to each) ECC protected
Reception function	<ul style="list-style-type: none"> Receives data frames and remote frames. Selects ID format (standard ID, extended ID, or both IDs) to be received. Sets interrupt enable/disable for each FIFO. Mirror function (CAN node receives its own transmitted message.) Timestamp function (to record message reception time as a 16-bit timer value)

Table 20.15 RS-CANFD Module Specifications (2/3)

Item	Specification
Reception filter function	<ul style="list-style-type: none"> • Selects receive messages according to 320 receive rules. • Sets the number of receive rules (0 to 128) for each channel. • Acceptance filter processing: Sets ID and mask for each receive rule. • DLC filter processing: Sets DLC check value for each acceptance rule.
Receive message transfer function	<ul style="list-style-type: none"> • Routing function Transfer receive messages to arbitrary destinations (can be transferred to up to 8 buffers) Transfer destination: Receive buffer, receive FIFO buffer, and/or transmit/receive FIFO buffer • Label addition function Stores label information together with the message in a receive buffer and FIFO buffer.
Transmission function	<ul style="list-style-type: none"> • Transmits data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be transmitted. • Sets interrupt enable/disable for each transmit buffer and transmit/receive FIFO buffer. • Selects ID priority transmission or transmit buffer number priority transmission. • Transmit request can be aborted (possible to confirm with a flag) • One-shot transmission function
Interval transmission function	Transmit messages at configurable intervals (transmit mode or gateway mode of transmit/receive FIFO buffers)
Transmit queue function	Transmits all stored messages according to the ID priority.
Transmit history function	Stores the history information of transmitted messages. Adds timestamp (recording message transmission time as a 16-bit timer value) to the history information.
Gateway function	A received message is automatically routed to a different channel.
Bus off recovery mode selection	Selects the method for returning from bus off state. <ul style="list-style-type: none"> • ISO11898-1 compliant • Automatic entry to channel halt mode at bus-off entry • Automatic entry to channel halt mode at bus-off end • Entry to channel halt mode by program request • Transition to the error-active state by program request (forcible return from bus-off)
Error status monitoring	<ul style="list-style-type: none"> • Monitors CAN protocol errors (stuff error, form error, ACK error, CRC error, bit error, ACK delimiter error, and bus dominant lock). • Detects error status transitions (error warning, error passive, bus off entry, and bus off recovery) • Reads the error counter. • Monitors DLC errors.
Interrupt source	17 sources <ul style="list-style-type: none"> • Global interrupts (2 sources) Receive FIFO interrupt Global error interrupt • Channel interrupts (3 sources/channel) CANm transmit interrupt (m = see Table 20.6) <ul style="list-style-type: none"> – CANm transmit complete interrupt – CANm transmit abort interrupt – CANm transmit/receive FIFO transmit complete interrupt (in transmit mode, gateway mode) – CANm transmit history interrupt – CANm transmit queue interrupt CANm transmit/receive FIFO receive complete interrupt (in receive mode, gateway mode) CANm error interrupt
CAN stop mode	Reduces power consumption by stopping clock supply to the RS-CANFD module.
CAN communication clock	The source is selectable on-chip peripheral clock (clkc) or crystal clock (clk_xincan).

Table 20.15 RS-CANFD Module Specifications (3/3)

Item	Specification
Test function	Test function for user evaluation <ul style="list-style-type: none"> • Listen-only mode • Self-test mode 0 (external loopback) • Self-test mode 1 (internal loopback) • Restricted operation mode • RAM test (read/write test) • Inter-channel communication test [CRC error test enabled]

20.2.2 Interface Modes

The RS-CANFD has two interface modes.

- Classical CAN mode: Handles only classical CAN frames.
- CAN FD mode: Handles classical CAN frames and CAN FD frames.

These two modes use different register maps with the same base address. Register maps change by switching interface modes.

Interface modes are switched by the RCMC bit in the RSCFDn(CFD)GRMCFG register.

20.2.3 CAN FD Protocol

This product supports CAN FD according to the ISO 11898-1(2015) protocol.

20.2.4 Block Diagram

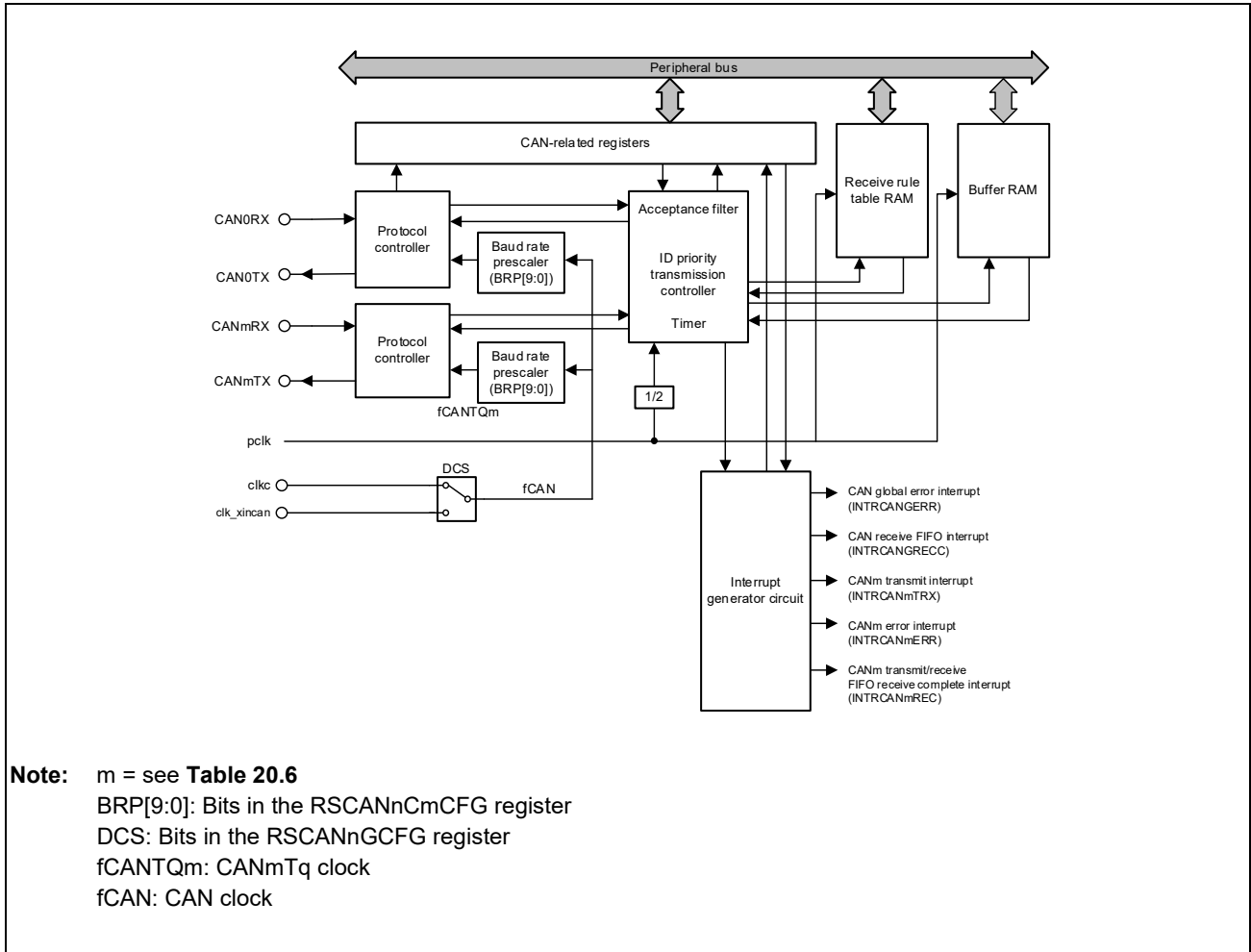


Figure 20.1 RS-CANFD Module Block Diagram

In CAN FD mode, different clock signals are input to the baud rate prescaler and the protocol controller respectively. See **Section 20.11.1.3, Communication Speed Setting**.

20.3 Registers (Classical CAN Mode)

20.3.1 List of Registers

The following tables list RS-CANFD registers to be used in classical CAN mode.

For details about <RSCFDn_base>, see **Section 20.1.2, Register Base Address**.

For details about registers initialized in Global reset mode or Channel reset mode, see following.

- **Table 20.186, Registers Initialized in Global Reset Mode or Channel Reset Mode**
- **Table 20.187, Registers Initialized Only in Global Reset Mode**

Table 20.16 List of Registers (1/4)

Module Name	Register Name	Symbol	Address	Access	Access Protection
Interface Mode-Related Registers					
RSCANn	Global Interface Mode Select Register	RSCANnGRMCFG	<RSCFDn_base> + 04FC _H	8,16,32	—
Channel-Related Registers					
RSCANn	Channel m Configuration Register	RSCANnCmCFG	<RSCFDn_base> + 0000 _H + (10 _H × m)	8,16,32	—
RSCANn	Channel m Control Register	RSCANnCmCTR	<RSCFDn_base> + 0004 _H + (10 _H × m)	8,16,32	—
RSCANn	Channel m Status Register	RSCANnCmSTS	<RSCFDn_base> + 0008 _H + (10 _H × m)	8,16,32	—
RSCANn	Channel m Error Flag Register	RSCANnCmERFL	<RSCFDn_base> + 000C _H + (10 _H × m)	8,16,32	—
Global-Related Registers					
RSCANn	Global Configuration Register	RSCANnGCFG	<RSCFDn_base> + 0084 _H	8,16,32	—
RSCANn	Global Control Register	RSCANnGCTR	<RSCFDn_base> + 0088 _H	8,16,32	—
RSCANn	Global Status Register	RSCANnGSTS	<RSCFDn_base> + 008C _H	8,16,32	—
RSCANn	Global Error Flag Register	RSCANnGERFL	<RSCFDn_base> + 0090 _H	8,16,32	—
RSCANn	Global Timestamp Counter Register	RSCANnGTSC	<RSCFDn_base> + 0094 _H	16,32	—
RSCANn	Global TX Interrupt Status Register 0	RSCANnGTINTSTS0	<RSCFDn_base> + 0460 _H	8,16,32	—
RSCANn	Global TX Interrupt Status Register 1	RSCANnGTINTSTS1	<RSCFDn_base> + 0464 _H	8,16,32	—
RSCANn	Global FD Configuration Register	RSCANnGFDCFG	<RSCFDn_base> + 0474 _H	8,16,32	—
Receive Rule-Related Registers					
RSCANn	Receive Rule Entry Control Register	RSCANnGAFLECTR	<RSCFDn_base> + 0098 _H	8,16,32	—
RSCANn	Receive Rule Configuration Register 0	RSCANnGAFLCFG0	<RSCFDn_base> + 009C _H	8,16,32	—
RSCANn	Receive Rule Configuration Register 1	RSCANnGAFLCFG1	<RSCFDn_base> + 00A0 _H	8,16,32	—
RSCANn	Receive Rule ID Register j	RSCANnGAFLIDj	<RSCFDn_base> + 0500 _H + (10 _H × j)	8,16,32	—
RSCANn	Receive Rule Mask Register j	RSCANnGAFLMj	<RSCFDn_base> + 0504 _H + (10 _H × j)	8,16,32	—
RSCANn	Receive Rule Pointer 0 Register j	RSCANnGAFLP0_j	<RSCFDn_base> + 0508 _H + (10 _H × j)	8,16,32	—
RSCANn	Receive Rule Pointer 1 Register j	RSCANnGAFLP1_j	<RSCFDn_base> + 050C _H + (10 _H × j)	8,16,32	—

Table 20.16 List of Registers (2/4)

Module Name	Register Name	Symbol	Address	Access	Access Protection
Receive Buffer-Related Registers					
RSCANn	Receive Buffer Number Register	RSCANnRMNB	<RSCFDn_base> + 00A4 _H	8,16,32	—
RSCANn	Receive Buffer New Data Register y	RSCANnRMNDy	<RSCFDn_base> + 00A8 _H + (04 _H × y)	8,16,32	—
RSCANn	Receive Buffer ID Register q	RSCANnRMIDq	<RSCFDn_base> + 0600 _H + (10 _H × q)	8,16,32	—
RSCANn	Receive Buffer Pointer Register q	RSCANnRMPTRq	<RSCFDn_base> + 0604 _H + (10 _H × q)	8,16,32	—
RSCANn	Receive Buffer Data Field 0 Register q	RSCANnRMDf0_q	<RSCFDn_base> + 0608 _H + (10 _H × q)	8,16,32	—
RSCANn	Receive Buffer Data Field 1 Register q	RSCANnRMDf1_q	<RSCFDn_base> + 060C _H + (10 _H × q)	8,16,32	—
Receive FIFO Buffer-Related Registers					
RSCANn	Receive FIFO Buffer Configuration and Control Register x	RSCANnRFCCx	<RSCFDn_base> + 00B8 _H + (04 _H × x)	8,16,32	—
RSCANn	Receive FIFO Buffer Status Register x	RSCANnRFSTx	<RSCFDn_base> + 00D8 _H + (04 _H × x)	8,16,32	—
RSCANn	Receive FIFO Buffer Pointer Control Register x	RSCANnRFPCTRx	<RSCFDn_base> + 00F8 _H + (04 _H × x)	8,16,32	—
RSCFDn	Receive FIFO Buffer Access ID Register x	RSCANnRFIDx	<RSCFDn_base> + 0E00 _H + (10 _H × x)	8,16,32	—
RSCANn	Receive FIFO Buffer Access Pointer Register x	RSCANnRFPTRx	<RSCFDn_base> + 0E04 _H + (10 _H × x)	8,16,32	—
RSCANn	Receive FIFO Buffer Access Data Field 0 Register x	RSCANnRFDF0_x	<RSCFDn_base> + 0E08 _H + (10 _H × x)	8,16,32	—
RSCANn	Receive FIFO Buffer Access Data Field 1 Register x	RSCANnRFDF1_x	<RSCFDn_base> + 0E0C _H + (10 _H × x)	8,16,32	—
Transmit/Receive FIFO Buffer-Related Registers					
RSCANn	Transmit/Receive FIFO Buffer Configuration and Control Register k	RSCANnCFCCk	<RSCFDn_base> + 0118 _H + (04 _H × k)	8,16,32	—
RSCANn	Transmit/Receive FIFO Buffer Status Register k	RSCANnCFSTk	<RSCFDn_base> + 0178 _H + (04 _H × k)	8,16,32	—
RSCANn	Transmit/Receive FIFO Buffer Pointer Control Register k	RSCANnCFPCTRk	<RSCFDn_base> + 01D8 _H + (04 _H × k)	8,16,32	—
RSCANn	Transmit/Receive FIFO Buffer Access ID Register k	RSCANnCFIDk	<RSCFDn_base> + 0E80 _H + (10 _H × k)	8,16,32	—
RSCANn	Transmit/Receive FIFO Buffer Access Pointer Register k	RSCANnCFPTRk	<RSCFDn_base> + 0E84 _H + (10 _H × k)	8,16,32	—
RSCANn	Transmit/Receive FIFO Buffer Access Data Field 0 Register k	RSCANnCFDF0_k	<RSCFDn_base> + 0E88 _H + (10 _H × k)	8,16,32	—
RSCANn	Transmit/Receive FIFO Buffer Access Data Field 1 Register k	RSCANnCFDF1_k	<RSCFDn_base> + 0E8C _H + (10 _H × k)	8,16,32	—

Table 20.16 List of Registers (3/4)

Module Name	Register Name	Symbol	Address	Access	Access Protection
FIFO Status-Related Registers					
RSCANn	FIFO Empty Status Register	RSCANnFESTS	<RSCFDn_base> + 0238 _H	8,16,32	—
RSCANn	FIFO Full Status Register	RSCANnFFSTS	<RSCFDn_base> + 023C _H	8,16,32	—
RSCANn	FIFO Message Lost Status Register	RSCANnFMSTS	<RSCFDn_base> + 0240 _H	8,16,32	—
RSCANn	Receive FIFO Buffer Interrupt Flag Status Register	RSCANnRFISTS	<RSCFDn_base> + 0244 _H	8,16,32	—
RSCANn	Transmit/Receive FIFO Buffer Receive Interrupt Flag Status Register	RSCANnCFRISTS	<RSCFDn_base> + 0248 _H	8,16,32	—
RSCANn	Transmit/Receive FIFO Buffer Transmit Interrupt Flag Status Register	RSCANnCFTISTS	<RSCFDn_base> + 024C _H	8,16,32	—
Transmit Buffer-Related Registers					
RSCANn	Transmit Buffer Control Register p	RSCANnTMCp	<RSCFDn_base> + 0250 _H + (01 _H × p)	8	—
RSCANn	Transmit Buffer Status Register p	RSCANnTMSTSp	<RSCFDn_base> + 02D0 _H + (01 _H × p)	8	—
RSCANn	Transmit Buffer ID Register p	RSCANnTMIDp	<RSCFDn_base> + 1000 _H + (10 _H × p)	8,16,32	—
RSCANn	Transmit Buffer Pointer Register p	RSCANnTMPTRp	<RSCFDn_base> + 1004 _H + (10 _H × p)	8,16,32	—
RSCANn	Transmit Buffer Data Field 0 Register p	RSCANnTMDf0_p	<RSCFDn_base> + 1008 _H + (10 _H × p)	8,16,32	—
RSCANn	Transmit Buffer Data Field 1 Register p	RSCANnTMDf1_p	<RSCFDn_base> + 100C _H + (10 _H × p)	8,16,32	—
RSCANn	Transmit Buffer Interrupt Enable Configuration Register y	RSCANnTMIECy	<RSCFDn_base> + 0390 _H + (04 _H × y)	8,16,32	—
Transmit Buffer Status-Related Registers					
RSCANn	Transmit Buffer Transmit Request Status Register y	RSCANnTMTRSTSy	<RSCFDn_base> + 0350 _H + (04 _H × y)	8,16,32	—
RSCANn	Transmit Buffer Transmit Abort Request Status Register y	RSCANnTMTARSTSy	<RSCFDn_base> + 0360 _H + (04 _H × y)	8,16,32	—
RSCANn	Transmit Buffer Transmit Complete Status Register y	RSCANnTMTcSTSy	<RSCFDn_base> + 0370 _H + (04 _H × y)	8,16,32	—
RSCANn	Transmit Buffer Transmit Abort Status Register y	RSCANnTMTASTSy	<RSCFDn_base> + 0380 _H + (04 _H × y)	8,16,32	—
Transmit Queue-Related Registers					
RSCANn	Transmit Queue Configuration and Control Register m	RSCANnTXQCCm	<RSCFDn_base> + 03A0 _H + (04 _H × m)	8,16,32	—
RSCANn	Transmit Queue Status Register m	RSCANnTXQSTSm	<RSCFDn_base> + 03C0 _H + (04 _H × m)	8,16,32	—
RSCANn	Transmit Queue Pointer Control Register m	RSCANnTXQPCTrm	<RSCFDn_base> + 03E0 _H + (04 _H × m)	8,16,32	—
Transmit History-Related Registers					
RSCANn	Transmit History List Configuration and Control Register m	RSCANnTHLCCm	<RSCFDn_base> + 0400 _H + (04 _H × m)	8,16,32	—
RSCANn	Transmit History List Status Register m	RSCANnTHLSTSm	<RSCFDn_base> + 0420 _H + (04 _H × m)	8,16,32	—
RSCANn	Transmit History List Pointer Control Register m	RSCANnTHLPCTrm	<RSCFDn_base> + 0440 _H + (04 _H × m)	8,16,32	—
RSCANn	Transmit History List Access Register m	RSCANnTHLACCm	<RSCFDn_base> + 1800 _H + (04 _H × m)	8,16,32	—

Table 20.16 List of Registers (4/4)

Module Name	Register Name	Symbol	Address	Access	Access Protection
Test-Related Registers					
RSCANn	Global Test Configuration Register	RSCANnGTSTCFG	<RSCFDn_base> + 0468 _H	8,16,32	—
RSCANn	Global Test Control Register	RSCANnGTSTCTR	<RSCFDn_base> + 046C _H	8,16,32	RSCANn GLOCKK
RSCANn	Global Lock Key Register	RSCANnGLOCKK	<RSCFDn_base> + 047C _H	16,32	—
RSCANn	RAM Test Page Access Register r	RSCANnRPGACCr	<RSCFDn_base> + 1900 _H + (04 _H × r)	8,16,32	RSCANn CFGTS TCTR

Table 20.17 Transmit Buffer p Assigned for Each Channel

	CANm
Transmit buffer p	Transmit buffer 16 × m + 0
	Transmit buffer 16 × m + 1
	Transmit buffer 16 × m + 2
	Transmit buffer 16 × m + 3
	Transmit buffer 16 × m + 4
	Transmit buffer 16 × m + 5
	Transmit buffer 16 × m + 6
	Transmit buffer 16 × m + 7
	Transmit buffer 16 × m + 8
	Transmit buffer 16 × m + 9
	Transmit buffer 16 × m + 10
	Transmit buffer 16 × m + 11
	Transmit buffer 16 × m + 12
	Transmit buffer 16 × m + 13
	Transmit buffer 16 × m + 14
	Transmit buffer 16 × m + 15

Table 20.18 Transmit/Receive FIFO Buffer k Assigned for Each Channel

	CANm
Transmit/receive FIFO buffer k	Transmit/receive FIFO buffer 3 × m + 0
	Transmit/receive FIFO buffer 3 × m + 1
	Transmit/receive FIFO buffer 3 × m + 2

Table 20.19 Transmit Buffer p Linked to Transmit/Receive FIFO Buffer by the Set Value of CFTML[3:0] Bits

Set value of CFTML[3:0] bit	Transmit buffer p linked to transmit/receive FIFO buffer
0000 _B	Transmit buffer $16 \times m + 0$
0001 _B	Transmit buffer $16 \times m + 1$
0010 _B	Transmit buffer $16 \times m + 2$
0011 _B	Transmit buffer $16 \times m + 3$
0100 _B	Transmit buffer $16 \times m + 4$
0101 _B	Transmit buffer $16 \times m + 5$
0110 _B	Transmit buffer $16 \times m + 6$
0111 _B	Transmit buffer $16 \times m + 7$
1000 _B	Transmit buffer $16 \times m + 8$
1001 _B	Transmit buffer $16 \times m + 9$
1010 _B	Transmit buffer $16 \times m + 10$
1011 _B	Transmit buffer $16 \times m + 11$
1100 _B	Transmit buffer $16 \times m + 12$
1101 _B	Transmit buffer $16 \times m + 13$
1110 _B	Transmit buffer $16 \times m + 14$
1111 _B	Transmit buffer $16 \times m + 15$

Table 20.20 Transmit Buffer p Assigned for Transmit Queue for Each Channel

Set Value of TXQDC[3:0] Bits	Transmit Buffer p Assigned for Transmit Queue
0000 _B	Setting prohibited
0001 _B	Setting prohibited
0010 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 13$
0011 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 12$
0100 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 11$
0101 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 10$
0110 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 9$
0111 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 8$
1000 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 7$
1001 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 6$
1010 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 5$
1011 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 4$
1100 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 3$
1101 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 2$
1110 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 1$
1111 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 0$

20.3.2 Details of Interface Mode-Related Registers

20.3.2.1 RSCANnGRMCFG — Global Interface Mode Select Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCMC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 20.21 RSCANnGRMCFG Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	RCMC	Interface Mode Select 0: Classical CAN mode 1: CAN FD mode

Note 1. The RSCANnGRMCFG register and the RSCFDnCFDGRMCFG register are the same register. Therefore, set either one of the registers.

Modify the RSCANnGRMCFG register only in global reset mode. Before setting other RS-CANFD registers, set this register.

RCMC Bit

Setting this bit to 0 makes classical CAN mode available.

Setting this bit to 1 causes the RS-CANFD module to transition to CAN FD mode.

To switch the RS-CANFD module from CAN FD mode to classical CAN mode, set the values after reset to all respective registers and bits allocated only to the register map in CAN FD mode, and then modify the value of RSCANnGRMCFG register.

20.3.3 Details of Channel-Related Registers

20.3.3.1 RSCANnCmCFG — Channel m Configuration Register

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	—	—	—	SJW[1:0]		—	TSEG2[2:0]			TSEG1[3:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	—	—	—	—	BRP[9:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 20.22 RSCANnCmCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
25, 24	SJW [1:0]	Resynchronization Jump Width Control b25 b24 0 0: 1 Tq 0 1: 2 Tq 1 0: 3 Tq 1 1: 4 Tq
23	—	Reserved When read, the value after reset is returned. When writing to this bit, write the value after reset.
22 to 20	TSEG2 [2:0]	Time Segment 2 Control b22 b21 b20 0 0 0: Setting is prohibited for channels for CAN communications. 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq

Table 20.22 RSCANnCmCFG Register Contents (2/2)

Bit Position	Bit Name	Function
19 to 16	TSEG1 [3:0]	Time Segment 1 Control b19 b18 b17 b16 0 0 0 0: Setting is prohibited for channels for CAN communications. 0 0 0 1: Setting is prohibited for channels for CAN communications. 0 0 1 0: Setting is prohibited for channels for CAN communications. 0 0 1 1: 4Tq 0 1 0 0: 5Tq 0 1 0 1: 6Tq 0 1 1 0: 7Tq 0 1 1 1: 8Tq 1 0 0 0: 9Tq 1 0 0 1: 10Tq 1 0 1 0: 11Tq 1 0 1 1: 12Tq 1 1 0 0: 13Tq 1 1 0 1: 14Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq
15 to 10	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
9 to 0	BRP [9:0]	Prescaler Division Ratio Setting When these bits are set to P (0 to 1023), the baud rate prescaler divides fCAN by P + 1.

Modify the RSCANnCmCFG register in channel reset mode or channel halt mode. Set this register in channel reset mode before transitioning to channel communication mode or channel halt mode. For a description of the bit timing parameters and settings, see **Section 20.11.1, Initial Settings**.

SJW[1:0] Bits

These bits are used to specify a Tq value for the resynchronization jump width. Allowed values are 1 Tq to 4 Tq, inclusive.

Set a value less than or equal to the value of the TSEG2 bits.

TSEG2[2:0] Bits

These bits are used to specify a Tq value for the length of phase segment 2 (PHASE_SEG2).

Allowed values are 2 Tq to 8 Tq, inclusive.

Set a value smaller than the value of the TSEG1 bits.

TSEG1[3:0] Bits

These bits are used to specify a Tq value for the total length of the propagation segment (PROP_SEG) and phase segment 1 (PHASE_SEG1).

Allowed values are 4 Tq to 16 Tq, inclusive.

BRP[9:0] Bits

The CANmTq clock (fCANTQm) is calculated by dividing the CAN clock (fCAN) by the baud rate prescaler ((BRP[9:0]) + 1). One clock cycle of the CANmTq clock is 1 Time Quantum (Tq).

20.3.3.2 RSCANnCMCTR — Channel m Control Register

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0005_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CRCT	—	—	—	CTMS[1:0]	CTME	ERRD	BOM[1:0]	—	—	—	—	—	—	TAIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLPR	CHMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 20.23 RSCANnCMCTR Register Contents (1/2)

Bit Position	Bit Name	Function
31	—	Reserved When read, the value after reset is returned. When writing to this bit, write the value after reset.
30	CRCT	CRC Error Test Enable 0: The first bit of the reception ID field is not inverted. 1: The first bit of the reception ID field is inverted.
29 to 27	—	Reserved When read, the value after reset is returned. When writing to this bit, write the value after reset.
26, 25	CTMS[1:0]	Communication Test Mode Select b26 b25 0 0: Standard test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback mode) 1 1: Self-test mode 1 (internal loopback mode)
24	CTME	Communication Test Mode Enable 0: Communication test mode is disabled. 1: Communication test mode is enabled.
23	ERRD	Error Display Mode Select 0: Only the first error information is displayed after bit 14 to 8 in the RSCANnCMERFL register are all cleared. 1: All error information is displayed.
22, 21	BOM[1:0]	Bus Off Recovery Mode Select b22 b21 0 0: ISO11898-1 compliant 0 1: Transition to channel halt mode at bus-off entry 1 0: Transition to channel halt mode at bus-off end 1 1: Transition to channel halt mode (during bus-off recovery period) by a program request
20 to 17	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
16	TAIE	Transmit Abort Interrupt Enable 0: Transmit abort interrupt is disabled. 1: Transmit abort interrupt is enabled.

Table 20.23 RSCANnCMCTR Register Contents (2/2)

Bit Position	Bit Name	Function
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.
14	BLIE	Bus Lock Interrupt Enable 0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled.
13	OLIE	Overload Frame Transmit Interrupt Enable 0: Overload frame transmit interrupt is disabled. 1: Overload frame transmit interrupt is enabled.
12	BORIE	Bus Off Recovery Interrupt Enable 0: Bus off recovery interrupt is disabled. 1: Bus off recovery interrupt is enabled.
11	BOEIE	Bus Off Entry Interrupt Enable 0: Bus off entry interrupt is disabled. 1: Bus off entry interrupt is enabled.
10	EPIE	Error Passive Interrupt Enable 0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled.
9	EWIE	Error Warning Interrupt Enable 0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled.
8	BEIE	Bus Error Interrupt Enable 0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled.
7 to 4	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	RTBO	Forcible Return from Bus-off When this bit is set to 1, forcible return from the bus off state is made. This bit is always read as 0.
2	CSLPR	Channel Stop Mode 0: Other than channel stop mode 1: Channel stop mode
1, 0	CHMDC[1:0]	Mode Select b1 b0 0 0: Channel communication mode 0 1: Channel reset mode 1 0: Channel halt mode 1 1: Setting prohibited

CRCT Bit

This bit is used to test the CRC generation circuit in the RS-CANFD module. Setting this bit to 1 inverts the first bit of the ID field when a message is received. With this inversion of bit, the CRC calculation result does not match the normal CRC value of the received frame, which can detect a CRC error (the CERR bit in the RSCANnCMERFL register is 1).

When using this function, note the following.

- This function is available while the CTME bit in the RSCANnCMCTR register is 1 (communication test mode enabled).

- This function cannot communicate with other CAN nodes. Use this function for inter-channel communication test (the CmICBCE bit in the RSCANnGTSTCFG register is 1).
- Bit inversion in the ID field may cause bit stuffing rule violation. In that case, no CRC error is detected but a stuff error is detected.

Modify this bit only in channel halt mode. This bit is always 0 in channel reset mode.

CTMS[1:0] Bits

These bits are used to select a communication test mode. Modify these bits in channel halt mode only. These bits are set to 0 in channel reset mode.

CTME Bit

Setting this bit to 1 enables communication test mode. Modify this bit in channel halt mode. This bit is set to 0 in channel reset mode.

ERRD Bit

This bit is used to control display mode of bits 14 to 8 in the RSCANnCmERFL register.

When this bit is cleared to 0, if any error is detected while the flags of bits 14-8 in the RSCANnCmERFL register are all 0, only the flags of the first error event are set to 1. If two or more errors occur in first error event, all the flags of the detected errors are set to 1.

When this bit is set to 1, all the flags of errors that have occurred are set to 1 regardless of the error occurrence order.

Modify this bit only in channel reset mode or channel halt mode.

BOM[1:0] Bits

These bits are used to select a bus off recovery mode of the RS-CANFD module.

When the BOM[1:0] bits are set to 00_B, return from the bus off state is compliant with the CAN specifications. That is, the RS-CANFD module reenters the CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. A bus off recovery interrupt request is generated at the time of return from the bus off state. Even if the CHMDC[1:0] bits are set to 10_B (channel halt mode) before recessive bits are detected 128 times, the RS-CANFD module does not transition to channel halt mode until recessive bits are detected 128 times.

When the RS-CANFD module reaches the bus off state when the BOM[1:0] bits are set to 01_B, the CHMDC[1:0] bits in the RSCANnCmCTR register (m = see **Table 20.6**) are set to 10_B and the RS-CANFD module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits in the RSCANnCmSTS register are cleared to 00_H.

When the RS-CANFD module reaches the bus off state when the BOM[1:0] bits are set to 10_B, the CHMDC[1:0] bits are set to 10_B and the RS-CANFD module transitions to channel halt mode after return from the bus off state (11 consecutive recessive bits are detected 128 times). A bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H.

When the BOM[1:0] bits are set to 11_B and the CHMDC[1:0] bits are set to 10_B while the RS-CANFD module is in the bus off state, the RS-CANFD module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H. However, if 11 consecutive recessive bits are detected 128 times and the RS-CANFD module has recovered to the error active state from the bus off state before the CHMDC[1:0] bits are set to 10_B, a bus off recovery interrupt request is generated.

When the RS-CANFD module transitions to channel halt mode (at bus off entry when the BOM[1:0] bits are 01_B or at bus off end when the BOM[1:0] bits are 10_B), if the CHMDC[1:0] bits are written by a program at the same time, writing by a program takes precedence. Modify BOM[1:0] bits in channel reset mode.

TAIE Bit

When transmit abort of the transmit buffer is completed with the TAIE bit set to 1, an interrupt request is generated. Modify this bit only in channel reset mode.

ALIE Bit

When the ALF flag in the RSCANnCMERFL register is set to 1 with the ALIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BLIE Bit

When the BLF flag in the RSCANnCMERFL register is set to 1 with the BLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

OLIE Bit

When the OVLF flag in the RSCANnCMERFL register is set to 1 with the OLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BORIE Bit

When the BORF flag in the RSCANnCMERFL register is set to 1 with the BORIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BOEIE Bit

When the BOEF flag in the RSCANnCMERFL register is set to 1 with the BOEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EPIE Bit

When the EPF flag in the RSCANnCMERFL register is set to 1 with the EPIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EWIE Bit

When the EWF flag in the RSCANnCMERFL register is set to 1 with the EWIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BEIE Bit

When the BEF flag in the RSCANnCMERFL register is set to 1 with the BEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

RTBO Bit

Setting this bit to 1 (forcible return from the bus off state) in the bus off state changes the state forcibly from the bus off state to the error active state. This bit is automatically cleared to 0. Setting this bit to 1 clears the TEC[7:0] and REC[7:0] bits in the RSCANnCMSTS register to 00_H and also clears the BOSTS flag in the RSCANnCMSTS register to 0 (not in bus off state). The other registers remain unchanged. No bus off recovery interrupt request due to return from the bus off state is generated. Use this bit only when the BOM[1:0] bits in the RSCANnCMCTR register are 00_B (ISO11898-1 compliant).

A delay of up to 1 CAN bit time occurs after the RTBO bit is set to 1 until the RS-CANFD module transitions to the error active state. Set this bit to 1 in channel communication mode.

CSLPR Bit

Setting this bit to 1 places the channel in channel stop mode.

Clearing this bit to 0 makes the channel leave from channel stop mode.

Do not rewrite this bit in channel communication mode or in channel halt mode.

CHMDC[1:0] Bits

These bits are used to select a channel mode (channel communication mode, channel reset mode, or channel halt mode). For details, see **Section 20.6.2, Channel Modes**. Setting the CSLPR bit to 1 in channel reset mode allows transition to channel stop mode. Do not set these bits to 11_B. When the RS-CANFD module has transitioned to channel halt mode by the setting of the BOM[1:0] bits, the CHMDC[1:0] bits are automatically set to 10_B.

20.3.3.3 RSCANnCmSTS — Channel m Status Register

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0005_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEC[7:0]								REC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	COM STS	REC STS	TRM STS	BOSTS	EPSTS	CSLP STS	CHLT STS	CRST STS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.24 RSCANnCmSTS Register Contents

Bit Position	Bit Name	Function
31 to 24	TEC [7:0]	The transmit error counter (TEC) can be read.
23 to 16	REC [7:0]	The receive error counter (REC) can be read.
15 to 8	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
7	COMSTS	Communication Status Flag 0: Communication is not ready. 1: Communication is ready.
6	RECSTS	Receive Status Flag 0: Bus idle, in transmission, or bus off state 1: In reception
5	TRMSTS	Transmit Status Flag 0: Bus idle or in reception 1: In transmission or bus off state
4	BOSTS	Bus Off Status Flag 0: Not in bus off state 1: In bus off state
3	EPSTS	Error Passive Status Flag 0: Not in error passive state 1: In error passive state
2	CSLPSTS	Channel Stop Status Flag 0: Not in channel stop mode 1: In channel stop mode
1	CHLTSTS	Channel Halt Status Flag 0: Not in channel halt mode 1: In channel halt mode
0	CRSTSTS	Channel Reset Status Flag 0: Not in channel reset mode 1: In channel reset mode

TEC[7:0] Bits

These bits indicate the transmit error counter value. For transmit error counter increment/decrement conditions, see *the CAN specification (ISO11898-1)*.

These bits are cleared to 0 in channel reset mode.

REC[7:0] Bits

These bits indicate the receive error counter value. For receive error counter increment/decrement conditions, see *the CAN specification (ISO11898-1)*.

These bits are cleared to 0 in channel reset mode.

COMSTS Flag

This flag indicates that communication is ready.

This flag is set to 1 when the RS-CANFD module has detected 11 consecutive recessive bits after it has transitioned from channel reset mode or channel halt mode to channel communication mode. This flag is cleared to 0 in channel reset mode or channel halt mode.

RECSTS Flag

This flag is set to 1 when reception has started, and is cleared to 0 when the bus has become idle or transmission has started.

TRMSTS Flag

This flag is set to 1 when transmission has started, and is cleared to 0 when the bus has become idle or reception has started. This flag remains 1 in the bus off state.

BOSTS Flag

This flag is set to 1 when the TEC[7:0] value exceeds 255 and the RS-CANFD module has entered the bus off state (TEC[7:0] value > 255), and is cleared to 0 when the RS-CANFD module has exited the bus off state.

EPSTS Flag

This flag is set to 1 when the TEC[7:0] or REC[7:0] value exceeds 127 and the RS-CANFD module has entered the error passive state ($128 \leq \text{TEC}[7:0] \text{ value} \leq 255$ or $128 \leq \text{REC}[7:0] \text{ value}$), and is cleared to 0 when the RS-CANFD module has exited the error passive state or has entered channel reset mode.

CSLPSTS Flag

This flag is set to 1 when the RS-CANFD module has transitioned to channel stop mode, and is cleared to 0 when the RS-CANFD module has returned from channel stop mode.

CHLTSTS Flag

This flag is set to 1 when the RS-CANFD module has transitioned to channel halt mode, and is cleared to 0 when the RS-CANFD module has exited channel halt mode.

CRSTSTS Flag

This flag is set to 1 when the RS-CANFD module has transitioned to channel reset mode, and is cleared to 0 when the RS-CANFD module has transitioned to channel communication mode or channel halt mode. This flag remains 1 if the RS-CANFD module transitions from channel reset mode to channel stop mode.

20.3.3.4 RSCANnCmERFL — Channel m Error Flag Register

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CRCREG[14:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLf	BORF	BOEF	EPF	EWf	BEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Note 1. The only effective value for writing to this flag bit is 0, which clears the status. Any other writing to the bit results in retention of the status and does not change the value.

Table 20.25 RSCANnCmERFL Register Contents (1/2)

Bit Position	Bit Name	Function
31	—	Reserved When read, the value after reset is returned. When writing to this bit, write the value after reset.
30 to 16	CRCREG[14:0]	CRC Calculation Data A CRC value calculated based on the transmitted message or received message is indicated.
15	—	Reserved When read, the value after reset is returned. When writing to this bit, write the value after reset.
14	ADERR	ACK Delimiter Error Flag 0: No ACK delimiter error is detected. 1: ACK delimiter error is detected.
13	B0ERR	Dominant Bit Error Flag 0: No dominant bit error is detected. 1: Dominant bit error is detected.
12	B1ERR	Recessive Bit Error Flag 0: No recessive bit error is detected. 1: Recessive bit error is detected.
11	CERR	CRC Error Flag 0: No CRC error is detected. 1: CRC error is detected.
10	AERR	ACK Error Flag 0: No ACK error is detected. 1: ACK error is detected.
9	FERR	Form Error Flag 0: No form error is detected. 1: Form error is detected.
8	SERR	Stuff Error Flag 0: No stuff error is detected. 1: Stuff error is detected.
7	ALF	Arbitration Lost Flag 0: No arbitration lost is detected. 1: Arbitration lost is detected.

Table 20.25 RSCANnCmERFL Register Contents (2/2)

Bit Position	Bit Name	Function
6	BLF	Bus Lock Flag 0: No channel bus lock is detected. 1: Channel bus lock is detected.
5	OVLf	Overload Flag 0: No overload is detected. 1: Overload is detected.
4	BORF	Bus Off Recovery Flag 0: No bus off recovery is detected. 1: Bus off recovery is detected.
3	BOEF	Bus Off Entry Flag 0: No bus off entry is detected. 1: Bus off entry is detected.
2	EPF	Error Passive Flag 0: No error passive is detected. 1: Error passive is detected.
1	EWf	Error Warning Flag 0: No error warning is detected. 1: Error warning is detected.
0	BEF	Bus Error Flag 0: No channel bus error is detected. 1: Channel bus error is detected.

See the *CAN specification (ISO11898-1)* for a description of error occurrence conditions. To clear each flag of this register, the program must write 0 to the corresponding bit. These flags cannot be set to 1 by the program. If any of these flags is set to 1 at the same time that the program writes 0 to the flag, the flag is still set to 1. The channel reset mode transition clears all of these flags to 0.

If the ERRD bit in the RSCANnCmCTR register is set to 0 (i.e., only the flags for the first error event are displayed) and an error related to bits 14 to 8 of RSCANnCmERFL is detected, the flag bits are set to 1 by the error event if bits 14 to 8 were all 0 when the error occurred.

CRCREG[14:0] Flags

When the CTME bit in the RSCANnCmCTR register is set to 1 (communication test mode is enabled), the CRC value calculated based on the transmitted or received message can be read. When the CTME bit is set to 0 (communication test mode is disabled), these bits are always read as 0. This bit is always 0 in channel reset mode.

ADERR Flag

This flag is set to 1 when a form error has been detected in the ACK delimiter during transmission.

B0ERR Flag

This flag is set to 1 when a recessive bit has been detected even though a dominant bit was transmitted.

B1ERR Flag

This flag is set to 1 when a dominant bit has been detected even though a recessive bit was transmitted.

CERR Flag

This flag is set to 1 when a CRC error has been detected.

AERR Flag

This flag is set to 1 when an ACK error has been detected.

FERR Flag

This flag is set to 1 when a form error has been detected.

SERR Flag

This flag is set to 1 when a stuff error has been detected.

ALF Flag

This flag is set to 1 when an arbitration-lost has been detected.

BLF Flag

This flag is set to 1 when 32 consecutive dominant bits have been detected on the CAN bus in channel communication mode. After that, detection of bus lock is restarted when either of the following conditions is met.

- A recessive bit is detected after the BLF bit has been modified from 1 to 0.
- The RS-CANFD module transitions to channel reset mode and returns to channel communication mode after the BLF bit has been modified from 1 to 0.

OVLV Flag

This flag is set to 1 when the overload frame transmit condition has been detected when performing reception or transmission.

BORF Flag

This flag is set to 1 when 11 consecutive recessive bits have been detected 128 times and the RS-CANFD module returns from the bus off state. However, this flag is not set to 1 if the RS-CANFD module returns from the bus off state in either of the following cases before 11 consecutive recessive bits are detected 128 times.

- The CHMDC[1:0] bits in the RSCANnCMCTR register are set to 01_B (channel reset mode).
- The RTBO bit in the RSCANnCMCTR register is set to 1 (forcible return from the bus off state).
- The BOM[1:0] bits in the RSCANnCMCTR register are set to 01_B (transition to channel halt mode at bus off entry).
- The CHMDC[1:0] bits in the RSCANnCMCTR register are set to 10_B (channel halt mode) before 11 consecutive recessive bits are detected 128 times with the BOM[1:0] bits set to 11_B (transition to channel halt mode upon a request from the program during bus off).

BOEF Flag

This flag is set to 1 when the state becomes bus off state (TEC[7:0] value > 255). This flag is also set to 1 when the state becomes bus off state with the BOM[1:0] bits in the RSCANnCMCTR register (m = see **Table 20.6**) are set to 01_B (transition to channel halt mode at bus off entry).

EPF Flag

This flag is set to 1 when the error passive state is entered ($(128 \leq \text{TEC}[7:0] \leq 255)$ or $(128 \leq \text{REC}[7:0])$). This flag is set to 1 only when the REC[7:0] or TEC[7:0] value exceeds 127 for the first time. Therefore, if the program writes 0 to this flag with the REC[7:0] or TEC[7:0] value remaining over 127, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 127 or less and then either the REC[7:0] or TEC[7:0] value exceeds 127 again.

EWV Flag

This flag is set to 1 only when the REC[7:0] or TEC[7:0] value exceeds 95 for the first time. Therefore, if the program writes 0 to this flag with the REC[7:0] or TEC[7:0] value remaining over 95, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 95 or less and then either the REC[7:0] or TEC[7:0] value exceeds 95 again.

BEF Flag

This flag is set to 1 when any one of the ADERR, B0ERR, B1ERR, CERR, AERR, FERR, and SERR flags in the RSCANnCMERFL register is set to 1.

NOTE

To clear the flag of this register to 0, use a store instruction to write “0” to that flag and “1” to the other flags.

20.3.4 Details of Global-Related Registers

20.3.4.1 RSCANnGCFG — Global Configuration Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ITRCP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSBTCS[2:0]		TSSS	TSP[3:0]				TMT SCE	EEFE	—	DCS	MME	DRE	DCE	TPRI	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 20.26 RSCANnGCFG Register Contents (1/2)

Bit Position	Bit Name	Function																																				
31 to 16	ITRCP[15:0]	Interval Timer Prescaler Set When these bits are set to q, the peripheral function clock A is divided by q. When using interval timer, setting 0000 _H is prohibited.																																				
15 to 13	TSBTCS[2:0]	Timestamp Clock Source Select <table border="0" style="margin-left: 20px;"> <tr> <td>b15</td> <td>b14</td> <td>b13</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0: Channel 0 bit time clock</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1: Channel 1 bit time clock</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0: Channel 2 bit time clock</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1: Channel 3 bit time clock</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0: Channel 4 bit time clock</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1: Setting prohibited</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0: Setting prohibited</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1: Setting prohibited</td> </tr> </table>	b15	b14	b13		0	0	0	0: Channel 0 bit time clock	0	0	1	1: Channel 1 bit time clock	0	1	0	0: Channel 2 bit time clock	0	1	1	1: Channel 3 bit time clock	1	0	0	0: Channel 4 bit time clock	1	0	1	1: Setting prohibited	1	1	0	0: Setting prohibited	1	1	1	1: Setting prohibited
b15	b14	b13																																				
0	0	0	0: Channel 0 bit time clock																																			
0	0	1	1: Channel 1 bit time clock																																			
0	1	0	0: Channel 2 bit time clock																																			
0	1	1	1: Channel 3 bit time clock																																			
1	0	0	0: Channel 4 bit time clock																																			
1	0	1	1: Setting prohibited																																			
1	1	0	0: Setting prohibited																																			
1	1	1	1: Setting prohibited																																			
12	TSSS	Timestamp Source Select 0: pclk/2*1 1: Bit time clock is used as the source clock of the timestamp counter.																																				

Table 20.26 RSCANnGCFG Register Contents (2/2)

Bit Position	Bit Name	Function
11 to 8	TSP[3:0]	Timestamp Clock Source Division b11 b10 b9 b8 0 0 0 0: Not divided 0 0 0 1: Divided by 2 0 0 1 0: Divided by 4 0 0 1 1: Divided by 8 0 1 0 0: Divided by 16 0 1 0 1: Divided by 32 0 1 1 0: Divided by 64 0 1 1 1: Divided by 128 1 0 0 0: Divided by 256 1 0 0 1: Divided by 512 1 0 1 0: Divided by 1024 1 0 1 1: Divided by 2048 1 1 0 0: Divided by 4096 1 1 0 1: Divided by 8192 1 1 1 0: Divided by 16384 1 1 1 1: Divided by 32768
7	TMTSCE	Transmission Timestamp Enable 0: Transmission timestamp is disabled. 1: Transmission timestamp is enabled.
6	EEFE	ECC Error Flag Enable 0: The ECC error flag is disabled. 1: The ECC error flag is enabled.
5	—	Reserved When read, the value after reset is returned. When writing to this bit, write the value after reset.
4	DCS	CAN Clock Source Select 0: clkc 1: clk_xincan
3	MME	Mirror Function Enable 0: Mirror function is disabled. 1: Mirror function is enabled.
2	DRE	DLC Replacement Enable 0: DLC replacement is disabled. 1: DLC replacement is enabled.
1	DCE	DLC Check Enable 0: DLC check is disabled. 1: DLC check is enabled.
0	TPRI	Transmit Priority Select 0: ID priority 1: Transmit buffer number priority

Note 1. Set TSBTCS[2:0] to 000_b before select pclk/2 as the timestamp count source.

Modify the RSCANnGCFG register in global reset mode.

ITRCP[15:0] Bits

These bits are used to set a clock source division value of the interval timer for FIFO buffers. For details, see **Section 20.8.3.1, Interval Transmission Function**.

TSBTCS[2:0] Bits

When the TSSS bit is 1, these bits are used to select the channel of the bit time clock that will be the clock source of the timestamp counter.

TSSS Bit

This bit is used to select a clock source of the timestamp counter.

TSP[3:0] Bits

These bits specify the prescaler used to divide the clock source (selected by the TSBTCS[2:0] and TSSS bits) to generate the timestamp counter clock signal.

TMTSCE Bit

Setting this bit to 1 makes it possible to store the timestamp of a transmitted message in the transmit history buffer. The timestamp is stored in TMTS[15:0] bits in the RSCANnTHLACCm register.

EEFE Bit

Setting this bit to 1 sets the EEFm bit in the RSCANnGERFL register to 1 when a 2-bit ECC error is detected during the transmission priority determination. At this time, the message in which a 2-bit ECC error was detected is not transmitted.

DCS Bit

When this bit is set to 0, clk_c is used as the clock source of the CAN clock (fCAN).

When this bit is set to 1, the main clock is used as the clock source of the CAN clock (fCAN).

MME Bit

Setting this bit to 1 makes the mirror function available.

DRE Bit

When the DRE bit is set to 1, the DLC value of the receive rule is stored in the buffer instead of the DLC value of the received message after the DLC value has passed through the DLC filter. In this case, a value of 00_H is stored in the data byte that exceeds the DLC value of the receive rule.

When the DCE bit is set to 1 (DLC check is enabled), the DLC replacement function is available.

DCE Bit

Setting this bit to 1 makes the DLC check function available. Set the GAFLDLC[3:0] bits in the RSCANnGAFLP0_j register to 0000_B before clearing the DCE bit in the RSCANnGCFG register to 0.

TPRI Bit

This bit is used to set the transmit priority.

When this bit is set to 0, ID priority is selected and the transmit priority complies with the CAN bus arbitration rule (ISO11898-1 specifications). When this bit is set to 1, transmit buffer number priority is selected and the lowest-numbered transmit buffer among those specified for transmission takes precedence.

When using the transmit queue, clear this bit to 0.

20.3.4.2 RSCANnGCTR — Global Control Register

Value after reset: 0000 0005_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 20.27 RSCANnGCTR Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
16	TSRST	Timestamp Counter Reset Setting the TSRST bit to 1 resets the timestamp counter. This bit is always read as 0.
15 to 11	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
10	THLEIE	Transmit History Buffer Overflow Interrupt Enable 0: Transmit history buffer overflow interrupt is disabled. 1: Transmit history buffer overflow interrupt is enabled.
9	MEIE	FIFO Message Lost Interrupt Enable 0: FIFO message lost interrupt is disabled. 1: FIFO message lost interrupt is enabled.
8	DEIE	DLC Error Interrupt Enable 0: DLC error interrupt is disabled. 1: DLC error interrupt is enabled.
7 to 3	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	GSLPR	Global Stop Mode 0: Other than global stop mode 1: Global stop mode
1, 0	GMDC[1:0]	Global Mode Select b1 b0 0 0: Global operating mode 0 1: Global reset mode 1 0: Global test mode 1 1: Setting prohibited

TSRST Bit

This bit is used to reset the timestamp counter. When this bit is set to 1, the RSCANnGTSC register is cleared to 0000_H.

THLEIE Bit

When the THLEIE bit is set to 1 and the THLES flag in the RSCANnGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

MEIE Bit

When the MEIE bit is set to 1 and the MES flag in the RSCANnGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

DEIE Bit

When the DEIE bit is set to 1 and the DEF flag in the RSCANnGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

GSLPR Bit

Setting this bit to 1 places the RS-CANFD module in global stop mode.

Clearing this bit to 0 releases the RS-CANFD module from global stop mode.

Do not modify this bit from 0 to 1 in global operation mode or global test mode.

GMDC[1:0] Bits

These bits are used to select a mode of the entire RS-CANFD module (global operating mode, global reset mode, or global test mode). For details, see **Section 20.6.1, Global Modes**. Setting the GSLPR bit to 1 in global reset mode makes the RS-CANFD module transition to global stop mode.

20.3.4.3 RSCANnGSTS — Global Status Register

Value after reset: 0000 000D_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	GRAM INIT	GSLPS TS	GHLT TS	GRST TS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.28 RSCANnGSTS Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	GRAMINIT	CAN RAM Initialization Status Flag 0: CAN RAM initialization is completed. 1: CAN RAM initialization is ongoing.
2	GSLPSTS	Global Stop Status Flag 0: Not in global stop mode 1: In global stop mode
1	GHLTSTS	Global Test Status Flag 0: Not in global test mode 1: In global test mode
0	GRSTSTS	Global Reset Status Flag 0: Not in global reset mode 1: In global reset mode

GRAMINIT Flag

This flag indicates the initialization status of the CAN RAM.

This flag is set to 1 after the MCU has been reset, and is cleared to 0 when CAN RAM initialization is completed.

GSLPSTS Flag

This flag is set to 1 when the RS-CANFD module has transitioned to global stop mode, and is cleared to 0 when the RS-CANFD module has returned from global stop mode.

GHLTSTS Flag

This flag is set to 1 when the RS-CANFD module has transitioned to global test mode, and is cleared to 0 when the RS-CANFD module has exited global test mode.

GRSTSTS Flag

This flag is set to 1 when the RS-CANFD module has transitioned to global reset mode, and is cleared to 0 when the RS-CANFD module has exited global reset mode. This flag remains 1 if the RS-CANFD module has transitioned from global reset mode to global stop mode.

20.3.4.4 RSCANnGERFL — Global Error Flag Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	EEF4	EEF3	EEF2	EEF1	EEF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	THLES	MES	DEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1

Note 1. The only effective value for writing to this flag bit is 0, which clears the status. Any other writing to the bit results in retention of the status and does not change the value.

Table 20.29 RSCANnGERFL Register Contents

Bit Position	Bit Name	Function
31 to 21	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
20	EEF4	ECC Error Flag for Unit Channel 4 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
19	EEF3	ECC Error Flag for Unit Channel 3 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
18	EEF2	ECC Error Flag for Unit Channel 2 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
17	EEF1	ECC Error Flag for Unit Channel 1 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
16	EEF0	ECC Error Flag for Unit Channel 0 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
15 to 14, 7 to 6, 4 to 3	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
13 to 8, 5	—	Reserved When read, the undefined value is returned. When writing these bits, write the value after reset.
2	THLES	Transmit History Buffer Overflow Status Flag 0: No transmit history buffer overflow is present. 1: A transmit history buffer overflow is present.
1	MES	FIFO Message Lost Status Flag 0: No FIFO message lost error is present. 1: A FIFO message lost error is present.
0	DEF	DLC Error Flag 0: No DLC error is present. 1: A DLC error is present.

All flags in the RSCANnGERFL register are cleared to 0 in global reset mode.

EEFm Flag

While the EEFE bit in the RSCANnGCFG register is 1, when a 2-bit ECC error is detected during the transmission priority determination of channel m (m = see **Table 20.6**), the EEFm flag is set to 1, disabling message transmission. This flag can be cleared to 0 by writing 0 by the program.

THLES Flag

The THLES flag is set to 1 when any one of the THLELT flags in the RSCANnTHLSTSm register (m = see **Table 20.6**) is set to 1.

This flag is cleared to 0 when the THLELT flags of all channels are set to 0.

MES Flag

The MES flag is set to 1 when any one of the RFMLT flags in the RSCANnRFSTSm register (x = see **Table 20.6**) or the CFMLT flags in the RSCANnCFSTSk register (k = see **Table 20.6**) is set to 1.

This flag is cleared to 0 when all the RFMLT flags and CFMLT flags are set to 0.

DEF Flag

The DEF flag is set to 1 when an error has been detected during the DLC check. This flag can be cleared to 0 by writing 0 by the program.

NOTE

To clear the flag of the register to 0, the program must write 0 to the given flag. When writing 0, use a store instruction to write 0 to the given flag and 1 to the other flags.

20.3.4.5 RSCANnGTSC — Global Timestamp Counter Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.30 RSCANnGTSC Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When read, the value after reset is returned.
15 to 0	TS[15:0]	Timestamp value The timestamp counter value can be read. Counter value: 0000 _H to FFFF _H

TS[15:0] Bits

When the TS[15:0] bits are read, the read value shows the timestamp counter (16-bit free-running counter) value at that time. The TS[15:0] value at the time when the SOF of the received message is detected is stored in the receive buffer or the FIFO buffer. The timestamp counter is initialized in global reset mode.

Start timing and stop timing of the timestamp counter depend on the count source.

- When the TSSS bit value in the RSCANnGCFG register is 0 (pclk):
The timestamp counter starts counting when the RS-CANFD module has transitioned to global operating mode. This counter stops counting when the RS-CANFD module has transitioned to global stop mode or global test mode.
- When the TSSS bit value in the RSCANnGCFG register is 1 (CANm bit time clock):
The timestamp counter starts counting when the corresponding channel has transitioned to channel communication mode. This counter stops counting when the corresponding channel has transitioned to channel reset mode or channel halt mode.

20.3.4.6 RSCANnGTINTSTS0 — Global TX Interrupt Status Register 0

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	THIF3	CFTIF3	TQIF3	TAIF3	TSIF3	—	—	—	THIF2	CFTIF2	TQIF2	TAIF2	TSIF2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}	R	R	R	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THIF1	CFTIF1	TQIF1	TAIF1	TSIF1	—	—	—	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}	R	R	R	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}

Note 1. The bit is automatically cleared in global reset or channel reset mode.

Table 20.31 RSCANnGTINTSTS0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
28	THIF3	Unit Channel 3 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
27	CFTIF3	Unit Channel 3 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
26	TQIF3	Unit Channel 3 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
25	TAIF3	Unit Channel 3 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
24	TSIF3	Unit Channel 3 Transmit Buffer Transmit Complete Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
23 to 21	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
20	THIF2	Unit Channel 2 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
19	CFTIF2	Unit Channel 2 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
18	TQIF2	Unit Channel 2 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
17	TAIF2	Unit Channel 2 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.

Table 20.31 RSCANnGTINTSTS0 Register Contents (2/2)

Bit Position	Bit Name	Function
16	TSIF2	Unit Channel 2 Transmit Buffer Transmit Complete Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
15 to 13	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
12	THIF1	Unit Channel 1 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
11	CFTIF1	Unit Channel 1 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
10	TQIF1	Unit Channel 1 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
9	TAIF1	Unit Channel 1 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested
8	TSIF1	Unit Channel 1 Transmit Buffer Transmit Complete Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
7 to 5	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
4	THIF0	Unit Channel 0 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
3	CFTIF0	Unit Channel 0 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
2	TQIF0	Unit Channel 0 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
1	TAIF0	Unit Channel 0 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested
0	TSIF0	Unit Channel 0 Transmit Buffer Transmit Complete Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.

TSIFm Bit

When the RSCANnTMIECy.TMIEp bit is set to 1 (transmit buffer interrupt is enabled) and corresponding RSCANnTMSTSp.TMTRF[1:0] flags are set to 10_B (transmission has been completed, without transmit abort request) or 11_B (transmission has been completed, with transmit abort request), the TSIFm bit becomes 1.

The TSIFm flag becomes 0 by setting 00_B to all the TMTRF[1:0] flags which satisfy the condition to lead TSIFm = 1. The TSIFm flag also becomes 0 by setting the TMIEp bit to 0.

TAIFm Bit

When the RSCANnCMCTR.TAIE bit is set to 1 (transmit abort interrupt is enabled) and the RSCANnTMSTSp.TMTRF[1:0] flags are 01_B (transmit abort has been completed), the TAIFm bit becomes 1.

The TAIFm flag becomes 0 by setting 00_B to all the TMTRF[1:0] flags of which transmit abort has been completed.

TQIFm Bit

When the RSCANnTXQCCm.TXQIE bit is set to 1 (transmit queue interrupt is enabled) and the RSCANnTXQSTSm.TXQIF bit is set to 1 (a transmit queue interrupt request is present), the TQIFm bit becomes 1.

When the RSCANnTXQSTSm.TXQIF bit (transmit queue interrupt request) is set to 0, the TQIFm bit becomes 0. The TQIFm flag also becomes 0 by setting the TXQIE bit to 0.

CFTIFm Bit

When the RSCANnCFCCk.CFTXIE bit is set to 1 (transmit/receive FIFO transmit interrupt is enabled) and the RSCANnCFSTSk.CFTXIF bit is set to 1 (a transmit/receive FIFO transmit interrupt request is present), the CFTIFm bit becomes 1.

The CFTIFm bit becomes 0 by setting all the RSCANnCFSTSk.CFTXIF bits to 0. The CFTIFm flag also becomes 0 by setting the CFTXIE bit to 0.

THIFm Bit

When the RSCANnTHLCCm.THLIE bit is set to 1 (transmit history interrupt is enabled) and the RSCANnTHLSTSm.THLIF bit is set to 1 (a transmit history interrupt request is present), the THIFm bit becomes 1.

The THIFm bit becomes 0 by setting the RSCANnTHLSTSm.THLIF bit to 0. The THIFm flag also becomes 0 by setting the THLIE bit to 0.

20.3.4.7 RSCANnGTINTSTS1 — Global TX Interrupt Status Register 1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	THIF4	CFTIF4	TQIF4	TAIF4	TSIF4
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R*	R*	R*	R*	R*

Note 1. The bit is automatically cleared in global reset or channel reset mode.

Table 20.32 RSCANnGTINTSTS1 Register Contents

Bit Position	Bit Name	Function
31 to 5	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
4	THIF4	Unit Channel 4 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
3	CFTIF4	Unit Channel 4 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
2	TQIF4	Unit Channel 4 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
1	TAIF4	Unit Channel 4 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
0	TSIF4	Unit Channel 4 Transmit Buffer Transmit Complete Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.

TSIFm Bit

When the RSCANnTMIECy.TMIE bit is set to 1 (transmit buffer interrupt is enabled) and corresponding RSCANnTMSTSp.TMTRF[1:0] flags are set to 10_B (transmission has been completed, without transmit abort request) or 11_B (transmission has been completed, with transmit abort request), the TSIFm bit becomes 1.

The TSIFm flag becomes 0 by setting 00_B to all the TMTRF[1:0] flags which satisfy the condition to lead TSIFm = 1. The TSIFm flag also becomes 0 by setting the TMIE bit to 0.

TAIFm Bit

When the RSCANnCmCTR.TAIE bit is set to 1 (transmit abort interrupt is enabled) and the RSCANnTMSTSp.TMTRF[1:0] flags are 01_B (transmit abort has been completed), the TAIFm bit becomes 1.

The TAIFm flag becomes 0 by setting 00_B to all the TMTRF[1:0] flags of which transmit abort has been completed.

TQIFm Bit

When the RSCANnTXQCCm.TXQIE bit is set to 1 (transmit queue interrupt is enabled) and the RSCANnTXQSTSm.TXQIF bit is set to 1 (a transmit queue interrupt request is present), the TQIFm bit becomes 1.

When the RSCANnTXQSTSm.TXQIF bit (transmit queue interrupt request) is set to 0, the TQIFm bit becomes 0. The TQIFm flag also becomes 0 by setting the TXQIE bit to 0.

CFTIFm Bit

When the RSCANnCFCCk.CFTXIE bit is set to 1 (transmit/receive FIFO transmit interrupt is enabled) and the RSCANnCFSTSk.CFTXIF bit is set to 1 (a transmit/receive FIFO transmit interrupt request is present), the CFTIFm bit becomes 1.

The CFTIFm bit becomes 0 by setting all the RSCANnCFSTSk.CFTXIF bits to 0. The CFTIFm flag also becomes 0 by setting the CFTXIE bit to 0.

THIFm Bit

When the RSCANnTHLCCm.THLIE bit is set to 1 (transmit history interrupt is enabled) and the RSCANnTHLSTSm.THLIF bit is set to 1 (a transmit history interrupt request is present), the THIFm bit becomes 1.

The THIFm bit becomes 0 by setting the RSCANnTHLSTSm.THLIF bit to 0. The THIFm flag also becomes 0 by setting the THLIE bit to 0.

20.3.4.8 RSCANnGFDCFG — Global FD Configuration Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TSCCFG[1:0]		—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Table 20.33 RSCANnGFDCFG Register Contents

Bit Position	Bit Name	Function
31 to 10	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
9 to 8	TSCCFG[1:0]	Timestamp capture configuration <small>b9 b8</small> 0 0: Captured at a sample point in the SOF bit 0 1: Captured when a valid frame has been transmitted/received 1 0: Setting prohibited 1 1: Setting prohibited
7 to 1	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	—	Reserved When read, an undefined value is returned. The write value should be the value after reset.

TSCCFG Bit

Select a point where a time-stamp value is captured. Modify this bit only in global reset mode.

20.3.5 Details of Receive Rule-Related Registers

20.3.5.1 RSCANnGAFLECTR — Receive Rule Entry Control Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	AFL DAE	—	—	—	AFLPN[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 20.34 RSCANnGAFLECTR Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
8	AFLDAE	Receive Rule Table Write Enable 0: Receive rule table write is disabled. 1: Receive rule table write is enabled.
7 to 5	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
4 to 0	AFLPN [4:0]	Receive Rule Table Page Number Configuration A page number can be selected in the range of page 0 (00000 _B) to page 19 (10011 _B).

AFLDAE Bit

Setting this bit to 0 disables data writing to the receive rule table. After data writing to the receive rule table is completed, set this bit to 0 to disable writing to the receive rule table. The receive rule table can be read regardless of the value of this bit.

Set the AFLDAE bit to 1 in global reset mode.

AFLPN[4:0] Bits

These bits are used to set the page number of the receive rule table. Sixteen receive rules can be set per page.

Set these bits to a value within a range of 00000_B to 10011_B.

20.3.5.2 RSCANnGAFLCFG0 — Receive Rule Configuration Register 0

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RNC0[7:0]								RNC1[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RNC2[7:0]								RNC3[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.35 RSCANnGAFLCFG0 Register Contents

Bit Position	Bit Name	Function
31 to 24	RNC0[7:0]	Number of Rules for Unit Channel 0 The number of rules for channel 0
23 to 16	RNC1[7:0]	Number of Rules for Unit Channel 1 The number of rules for channel 1
15 to 8	RNC2[7:0]	Number of Rules for Unit Channel 2 The number of rules for channel 2
7 to 0	RNC3[7:0]	Number of Rules for Unit Channel 3 The number of rules for channel 3

Modify the RSCANnGAFLCFG0 register in global reset mode.

Up to “64 × (number of channels)” rules can be registered to the receive rule tables in the entire module. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 128.
- The total of the number of rules allocated to each channel is not larger than the number of rules permitted to the entire unit.

RNC0[7:0] Bits

These bits are used to set the number of rules to be registered in the unit channel 0 receive rule table.

Set these bits to a value within a range of 00_H to 80_H.

RNC1[7:0] Bits

These bits are used to set the number of rules to be registered in the unit channel 1 receive rule table.

Set these bits to a value within a range of 00_H to 80_H.

RNC2[7:0] Bits

These bits are used to set the number of rules to be registered in the unit channel 2 receive rule table.

Set these bits to a value within a range of 00_H to 80_H.

RNC3[7:0] Bits

These bits are used to set the number of rules to be registered in the unit channel 3 receive rule table.

Set these bits to a value within a range of 00_H to 80_H.

20.3.5.3 RSCANnGAFLCFG1 — Receive Rule Configuration Register 1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RNC4[7:0]								—							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.36 RSCANnGAFLCFG1 Register Contents

Bit Position	Bit Name	Function
31 to 24	RNC4[7:0]	Number of Rules for Unit Channel 4 The number of rules for channel 4
23 to 0	—	Reserved These bits are always read as 0. The write value should always be 0.

Modify the RSCANnGAFLCFG1 register in global reset mode.

Up to “64 × (number of channels)” rules can be registered to the receive rule tables in the entire module. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 128.
- The total number of rules allocated to each channel is not larger than the number of rules permitted to the entire unit.

RNC4[7:0] Bits

These bits are used to set the number of rules to be registered in the unit channel 4 receive rule table.

Set these bits to a value within a range of 00_H to 80_H.

20.3.5.4 RSCANnGAFLIDj — Receive Rule ID Register j

Note: For the value of index "j", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLID E	GAFLR TR	GAFLB	GAFLID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.37 RSCANnGAFLIDj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDE	IDE Select 0: Standard ID 1: Extended ID
30	GAFLRTR	RTR Select 0: Data frame 1: Remote frame
29	GAFLLB	Receive Rule Target Message Select 0: When a message transmitted from another CAN node is received 1: When a message transmitted from own node is received
28 to 0	GAFLID[28:0]	ID Setting Set the ID of the receive rule. For the standard ID, set the ID in bits 10 to 0 and set bits 28 to 11 to 0.

Modify the RSCANnGAFLIDj register when the AFLDAE bit in the RSCANnGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDE Bit

This bit is used to select the ID format (standard ID or extended ID) of the receive rule. This bit is compared with the IDE bit in the received message during the acceptance filter processing.

GAFLRTR Bit

This bit is used to select the frame format (data frame or remote frame) of the receive rule. This bit is compared with the RTR bit in the received message during the acceptance filter processing.

GAFLLB Bit

Setting this bit to 0 causes data processing to be performed using the receive rule when a message transmitted from another CAN node is received.

Setting this bit to 1 when the mirror function is used causes data processing to be performed using the receive rule when a message transmitted from the own CAN node is received.

GAFLID[28:0] Bits

These bits are used to set the ID field of the receive rule. The ID set by these bits is compared with the ID in the received message during the acceptance filter processing.

20.3.5.5 RSCANnGAFLMj — Receive Rule Mask Register j

Note: For the value of index "j", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDEM	GAFLRTRM	—	GAFLIDM[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLIDM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.38 RSCANnGAFLMj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDEM	IDE Mask 0: The IDE bit is not compared. 1: The IDE bit is compared.
30	GAFLRTRM	RTR Mask 0: The RTR bit is not compared. 1: The RTR bit is compared.
29	—	Reserved When read, the value after reset is returned. When writing to this bit, write the value after reset.
28 to 0	GAFLIDM [28:0]	ID Mask 0: The corresponding ID bit is not compared. 1: The corresponding ID bit is compared.

Modify the RSCANnGAFLMj register when the AFLDAE bit in the RSCANnGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDEM Bit

When this bit is set to 1, filter processing is performed only for messages of the ID format specified by the GAFLIDE bit in the RSCANnGAFLIDj register.

When this bit is cleared to 0, the IDs of all the receive messages and the specified IDs are regarded as matched.

To set the GAFLIDEM bit to 0, set all the GAFLIDM[28:0] bits to 0 at the same time.

GAFLRTRM Bit

This bit is used to mask the RTR bit of the receive rule.

GAFLIDM[28:0] Bits

These bits are used to mask the corresponding ID bits of the receive rule.

20.3.5.6 RSCANnGAFLP0_j — Receive Rule Pointer 0 Register j

Note: For the value of index "j", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLDLC[3:0]				GAFLPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLR MV	GAFLRMDP[6:0]						—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 20.39 RSCANnGAFLP0_j Register Contents

Bit Position	Bit Name	Function
31 to 28	GAFLDLC [3:0]	Receive Rule DLC Set b31 b30 b29 b28 0 0 0 0: DLC check is disabled. 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	GAFLPTR [11:0]	Receive Rule Label Set Set the 12-bit label information.
15	GAFLRMV	Receive Buffer Enable 0: No receive buffer is used. 1: A receive buffer is used.
14 to 8	GAFLRMDP [6:0]	Receive Buffer Number Select Set the receive buffer number to store received messages.
7 to 0	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.

Modify the RSCANnGAFLP0_j register when the AFLDAE bit in the RSCANnGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLDLC[3:0] Bits

These bits are used to set the minimum data length necessary for receiving messages. If the data length of a message that is being filtered is equal to or larger than the value set by the GAFLDLC[3:0] bits, the message passes the DLC check. Setting these bits to 0000_B disables the DLC check function, allowing messages with any data length to pass the DLC check.

GAFLPTR[11:0] Bits

These bits are used to set a 12-bit label to be attached to messages that have passed through the filter. A label is attached when a message is stored in the receive buffer or the FIFO buffer.

GAFLRMV Bit

When this bit is set to 1, received messages that have passed through the filter are stored in the receive buffer selected by the GAFLRMDP[6:0] bits.

GAFLRMDP[6:0] Bits

These bits are used to select the number of a receive buffer that stores received messages having passed through the filter when the GAFLRMV bit is set to 1. Set these bits to a value smaller than the value set by the NRXMB[7:0] bits in the RSCANnRMNB register.

20.3.5.7 RSCANnGAFLP1_j — Receive Rule Pointer 1 Register j

Note: For the value of index "j", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	GAFLFDP[22:16]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLFDP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.40 RSCANnGAFLP1_j Register Contents

Bit Position	Bit Name	Function
31 to 23	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
22 to 8	GAFLFDP[22:8]	Transmit/Receive FIFO Buffer k Select (Bit position – 8 = target transmit/receive FIFO buffer number k) 0: Transmit/receive FIFO buffer is not selected. 1: Transmit/receive FIFO buffer is selected.
7 to 0	GAFLFDP[7:0]	Receive FIFO Buffer x Select (Bit position = target receive FIFO buffer number x) 0: Receive FIFO buffer is not selected. 1: Receive FIFO buffer is selected.

Modify the RSCANnGAFLP1_j register when the AFLDAE bit in the RSCANnGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLFDP [22:0] Bits

These bits are used to specify FIFO buffers that store received messages having passed through the filter. Up to eight FIFO buffers can be selected. However, when the GAFLRMV bit in the RSCANnGAFLP0_j register is set to 1 (a receive buffer is used), up to seven FIFO buffers are selectable. Selectable buffers are receive FIFO buffers and the transmit/receive FIFO buffers for which the CFM[1:0] bits in the RSCANnCFCCk register are set to 00_B (receive mode) or 10_B (gateway mode).

20.3.6 Details of Receive Buffer-Related Registers

20.3.6.1 RSCANnRMNB — Receive Buffer Number Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	NRXMB[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.41 RSCANnRMNB Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
7 to 0	NRXMB [7:0]	Receive Buffer Number Configuration Set the number of receive buffers. Set a value of 0 to 80.

Modify the RSCANnRMNB register in global reset mode.

NRXMB[7:0] Bits

These bits are used to set the total number of receive buffers of the entire RS-CANFD module. The maximum value is “16 × (number of channels)”.

Setting all of these bits 0 makes receive buffers unavailable.

20.3.6.2 RSCANnRMNDy — Receive Buffer New Data Register y

Note: For the value of index "y", see Table 20.6.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMNSq (q = y × 32 + 31)	RMNSq (q = y × 32 + 30)	RMNSq (q = y × 32 + 29)	RMNSq (q = y × 32 + 28)	RMNSq (q = y × 32 + 27)	RMNSq (q = y × 32 + 26)	RMNSq (q = y × 32 + 25)	RMNSq (q = y × 32 + 24)	RMNSq (q = y × 32 + 23)	RMNSq (q = y × 32 + 22)	RMNSq (q = y × 32 + 21)	RMNSq (q = y × 32 + 20)	RMNSq (q = y × 32 + 19)	RMNSq (q = y × 32 + 18)	RMNSq (q = y × 32 + 17)	RMNSq (q = y × 32 + 16)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMNSq (q = y × 32 + 15)	RMNSq (q = y × 32 + 14)	RMNSq (q = y × 32 + 13)	RMNSq (q = y × 32 + 12)	RMNSq (q = y × 32 + 11)	RMNSq (q = y × 32 + 10)	RMNSq (q = y × 32 + 9)	RMNSq (q = y × 32 + 8)	RMNSq (q = y × 32 + 7)	RMNSq (q = y × 32 + 6)	RMNSq (q = y × 32 + 5)	RMNSq (q = y × 32 + 4)	RMNSq (q = y × 32 + 3)	RMNSq (q = y × 32 + 2)	RMNSq (q = y × 32 + 1)	RMNSq (q = y × 32 + 0)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.42 RSCANnRMNDy Register Contents

Bit Position	Bit Name	Function
31 to 0	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 31 to y × 32 + 0) 0: Receive buffer q contains no new message. 1: Receive buffer q contains a new message.

Write 0 to the RSCANnRMNDy register in global operating mode or global test mode.

RMNSq Flags (q = see Table 20.6)

Each flag is set to 1 when a process starts to store a message in the corresponding receive buffer.

To clear these flags to 0, write 0 by the program. Use a store instruction to write "0" to the flag and "1" to other flags.

These bits cannot be set to 0 while a message is being stored. Time for storing a message is 10 pclk clock cycles.

These flags are cleared to 0 in global reset mode.

20.3.6.3 RSCANnRMIDq — Receive Buffer ID Register q

Note: For the value of index "q", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMIDE	RMRTR	—	RMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.43 RSCANnRMIDq Register Contents

Bit Position	Bit Name	Function
31	RMIDE	Receive Buffer IDE 0: Standard ID 1: Extended ID
30	RMRTR	Receive Buffer RTR 0: Data frame 1: Remote frame
29	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
28 to 0	RMID [28:0]	Receive Buffer ID Data The standard ID or extended ID of received message can be read. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

RMIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive buffer.

RMRTR Bit

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive buffer.

RMID[28:0] Bits

These bits indicate the ID of the message stored in the receive buffer.

20.3.6.4 RSCANnRMPTRq — Receive Buffer Pointer Register q

Note: For the value of index "q", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDLC[3:0]				RMPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.44 RSCANnRMPTRq Register Contents

Bit Position	Bit Name	Function
31 to 28	RMDLC [3:0]	Receive Buffer DLC Data b31 b30 b29 b28 0 0 0 0: No data byte 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	RMPTR [11:0]	Receive Buffer Label Data Label information of the received message can be read.
15 to 0	RMTS [15:0]	Receive Buffer Timestamp Data Timestamp value of the received message can be read.

RMDLC[3:0] Bits

These bits indicate the data length of the message stored in the receive buffer.

RMPTR[11:0] Bits

These bits indicate the label information of the message stored in the receive buffer.

RMTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the receive buffer.

20.3.6.5 RSCANnRMDF0_q — Receive Buffer Data Field 0 Register q

Note: For the value of index "q", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB3[7:0]								RMDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB1[7:0]								RMDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.45 RSCANnRMDF0_q Register Contents

Bit Position	Bit Name	Function
31 to 24	RMDB3 [7:0]	Receive Buffer Data Byte 3 Data in the message stored in the receive buffer can be read.
23 to 16	RMDB2 [7:0]	Receive Buffer Data Byte 2 Data in the message stored in the receive buffer can be read.
15 to 8	RMDB1 [7:0]	Receive Buffer Data Byte 1 Data in the message stored in the receive buffer can be read.
7 to 0	RMDB0 [7:0]	Receive Buffer Data Byte 0 Data in the message stored in the receive buffer can be read.

When the RMDLC[3:0] value in the RSCANnRMPTRq register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

20.3.6.6 RSCANnRMDF1_q — Receive Buffer Data Field 1 Register q

Note: For the value of index "q", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB7[7:0]								RMDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB5[7:0]								RMDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.46 RSCANnRMDF1_q Register Contents

Bit Position	Bit Name	Function
31 to 24	RMDB7 [7:0]	Receive Buffer Data Byte 7 Data in the message stored in the receive buffer can be read.
23 to 16	RMDB6 [7:0]	Receive Buffer Data Byte 6 Data in the message stored in the receive buffer can be read.
15 to 8	RMDB5 [7:0]	Receive Buffer Data Byte 5 Data in the message stored in the receive buffer can be read.
7 to 0	RMDB4 [7:0]	Receive Buffer Data Byte 4 Data in the message stored in the receive buffer can be read.

When the RMDLC[3:0] value in the RSCANnRMPTRq register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

20.3.7 Details of Receive FIFO Buffer-Related Registers

20.3.7.1 RSCANnRFCCx — Receive FIFO Buffer Configuration and Control Register x

Note: For the value of index "x", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	—	—	—	—	—	RFIE	RFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 20.47 RSCANnRFCCx Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
15 to 13	RFIGCV [2:0]	Receive FIFO Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.
12	RFIM	Receive FIFO Interrupt Source Select 0: An interrupt occurs when the condition set by the RFIGCV[2:0] bits is met. 1: An interrupt occurs each time a message has been received.
11	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
10 to 8	RFDC [2:0]	Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
7 to 2	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.

Table 20.47 RSCANnRFCCx Register Contents (2/2)

Bit Position	Bit Name	Function
1	RFIE	Receive FIFO Interrupt Enable 0: Receive FIFO interrupt is disabled. 1: Receive FIFO interrupt is enabled.
0	RFE	Receive FIFO Buffer Enable 0: No receive FIFO buffer is used. 1: Receive FIFO buffers are used.

RFIGCV[2:0] Bits

These bits are used to specify the number of received messages required to generate a receive FIFO interrupt request when the RFIM bit is set to 0 with a fraction for the total number of buffers (the setting of RFDC[2:0]).

When the RFDC[2:0] bits are set to 001_B (4 messages), set the RFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B. Modify these bits in global reset mode.

RFIM Bit

This bit is used to select a FIFO interrupt source. Modify this bit in global reset mode.

RFDC[2:0] Bits

These bits are used to select the number of messages that can be stored in a single receive FIFO buffer. If these bits are set to 000_B, do not use any receive FIFO buffer. Modify these bits in global reset mode.

RFIE Bit

Setting the RFIE bit to 1 enables receive FIFO interrupts. Modify this bit with the RFE bit is set to 0 (no receive FIFO buffer is used).

RFE Bit

Setting the RFE bit to 1 enables the use of receive FIFO buffers. Clearing this bit to 0 sets the RFEMP flag in the RSCANnRFSTs_x register to 1 (the receive FIFO buffer contains no unread message (buffer empty)). Modify this bit in global operating mode or global test mode.

Set this bit to 1 with another instruction after the settings to all bits in the RSCANnRFCC_x register have been done.

This bit is cleared to 0 in global reset mode.

20.3.7.2 RSCANnRFSTx — Receive FIFO Buffer Status Register x

Note: For the value of index "x", see **Table 20.6**.

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMC[7:0]							—	—	—	—	RFIF	RFMLT	RFFLL	RFEMP	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the status. Any other writing to the bit results in retention of the status and does not change the value.

Table 20.48 RSCANnRFSTx Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
15 to 8	RFMC[7:0]	Receive FIFO Unread Message Counter The number of unread messages stored in the receive FIFO buffer is displayed.
7 to 4	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	RFIF	Receive FIFO Interrupt Request Flag 0: No receive FIFO interrupt request is present. 1: A receive FIFO interrupt request is present.
2	RFMLT	Receive FIFO Message Lost Flag 0: No receive FIFO message is lost. 1: A receive FIFO message is lost.
1	RFFLL	Receive FIFO Buffer Full Status Flag 0: The receive FIFO buffer is not full. 1: The receive FIFO buffer is full.
0	RFEMP	Receive FIFO Buffer Empty Status Flag 0: The receive FIFO buffer contains unread messages. 1: The receive FIFO buffer contains no unread message (buffer empty).

RFMC[7:0] Flags

These flags indicate the number of unread messages in the receive FIFO buffer. These flags become 00_H when the RFE bit in the RSCANnRFCCx register is set to 0.

This flag is 00_H in global reset mode.

RFIF Flag

This flag is set to 1 when the conditions for receive FIFO interrupt request generation set by the RFIGCV[2:0] bits and the RFIM bit in the RSCANnRFCCx register are satisfied. This flag is cleared to 0 in global reset mode or by writing 0 to this flag. Modify this bit in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flags to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFMLT Flag

This flag is set to 1 if it is attempted to store a new message when the receive FIFO buffer is full. In this case, the new message is discarded.

This flag is cleared to 0 in global reset mode or by writing 0 to this flag.

Modify this bit in global operating mode or global test mode.

To clear the flag of the register to 0, the program must write 0 to the given flag. When writing 0, use a store instruction to write 0 to the given flag and 1 to the other flags.

RFLL Flag

This flag is set to 1 when the number of messages stored in the receive FIFO buffer matches the FIFO buffer depth set by the RFDC[2:0] bits in the RSCANnRFCCx register.

If the number of messages stored in the receive FIFO buffer becomes smaller than the FIFO buffer depth set by the RFDC[2:0] bits, this flag is cleared to 0. This flag is also cleared to 0 when the RFE bit in the RSCANnRFCCx register is set to 0 (no receive FIFO buffer is used) or in global reset mode.

RFEMP Flag

This flag is set to 1 when all messages in the receive FIFO buffer have been read. This flag is also set to 1 when the RFE bit in the RSCANnRFCCx register is 0 or in global reset mode.

This flag is cleared to 0 when even a single received message is stored in the receive FIFO buffer.

NOTE

To clear the RFMLT or RFIF flag to 0, use a store instruction to write “0” to the given flag and “1” to the other flags. Note that when using a bit-manipulation instruction for writing to this register, the read-modify-write operation may inadvertently clear the flags.

20.3.7.3 RSCANnRFPCTR_x — Receive FIFO Buffer Pointer Control Register x

Note: For the value of index "x", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 20.49 RSCANnRFPCTR_x Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When writing to these bits, write the value after reset.
7 to 0	RFPC [7:0]	Receive FIFO Pointer Control When FF _H is written to these bits, the read pointer moves to the next unread message in the receive FIFO buffer.

RFPC[7:0] Bits

When FF_H is written to the RFPC[7:0] bits, the read pointer moves to the next unread message in the receive FIFO buffer. At this time, the RFMC[7:0] (receive FIFO unread message counter) value in the RSCANnRFSTS_x register is decremented by 1. Read the RSCANnRFID_x, RSCANnRFPTR_x, RSCANnRFDf0_x, and RSCANnRFDf1_x registers to read messages in the receive FIFO buffer, and then write FF_H to the RFPC[7:0] bits.

Write FF_H to these bits when the RFE bit in the RSCANnRFCC_x register is set to 1 (receive FIFO buffers are used) and the RFEMP flag in the RSCANnRFSTS_x register is 0 (the receive FIFO buffer contains unread messages).

20.3.7.4 RSCANnRFIDx — Receive FIFO Buffer Access ID Register x

Note: For the value of index "x", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFIDE	RFRTTR	—	RFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.50 RSCANnRFIDx Register Contents

Bit Position	Bit Name	Function
31	RFIDE	Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	RFRTTR	Receive FIFO Buffer RTR 0: Data frame 1: Remote frame
29	—	Reserved When read, the value after reset is returned.
28 to 0	RFID [28:0]	Receive FIFO Buffer ID Data The standard ID or extended ID of received message can be read. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

RFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive FIFO buffer.

RFRTTR Bit

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive FIFO buffer.

RFID[28:0] Bits

These bits indicate the ID of the message stored in the receive FIFO buffer.

20.3.7.5 RSCANnRFPTRx — Receive FIFO Buffer Access Pointer Register x

Note: For the value of index "x", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDLC[3:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.51 RSCANnRFPTRx Register Contents

Bit Position	Bit Name	Function
31 to 28	RFDLC [3:0]	Receive FIFO Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	RFPTR [11:0]	Receive FIFO Buffer Label Data Label information of the received message can be read.
15 to 0	RFTS [15:0]	Receive FIFO Buffer Timestamp Data Timestamp value of the received message can be read.

RFDLC[3:0] Bits

These bits indicate the data length of the message stored in the receive FIFO buffer.

RFPTR[11:0] Bits

These bits indicate the label information of the message stored in the receive FIFO buffer.

RFTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the receive FIFO buffer.

20.3.7.6 RSCANnRFDF0_x — Receive FIFO Buffer Access Data Field 0 Register x

Note: For the value of index "x", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB3[7:0]								RFDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFDB1[7:0]								RFDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.52 RSCANnRFDF0_x Register Contents

Bit Position	Bit Name	Function
31 to 24	RFDB3 [7:0]	Receive FIFO Buffer Data Byte 3
23 to 16	RFDB2 [7:0]	Receive FIFO Buffer Data Byte 2
15 to 8	RFDB1 [7:0]	Receive FIFO Buffer Data Byte 1
7 to 0	RFDB0 [7:0]	Receive FIFO Buffer Data Byte 0

Data in the message stored in the receive FIFO buffer can be read.

When the RFDLC[3:0] value in the RSCANnRFPTRx register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

20.3.7.7 RSCANnRFDF1_x — Receive FIFO Buffer Access Data Field 1 Register x

Note: For the value of index "x", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB7[7:0]								RFDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFDB5[7:0]								RFDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.53 RSCANnRFDF1_x Register Contents

Bit Position	Bit Name	Function
31 to 24	RFDB7 [7:0]	Receive FIFO Buffer Data Byte 7
23 to 16	RFDB6 [7:0]	Receive FIFO Buffer Data Byte 6
15 to 8	RFDB5 [7:0]	Receive FIFO Buffer Data Byte 5
7 to 0	RFDB4 [7:0]	Receive FIFO Buffer Data Byte 4

Data in the message stored in the receive FIFO buffer can be read.

When the RFDLC[3:0] value in the RSCANnRFPTRx register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

20.3.8 Details of Transmit/Receive FIFO Buffer-Related Registers

20.3.8.1 RSCANnCFCCk — Transmit/Receive FIFO Buffer Configuration and Control Register k

Note: For the value of index "k", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFITT[7:0]							CFTML[3:0]				CFITR	CFITSS	CFM[1:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFIGCV[2:0]			CFIM	—	CFDC[2:0]			—	—	—	—	—	CFIXIE	CFRXIE	CFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 20.54 RSCANnCFCCk Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	CFITT[7:0]	Set a message transmission interval. Set value: 00 _H to FF _H
23 to 20	CFTML[3:0]	Transmit Buffer Link Configuration Set the transmit buffer number to be linked to the transmit/receive FIFO buffer.
19	CFITR	Transmit/Receive FIFO Interval Timer Resolution 0: Clock obtained by dividing pclk/2 by the value of the ITRCP [15:0] bits 1: Clock obtained by dividing pclk/2 by "the value of ITRCP [15:0] bits x 10"
18	CFITSS	Transmit/Receive FIFO Interval Timer Source Select 0: Clock selected by the CFITR bit (x1 / x10 cycles) 1: Bit time clock of the corresponding channel (FIFO is linked to the fixed channel.)
17 to 16	CFM[1:0]	Transmit/Receive FIFO Mode Select b17 b16 0 0: Receive mode 0 1: Transmit mode 1 0: Gateway mode 1 1: Setting prohibited
15 to 13	CFIGCV [2:0]	Transmit/Receive FIFO Receive Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.

Table 20.54 RSCANnCFCCk Register Contents (2/2)

Bit Position	Bit Name	Function
12	CFIM	Transmit/Receive FIFO Interrupt Source Select 0: <ul style="list-style-type: none"> • Receive mode / gateway mode When the number of received messages meets the condition set by the CFIGCV[2:0] bits, a FIFO receive interrupt request is generated. • Transmit mode / gateway mode When the buffer becomes empty upon completion of message transmission, a FIFO transmit interrupt request is generated. 1: <ul style="list-style-type: none"> • Receive mode / gateway mode A FIFO receive interrupt request is generated each time a message has been received. • Transmit mode / gateway mode FIFO transmit interrupt request is generated each time a message has been transmitted.
11	—	Reserved When read, the value after reset is returned. When writing to this bit, write the value after reset.
10 to 8	CFDC [2:0]	Transmit/Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
7 to 3	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	CFTXIE	Transmit/Receive FIFO Transmit Interrupt Enable 0: Transmit/receive FIFO transmit interrupt is disabled. 1: Transmit/receive FIFO transmit interrupt is enabled.
1	CFRXIE	Transmit/Receive FIFO Receive Interrupt Enable 0: Transmit/receive FIFO receive interrupt is disabled. 1: Transmit/receive FIFO receive interrupt is enabled.
0	CFE	Transmit/Receive FIFO Buffer Enable 0: No transmit/receive FIFO buffer is used. 1: Transmit/receive FIFO buffers are used.

CFITT[7:0] Bits

These bits are used to set a message transmission interval for continuous message transmission from a transmit/receive FIFO buffer whose CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode).

Clear the CFE bit to 0 (no transmit/receive FIFO buffer is used) before modifying the CFITT[7:0] bits.

CFTML[3:0] Bits

These bits are used to set the number of the transmission buffer to be linked to the transmission and reception FIFO buffers when the CFM[1:0] bits are set to 01_B (transmission mode) or 10_B (gateway mode). Three transmission and reception FIFO buffers are available per channel and channel number *m* to which FIFO buffer *k* is allocated is the integer part of the quotient $k/3$. The number *p* of the transmission buffer to be linked to FIFO buffer *k* will be $(16 \times m) + \text{CFTML}[3:0]$. (see **Table 20.19**)

For the relationship between transmission and reception FIFO buffer k and transmission buffer p , see **Table 20.17** and **Table 20.18**. Setting the CFDC[2:0] bits to 001_B or greater enables the setting of the CFTML[3:0] bits.

Do not make a link to the transmission queue of the same channel or any transmission buffer allocated to the transmission queue. Modify these bits in global reset mode.

CFITR Bit

This bit is enabled when the CFITSS bit is 0.

When this bit is 0, the interval timer clock source is selected as the $pelk/2$ divided by the value of the ITRCP [15:0] bits of the RSCANnGCFG register.

When this bit is 1, the interval timer clock source is selected as the $pelk/2$ divided by (ITRCP [15:0] bits \times 10) of the RSCANnGCFG register.

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFITSS Bit

When this bit is 0, the clock selected by the CFITR bit is the count source of the interval timer.

When this bit is 1, the bit time clock of the channel that is linked to FIFO is the count source of the interval timer.

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFM[1:0] Bits

These bits are used to select a transmit/receive FIFO mode. Modify these bits in global reset mode.

CFIGCV[2:0] Bits

These bits are used to specify the number of received messages required to generate a transmit/receive FIFO receive interrupt request when the CFM[1:0] bits are set to 00_B (receive mode) or 10_B (gateway mode) and the CFIM bit is set to 0 with a fraction for the total number of buffers (the setting of CFDC[2:0]).

When the CFDC[2:0] bits are set to 001_B (4 messages), set the CFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B.

Modify these bits only in global reset mode.

CFIM Bits

This bit is used to select a transmit/receive FIFO interrupt source. Modify this bit in global reset mode.

CFDC[2:0] Bit

These bits are used to set the number of messages that can be stored in a single transmit/receive FIFO buffer. If no transmit/receive FIFO buffer is used, set these bits to 000_B. Modify these bits only in global reset mode.

CFTXIE Bit

When this bit is set to 1 and the CFTXIF flag in the CFSTSk register becomes 1, a transmit/receive FIFO transmit interrupt request is generated.

Modify this bit when the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFRXIE Bit

When this bit is set to 1 and the CFTXIF flag in the RSCANnCFSTSk register becomes 1, a transmit/receive FIFO receive interrupt request is generated.

Modify this bit when the CFE bit is set to 0.

CFE Bit

Setting this bit to 1 enables the use of transmit/receive FIFO buffers.

When this bit is set to 0 in transmit mode or gateway mode, if a message in the transmit/receive FIFO buffer is being transmitted or to be transmitted next, the transmit/receive FIFO buffer becomes empty after completion of transmission,

CAN bus error detection, or arbitration lost. In other cases or in receive mode, the transmit/receive FIFO buffer becomes empty immediately.

This bit is cleared to 0 when the following conditions are met.

- Receive mode: Global reset mode
- Transmit mode or gateway mode: Channel reset mode

Modify this bit in the following mode.

- Receive mode: Global operating mode or global test mode
- Transmit mode or gateway mode: Channel communication mode or channel halt mode

After all other bits in the RSCANnCFCCk register have been set, set this bit to 1 by using another instruction.

20.3.8.2 RSCANnCFSTSk — Transmit/Receive FIFO Buffer Status Register k

Note: For the value of index "k", see **Table 20.6**.

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFMC[7:0]								—	—	—	CFTXIF	CFRXIF	CFMLT	CFLL	CFEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the status. Any other writing to the bit results in retention of the status and does not change the value.

Table 20.55 RSCANnCFSTSk Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
15 to 8	CFMC [7:0]	Transmit/Receive FIFO Message Counter The number of messages stored in the transmit/receive FIFO buffer is indicated.
7 to 5	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
4	CFTXIF	Transmit/Receive FIFO Transmit Interrupt Request Flag 0: No transmit/receive FIFO transmit interrupt request is present. 1: A transmit/receive FIFO transmit interrupt request is present.
3	CFRXIF	Transmit/Receive FIFO Receive Interrupt Request Flag 0: No transmit/receive FIFO receive interrupt request is present. 1: A transmit/receive FIFO receive interrupt request is present.
2	CFMLT	Transmit/Receive FIFO Message Lost Flag 0: No transmit/receive FIFO message is lost. 1: A transmit/receive FIFO message is lost.
1	CFLL	Transmit/Receive FIFO Buffer Full Status Flag 0: The transmit/receive FIFO buffer is not full. 1: The transmit/receive FIFO buffer is full.
0	CFEMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: The transmit/receive FIFO buffer contains messages. 1: The transmit/receive FIFO buffer contains no message (buffer empty).

CFMC[7:0] Bits

The CFMC[7:0] bits indicate the following values according to the setting of the CFM[1:0] bits in the RSCANnCFCCk register.

- When CFM[1:0] value is 01_B (transmit mode): Number of untransmitted messages in the buffer
- When CFM[1:0] value is 00_B (receive mode): Number of unread received messages in the buffer
- When CFM[1:0] value is 10_B (gateway mode): Number of untransmitted received messages in the buffer

These bits are cleared to 0 when any of the following conditions is met.

- When CFM[1:0] value is 00_B: global reset mode
- When CFM[1:0] value is 01_B or 10_B: channel reset mode
- When the CFE bit in the RSCAN_nCFCCk register is cleared to 0.

CFTXIF Flag

The CFTXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] value is 01_B or 10_B and the interrupt source selected by the CFIM bit in the RSCAN_nCFCCk register occurred.

The CFTXIF flag is set to 0 when any of the following conditions is met.

- When 0 is written in the CFTXIF flag.
- When the CFM[1:0] value is 00_B: global reset mode
- When the CFM[1:0] is 01_B or 10_B: channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flag of the register to 0, the program must write 0 to the given flag. When writing 0, use a store instruction to write 0 to the given flag and 1 to the other flags.

CFRXIF Flag

The CFRXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] value is 00_B or 10_B and the interrupt source selected by the CFIM bit in the RSCAN_nCFCCk register occurred.

The CFRXIF flag is set to 0 when any of the following conditions is met.

- When 0 is written in the CFRXIF flag.
- When the CFM[1:0] value is 00_B: global reset mode
- When the CFM[1:0] is 01_B or 10_B: channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flag of the register to 0, the program must write 0 to the given flag. When writing 0, use a store instruction to write 0 to the given flag and 1 to the other flags.

CFMLT Flag

The CFMLT flag is set to 1 when any of the following conditions is met.

- When transmit or receive FIFO buffer is full, and a new message is stored in this register.
In this case, a new message is terminated.

The CFMLT flag is set to 0 when any of the following conditions is met.

- When 0 is written in the CFMLT flag.
- When the CFM[1:0] value is 00_B: global reset mode
- When the CFM[1:0] is 01_B or 10_B: channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flag of the register to 0, the program must write 0 to the given flag. When writing 0, use a store instruction to write 0 to the given flag and 1 to the other flags.

CFLL Flag

The CFLL flag is set to 1 when any of the following conditions is met.

- When the number of messages stored in the transmit and receive FIFO buffer is equal to the FIFO buffer stages set by the CFDC[2:0] bits in the RSCANnCFCCk register

The CFLL flag is set to 0 when any of the following conditions is met.

- When the number of messages stored in the transmit and receive FIFO buffer is smaller than the FIFO buffer stages set by the CFDC[2:0] bits.
- When the CFE bit is 0 (no transmit/receive FIFO buffer is used): Not during transmit abort
- When the CFM[1:0] value is 00_B: global reset mode
- When the CFM[1:0] is 01_B or 10_B: channel reset mode

CFEMP Flag

The CFEMP flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] value is 00_B: All messages have been read or global reset mode.
- When the CFM[1:0] value is 01_B or 10_B: All messages have been transmitted or channel reset mode.
- When the CFE bit is 0 (no transmit/receive FIFO buffer is used): Not during transmit abort

The CFEMP flag is cleared to 0 when any of the following conditions is met.

- When the CFM[1:0] value is 00_B or 10_B: Any one of received messages has been stored in the transmit/receive FIFO buffer.
- When the CFM[1:0] value is 01_B: A value FF_H has been written to the RSCANnCFPCTRk register after data was written to the RSCANnCFIDk, RSCANnCFPTRk, RSCANnCFDF0k, and RSCANnCFDF1k registers.

NOTE

To clear the CFTXIF, CFRXIF, and CFMLT flags to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using a store instruction, set 1 to other flags.

20.3.8.3 RSCANnCFPCTRk — Transmit/Receive FIFO Buffer Pointer Control Register k

Note: For the value of index "k", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 20.56 RSCANnCFPCTRk Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When writing to these bits, write the value after reset.
7 to 0	CFPC [7:0]	Transmit/Receive FIFO Pointer Control <ul style="list-style-type: none"> • Receive mode: Writing FF_H to these bits move the read pointer to the next unread message in the transmit/receive FIFO buffer. • Transmit mode: Writing FF_H to these bits move the write pointer to the next stage of the transmit/receive FIFO buffer. • Gateway mode: Setting prohibited

CFPC[7:0] Bits

- Receive mode (CFM[1:0] value in the RSCANnCFCCk register is 00_B):

Writing FF_H to the CFPC[7:0] bits move the read pointer to the next unread message in the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value (transmit/receive FIFO message counter) in the RSCANnCFSTSk register is decremented by 1. Read the RSCANnCFIDk, RSCANnCFPTRk, RSCANnCFDF0_k, and RSCANnCFDF1_k registers to read messages in the transmit/receive FIFO buffer, and then write FF_H to the CFPC[7:0] bits.

When writing FF_H to these bits, make sure that the CFE bit in the RSCANnCFCCk register is set to 1 (transmit/receive FIFO buffers are used) and the CFEMP flag in the RSCANnCFSTSk register is 0 (the transmit/receive FIFO buffer contains messages).

- Transmit mode (CFM[1:0] value in the RSCANnCFCCk register is 01_B):

Writing FF_H to the CFPC[7:0] bits store the data written to the RSCANnCFIDk, RSCANnCFPTRk, RSCANnCFDF0_k, and RSCANnCFDF1_k registers in the transmit/receive FIFO buffer and move the write pointer to the next stage of the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value is incremented by 1. Write transmit messages to the RSCANnCFIDk, RSCANnCFPTRk, RSCANnCFDF0_k, and RSCANnCFDF1_k registers and then write FF_H to the CFPC[7:0] bits.

When writing FF_H to these bits, make sure that the CFE bit in the RSCANnCFCCk register is set to 1 and the CFLL flag in the RSCANnCFSTSk register is 0 (the transmit/receive FIFO buffer is not full).

- Gateway mode (CFM[1:0] value in the RSCANnCFCCk register is 10_B):

Setting prohibited

20.3.8.4 RSCANnCFIDk — Transmit/Receive FIFO Buffer Access ID Register k

Note: For the value of index "k", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFIDE	CFRTR	THLEN	CFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.57 RSCANnCFIDk Register Contents

Bit Position	Bit Name	Function
31	CFIDE	Transmit/Receive FIFO Buffer IDE bit 0: Standard ID 1: Extended ID
30	CFRTR	Transmit/Receive FIFO Buffer RTR bit 0: Data frame 1: Remote frame
29	THLEN	Transmit History Data Store Enable bit This bit is valid only when the CFM[1:0] value is 01 _B (transmit mode). 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	CFID [28:0]	Transmit/Receive FIFO Buffer ID Data <ul style="list-style-type: none"> When CFM[1:0] value is 01_B (transmit mode): Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11. When CFM[1:0] value is 00_B (receive mode): Standard ID or extended ID in the received message can be read. For standard ID, read bits 10 to 0. Bits 28 to 11 are read as 0.

This register can be written only when the CFM[1:0] value of the RSCANnCFCCk register is 01_B (transmit mode). This register can be read only when the CFM[1:0] value is 00_B (receive mode). This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, this bit is used to set the ID format of the message to be transmitted from the transmit/receive FIFO buffer.

CFRTR Bit

This bit indicates the data format (data frame or remote frame) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, set the data format of the message to be transmitted from the transmit/receive FIFO buffer.

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, and buffer type) of transmitted messages is stored in the transmit history buffer after transmission is completed.

This bit is enabled when the CFM[1:0] value is 01_B (transmit mode).

CFID[28:0] Bits

These bits indicate the ID of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B.

These bits set the ID of the message to be transmitted from the transmit/receive FIFO buffer when the CFM[1:0] value is 01_B.

20.3.8.5 RSCANnCFPTRk — Transmit/Receive FIFO Buffer Access Pointer Register k

Note: For the value of index "k", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDLC[3:0]				CFPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.58 RSCANnCFPTRk Register Contents

Bit Position	Bit Name	Function
31 to 28	CFDLC [3:0]	Transmit/Receive FIFO Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	CFPTR [11:0]	Transmit/Receive FIFO Buffer Label Data <ul style="list-style-type: none"> When CFM[1:0] value is 01_B (transmit mode): Set the label information to be stored in the transmit history buffer. Only CFPTR[7:0] bits are valid. When CFM[1:0] value is 00_B (receive mode): The label information of the received message can be read.
15 to 0	CFTS [15:0]	Transmit/Receive FIFO Buffer Timestamp Data These bits are valid only when the CFM[1:0] value is 00 _B (receive mode). The timestamp value of the received message can be read.

This register is writable only when the CFM[1:0] value of the RSCANnCFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFDLC[3:0] Bits

These bits indicate the data length of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, set the data length of the message to be transmitted from the transmit/receive FIFO buffer. If 9-byte or more data length is set, 8-byte data is transmitted.

CFPTR[11:0] Bits

These bits indicate the label information attached to the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, the CFPTR[7:0] value is stored in the transmit history buffer when message transmission has been completed.

CFTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the transmit/receive FIFO buffer.

These bits are valid when the CFM[1:0] value is 00_B.

20.3.8.6 RSCANnCFDF0_k — Transmit/Receive FIFO Buffer Access Data Field 0 Register k

Note: For the value of index "k", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB3[7:0]								CFDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB1[7:0]								CFDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.59 RSCANnCFDF0_k Register Contents

Bit Position	Bit Name	Function
31 to 24	CFDB3 [7:0]	Transmit/Receive FIFO Buffer Data Byte 3
23 to 16	CFDB2 [7:0]	Transmit/Receive FIFO Buffer Data Byte 2
15 to 8	CFDB1 [7:0]	Transmit/Receive FIFO Buffer Data Byte 1
7 to 0	CFDB0 [7:0]	Transmit/Receive FIFO Buffer Data Byte 0 <ul style="list-style-type: none"> • When CFM[1:0] value is 01_B (transmit mode): Set the data of the transmit/receive FIFO buffer. • When CFM[1:0] value is 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.

This register can be written only when the CFM[1:0] value of the RSCANnCFCCk register is 01_B (transmit mode). This register can be read only when the CFM[1:0] value is 00_B (receive mode). When the CFDLC[3:0] value in the RSCANnCFPTRk register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

20.3.8.7 RSCANnCFDF1_k — Transmit/Receive FIFO Buffer Access Data Field 1 Register k

Note: For the value of index "k", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB7[7:0]								CFDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB5[7:0]								CFDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.60 RSCANnCFDF1_k Register Contents

Bit Position	Bit Name	Function
31 to 24	CFDB7 [7:0]	Transmit/Receive FIFO Buffer Data Byte 7
23 to 16	CFDB6 [7:0]	Transmit/Receive FIFO Buffer Data Byte 6
15 to 8	CFDB5 [7:0]	Transmit/Receive FIFO Buffer Data Byte 5
7 to 0	CFDB4 [7:0]	Transmit/Receive FIFO Buffer Data Byte 4 <ul style="list-style-type: none"> When CFM[1:0] value is 01_B (transmit mode): Set the data of the transmit/receive FIFO buffer. When CFM[1:0] value is 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.

This register is writable only when the CFM[1:0] value of the RSCANnCFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). When the CFDLc[3:0] value in the RSCANnCFPTRk register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode)

20.3.9 Details of FIFO Staus-Related Registers

20.3.9.1 RSCANnFESTS — FIFO Empty Status Register

Value after reset: 03FF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CF14EMP	CF13EMP	CF12EMP	CF11EMP	CF10EMP	CF9EMP	CF8EMP
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7EMP	CF6EMP	CF5EMP	CF4EMP	CF3EMP	CF2EMP	CF1EMP	CF0EMP	RF7EMP	RF6EMP	RF5EMP	RF4EMP	RF3EMP	RF2EMP	RF1EMP	RF0EMP
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.61 RSCANnFESTS Register Contents

Bit Position	Bit Name	Function
31 to 23	—	Reserved When read, the value after reset is returned.
22	CF14EMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: Transmit/receive FIFO buffer k contains a message. 1: Transmit/receive FIFO buffer k contains no message. (k = see Table 20.6)
21	CF13EMP	
20	CF12EMP	
19	CF11EMP	
18	CF10EMP	
17	CF9EMP	
16	CF8EMP	
15	CF7EMP	
14	CF6EMP	Receive FIFO Buffer Empty Status Flag 0: Receive FIFO buffer x contains an unread message. 1: Receive FIFO buffer x contains no unread message. (x = see Table 20.6)
13	CF5EMP	
12	CF4EMP	
11	CF3EMP	
10	CF2EMP	
9	CF1EMP	
8	CF0EMP	
7	RF7EMP	
6	RF6EMP	Receive FIFO Buffer Empty Status Flag 0: Receive FIFO buffer x contains an unread message. 1: Receive FIFO buffer x contains no unread message. (x = see Table 20.6)
5	RF5EMP	
4	RF4EMP	
3	RF3EMP	
2	RF2EMP	
1	RF1EMP	
0	RF0EMP	

The RSCANnFESTS register is set to 003F FFFF_H in global reset mode.

CFkEMP Flag (k = see Table 20.6)

The CFkEMP flag is set to 1 when the CFEMP flag in the RSCANnCFSTSk register is set to 1 (the transmit/receive FIFO buffer contains no message). When the CFEMP flag is cleared to 0 (the transmit/receive FIFO buffer contains messages), the CFkEMP flag is cleared to 0.

RFxEMP Flag (x = see Table 20.6)

The RFxEMP flag is set to 1 when the RFEMP flag in the RSCANnRFSTStx register is set to 1 (the receive FIFO buffer contains no unread message). When the RFEMP flag is cleared to 0 (the receive FIFO buffer contains unread messages), the RFxEMP flag is cleared to 0.

20.3.9.2 RSCANnFFSTS — FIFO Full Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CF14FL L	CF13FL L	CF12FL L	CF11FL L	CF10FL L	CF9FLL	CF8FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7FLL	CF6FLL	CF5FLL	CF4FLL	CF3FLL	CF2FLL	CF1FLL	CF0FLL	RF7FLL	RF6FLL	RF5FLL	RF4FLL	RF3FLL	RF2FLL	RF1FLL	RF0FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.62 RSCANnFFSTS Register Contents

Bit Position	Bit Name	Function
31 to 23	—	Reserved When read, the value after reset is returned.
22	CF14FLL	Transmit/Receive FIFO Buffer Full Status Flag 0: Transmit/receive FIFO buffer k is not full. 1: Transmit/receive FIFO buffer k is full. (k = see Table 20.6)
21	CF13FLL	
20	CF12FLL	
19	CF11FLL	
18	CF10FLL	
17	CF9FLL	
16	CF8FLL	
15	CF7FLL	
14	CF6FLL	
13	CF5FLL	Receive FIFO Buffer Full Status Flag 0: Receive FIFO buffer x is not full. 1: Receive FIFO buffer x is full. (x = see Table 20.6)
12	CF4FLL	
11	CF3FLL	
10	CF2FLL	
9	CF1FLL	
8	CF0FLL	
7	RF7FLL	
6	RF6FLL	
5	RF5FLL	
4	RF4FLL	
3	RF3FLL	
2	RF2FLL	
1	RF1FLL	
0	RF0FLL	

The RSCANnFFSTS register is cleared to 0000 0000_H in global reset mode.

CFkFLL Flag (k = see Table 20.6)

The CFkFLL flag is set to 1 when the CFFLL flag in the RSCANnCFSTSk register is set to 1 (the transmit/receive FIFO buffer is full). When the CFFLL flag is cleared to 0, the CFkFLL flag is cleared to 0 (the transmit/receive FIFO buffer is not full).

RFxFLl Flag (x = see Table 20.6)

The RFxFLl flag is set to 1 when the RFFLL flag in the RSCANnRFSTSx register is set to 1 (the receive FIFO buffer is full). When the RFFLL flag is cleared to 0 (the receive FIFO buffer is not full), the RFxFLl flag is cleared to 0.

20.3.9.3 RSCANnFMSTS — FIFO Message Lost Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CF14MLT	CF13MLT	CF12MLT	CF11MLT	CF10MLT	CF9MLT	CF8MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7MLT	CF6MLT	CF5MLT	CF4MLT	CF3MLT	CF2MLT	CF1MLT	CF0MLT	RF7MLT	RF6MLT	RF5MLT	RF4MLT	RF3MLT	RF2MLT	RF1MLT	RF0MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.63 RSCANnFMSTS Register Contents

Bit Position	Bit Name	Function
31 to 23	—	Reserved When read, the value after reset is returned.
22	CF14MLT	Transmit/Receive FIFO Buffer Message Lost Status Flag 0: No transmit/receive FIFO buffer k message is lost. 1: A transmit/receive FIFO buffer k message is lost. (k = see Table 20.6)
21	CF13MLT	
20	CF12MLT	
19	CF11MLT	
18	CF10MLT	
17	CF9MLT	
16	CF8MLT	
15	CF7MLT	
14	CF6MLT	Receive FIFO Buffer Message Lost Status Flag 0: No receive FIFO buffer x message is lost. 1: A receive FIFO buffer x message is lost. (x = see Table 20.6)
13	CF5MLT	
12	CF4MLT	
11	CF3MLT	
10	CF2MLT	
9	CF1MLT	
8	CF0MLT	
7	RF7MLT	
6	RF6MLT	Receive FIFO Buffer Message Lost Status Flag 0: No receive FIFO buffer x message is lost. 1: A receive FIFO buffer x message is lost. (x = see Table 20.6)
5	RF5MLT	
4	RF4MLT	
3	RF3MLT	
2	RF2MLT	
1	RF1MLT	
0	RF0MLT	

The RSCANnFMSTS register is cleared to 0000 0000_H in global reset mode.

CFkMLT Flag (k = see Table 20.6)

The CFkMLT flag is set to 1 when the CFMLT flag in the RSCANnCFSTSk register is set to 1 (a transmit/receive FIFO message is lost). When the CFMLT flag is cleared to 0, the CFkMLT flag is cleared to 0.

RFxMLT Flag (x = see Table 20.6)

The RFxMLT flag is set to 1 when the RFMLT flag in the RSCANnRFSTSx register is set to 1 (a receive FIFO message is lost). When the RFMLT flag is cleared to 0, the RFxMLT flag is cleared to 0.

20.3.9.4 RSCANnRFISTS — Receive FIFO Buffer Interrupt Flag Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RF7IF	RF6IF	RF5IF	RF4IF	RF3IF	RF2IF	RF1IF	RF0IF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.64 RSCANnRFISTS Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned.
7	RF7IF	Receive FIFO Buffer Interrupt Request Status Flag
6	RF6IF	0: No receive FIFO buffer x interrupt request is present.
5	RF5IF	1: A receive FIFO buffer x interrupt request is present.
4	RF4IF	(x = see Table 20.6)
3	RF3IF	
2	RF2IF	
1	RF1IF	
0	RF0IF	

The RSCANnRFISTS register is cleared to 0000 0000_H in global reset mode.

RFxIF Flag (x = see Table 20.6)

The RFxIF flag is set to 1 when the RFIF flag in the RSCANnRFISTSx register is set to 1 (a receive FIFO interrupt request is present). When the RFIF flag is cleared to 0, the RFxIF flag is cleared to 0.

20.3.9.5 RSCANnCFRISTS — Transmit/Receive FIFO Buffer Receive Interrupt Flag Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	CF14RXIF	CF13RXIF	CF12RXIF	CF11RXIF	CF10RXIF	CF9RXIF	CF8RXIF	CF7RXIF	CF6RXIF	CF5RXIF	CF4RXIF	CF3RXIF	CF2RXIF	CF1RXIF	CF0RXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.65 RSCANnCFRISTS Register Contents

Bit Position	Bit Name	Function
31 to 15	—	Reserved When read, the value after reset is returned.
14	CF14RXIF	Transmit/Receive FIFO Buffer Receive Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k receive interrupt request is present. 1: A transmit/receive FIFO buffer k receive interrupt request is present. (k = see Table 20.6)
13	CF13RXIF	
12	CF12RXIF	
11	CF11RXIF	
10	CF10RXIF	
9	CF9RXIF	
8	CF8RXIF	
7	CF7RXIF	
6	CF6RXIF	
5	CF5RXIF	
4	CF4RXIF	
3	CF3RXIF	
2	CF2RXIF	
1	CF1RXIF	
0	CF0RXIF	

The RSCANnCFRISTS register is cleared to 0000 0000_H in global reset mode.

CFxRXIF Flag (k = see Table 20.6)

The CFxRXIF flag is set to 1 when the CFRXIF flag in the RSCANnCFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). When the CFRXIF flag is cleared to 0, the CFxRXIF flag is cleared to 0.

20.3.9.6 RSCANnCFSTIS — Transmit/Receive FIFO Buffer Transmit Interrupt Flag Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	CF14TXIF	CF13TXIF	CF12TXIF	CF11TXIF	CF10TXIF	CF9TXIF	CF8TXIF	CF7TXIF	CF6TXIF	CF5TXIF	CF4TXIF	CF3TXIF	CF2TXIF	CF1TXIF	CF0TXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.66 RSCANnCFSTIS Register Contents

Bit Position	Bit Name	Function
31 to 15	—	Reserved When read, the value after reset is returned.
14	CF14TXIF	Transmit/Receive FIFO Buffer Transmit Interrupt Request Status Flag
13	CF13TXIF	0: No transmit/receive FIFO buffer k transmit interrupt request is present.
12	CF12TXIF	1: A transmit/receive FIFO buffer k transmit interrupt request is present.
11	CF11TXIF	(k = see Table 20.6)
10	CF10TXIF	
9	CF9TXIF	
8	CF8TXIF	
7	CF7TXIF	
6	CF6TXIF	
5	CF5TXIF	
4	CF4TXIF	
3	CF3TXIF	
2	CF2TXIF	
1	CF1TXIF	
0	CF0TXIF	

The RSCANnCFSTIS register is cleared to 0000 0000_H in global reset mode.

CFkTXIF Flag (k = see Table 20.6)

The CFkTXIF flag is set to 1 when the CFTXIF flag in the RSCANnCFSTSk register is set to 1 (a transmit/receive FIFO transmit interrupt request is present). When the CFTXIF flag is cleared to 0, the CFkTXIF flag is cleared to 0.

20.3.10 Details of Transmit Buffer-Related Registers

20.3.10.1 RSCANnTMCp — Transmit Buffer Control Register p

Note: For the value of index "p", see **Table 20.6**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	TMOM	TMTAR	TMTR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W ^{*1}	R/W ^{*1}

Note 1. The only effective value for writing to this bit is 1, which sets the bit. Any other writing to the bit results in retention of the status and does not change the value.

Table 20.67 RSCANnTMCp Register Contents

Bit Position	Bit Name	Function
7 to 3	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	TMOM	One-Shot Transmission Enable 0: One-shot transmission is disabled. 1: One-shot transmission is enabled.
1	TMTAR	Transmit Abort Request 0: Transmit abort is not requested. 1: Transmit abort is requested.
0	TMTR	Transmit Request 0: Transmission is not requested. 1: Transmission is requested.

When the RSCANnTMCp register meets any of the following conditions, set it to 00_H.

- The RSCANnTMCp register ($p = m \times 16 +$ the value of the CFTML[3:0] bits) corresponds to the transmit buffer number selected by the CFTML[3:0] bits in the RSCANnCFCCk register.
- The RSCANnTMCp register ($p = (m \times 16 + 15)$ to $(m \times 16 + 15 +$ the value of the TXQDC[3:0] bits)) corresponds to the transmit buffer allocated to the transmit queue by the TXQDC[3:0] bits in the RSCANnTXQCCm register ($m =$ see **Table 20.6**).

All of the bits in the RSCANnTMCp register are 0 in channel reset mode. Modify the RSCANnTMCp register in channel communication mode or channel halt mode.

TMOM Bit

Setting this bit to 1 enables one-shot transmission. If transmission fails, retransmission defined in the CAN protocol is not performed.

Modify the TMOM bit when the TMTRM flag in the RSCANnTMSTSp register is set to 0. To set the TMOM bit to 1, also set the TMTR bit together.

TMTAR Bit

Setting this bit to 1 generates a transmit abort request for the message stored in the transmit buffer. However, a message that is being transmitted or to be transmitted next cannot be aborted.

When the TMTR bit is set to 1, the TMTAR bit can be set to 1.

The TMTAR bit is cleared to 0 when any of the following conditions is set. It is not cleared to 0 by writing 0 by the program.

- Transmission has been completed.
- Transmit abort has been completed.
- An error or arbitration-lost has been detected.

If this bit becomes 0 at the timing when the program writes 1 to this bit, this bit becomes 0.

TMTR Bit

Setting this bit to 1 transmits the message stored in the transmit buffer.

The TMTR bit is cleared to 0 when any of the following conditions is met. It is not cleared to 0 by writing 0 by the program.

- Transmission has been completed.
- Transmit abort has been completed by setting the TMTAR bit to 1.
- An error or arbitration lost has been detected with the TMOM bit set to 1.

Set the TMTR bit to 1 when the TMTRF[1:0] flag in the RSCANnTMSTSp register is 00_B.

20.3.10.2 RSCANnTMSTSp — Transmit Buffer Status Register p

Note: For the value of index "p", see **Table 20.6**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	TMTARM	TMTRM	TMTRF[1:0]		TMTSTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R

Table 20.68 RSCANnTMSTSp Register Contents

Bit Position	Bit Name	Function
7 to 5	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
4	TMTARM	Transmit Buffer Transmit Abort Request Status Flag 0: No transmit abort request is present. 1: A transmit abort request is present.
3	TMTRM	Transmit Buffer Transmit Request Status Flag 0: No transmit request is present. 1: A transmit request is present.
2,1	TMTRF[1:0]	Transmit Buffer Transmit Result Flag b2 b1 0 0: Transmission is in progress or no transmit request is present. 0 1: Transmit abort has been completed. 1 0: Transmission has been completed (without transmit abort request). 1 1: Transmission has been completed (with transmit abort request).
0	TMTSTS	Transmit Buffer Transmit Status Flag 0: Transmission is not in progress. 1: Transmission is in progress.

All of the bits in the RSCANnTMSTSp register are cleared to 0 in channel reset mode.

TMTARM Flag

The TMTARM flag is set to 1 when the TMTAR bit in the RSCANnTMCp register is set to 1, and is cleared to 0 when the TMTAR bit is set to 0.

TMTRM Flag

The TMTRM flag is set to 1 when the TMTR bit in the RSCANnTMCp register is set to 1, and is cleared to 0 when the TMTR bit is set to 0.

TMTRF[1:0] Flag

This flag indicates the result of transmission from the transmit buffer.

00_B: Transmission is in progress or no transmit request is present.

01_B: Transmission from the transmit buffer was aborted.

10_B: Transmission has been completed with the TMTAR bit in the RSCANnTMCp register set to 0 (transmit abort is not requested).

11_B: Transmission has been completed with the TMTAR bit in the RSCANnTMCp register set to 1 (transmit abort is requested).

Write 00_B to the TMTRF[1:0] flag in channel communication mode or channel halt mode. Do not write any value other than 00_B to this flag.

TMTSTS Flag

This flag is set to 1 when transmission from the transmit buffer starts, and is cleared to 0 when transmission from the transmit buffer has been completed or stopped due to a bus error or arbitration lost.

20.3.10.3 RSCANnTMIDp — Transmit Buffer ID Register p

Note: For the value of index "p", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIDE	TMRTR	THLEN	TMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.69 RSCANnTMIDp Register Contents

Bit Position	Bit Name	Function
31	TMIDE	Transmit Buffer IDE 0: Standard ID 1: Extended ID
30	TMRTR	Transmit Buffer RTR 0: Data frame 1: Remote frame
29	THLEN	Transmit History Data Store Enable 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	TMID [28:0]	Transmit Buffer ID Data Set standard ID or extended ID. For standard ID, write an ID to bits b10 to b0 and write 0 to bits b28 to b11.

Modify this register when the TMTRM bit in the corresponding RSCANnTMSTSp register is set to 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register. Write data only to the transmit buffer p ($p = m \times 16 + 15$) when this register is allocated to the transmit queue.

TMIDE Bit

This bit is used to set the ID format of the message to be transmitted from the transmit buffer.

TMRTR Bit

This bit is used to set the data format of the message to be transmitted from the transmit buffer.

THLEN Bit

Setting this bit to 1 stores a transmitted message in the transmit history buffer after transmission is completed.

TMID[28:0] Bits

These bits are used to set the ID of the message to be transmitted from the transmit buffer.

20.3.10.4 RSCANnTMPTRp — Transmit Buffer Pointer Register p

Note: For the value of index "p", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDLC[3:0]				—	—	—	—	TMPTR[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.70 RSCANnTMPTRp Register Contents

Bit Position	Bit Name	Function
31 to 28	TMDLC [3:0]	Transmit Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 x x x: 8 data bytes
27 to 24	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
23 to 16	TMPTR [7:0]	Transmit Buffer Label Data Set the label information to be stored in the transmit history buffer.
15 to 0	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.

Modify this register when the TMTRM bit in the corresponding RSCANnTMSTSp register is set to 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register. Write data only to the transmit buffer p ($p = m \times 16 + 15$) when this register is allocated to the transmit queue.

TMDLC[3:0] Bits

These bits are used to set the data length of the message to be transmitted from the transmit buffer when the TMRTR bit in the RSCANnTMIDp register is set to 0 (data frame). If the setting for the data length is 1001_B or above, 8-byte data is transmitted.

When the TMRTR bit is set to 1 (remote frame), these bits set the data length of the message to be requested.

TMPTR[7:0] Bits

When message transmission has been completed, the TMPTR[7:0] value is stored in the transmit history buffer.

20.3.10.5 RSCANnTMDF0_p — Transmit Buffer Data Field 0 Register p

Note: For the value of index "p", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB3[7:0]								TMDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB1[7:0]								TMDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.71 RSCANnTMDF0_p Register Contents

Bit Position	Bit Name	Function
31 to 24	TMDB3 [7:0]	Transmit Buffer Data Byte 3
23 to 16	TMDB2 [7:0]	Transmit Buffer Data Byte 2
15 to 8	TMDB1 [7:0]	Transmit Buffer Data Byte 1
7 to 0	TMDB0 [7:0]	Transmit Buffer Data Byte 0
Set transmit buffer data.		

Modify this register when the TMTRM bit in the corresponding RSCANnTMSTSp register is set to 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register. Write data only to the transmit buffer p ($p = m \times 16 + 15$) when this register is allocated to the transmit queue.

20.3.10.6 RSCANnTMDF1_p — Transmit Buffer Data Field 1 Register p

Note: For the value of index "p", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB7[7:0]								TMDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB5[7:0]								TMDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.72 RSCANnTMDF1_p Register Contents

Bit Position	Bit Name	Function
31 to 24	TMDB7 [7:0]	Transmit Buffer Data Byte 7
23 to 16	TMDB6 [7:0]	Transmit Buffer Data Byte 6
15 to 8	TMDB5 [7:0]	Transmit Buffer Data Byte 5
7 to 0	TMDB4 [7:0]	Transmit Buffer Data Byte 4
Set transmit buffer data.		

Modify this register when the TMTRM bit in the corresponding RSCANnTMSTSp register is set to 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register. Write data only to the transmit buffer p ($p = m \times 16 + 15$) when this register is allocated to the transmit queue.

20.3.10.7 RSCANnTMIECy — Transmit Buffer Interrupt Enable Configuration Register y

Note: For the value of index "y", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIEp (p = y × 32 + 31)	TMIEp (p = y × 32 + 30)	TMIEp (p = y × 32 + 29)	TMIEp (p = y × 32 + 28)	TMIEp (p = y × 32 + 27)	TMIEp (p = y × 32 + 26)	TMIEp (p = y × 32 + 25)	TMIEp (p = y × 32 + 24)	TMIEp (p = y × 32 + 23)	TMIEp (p = y × 32 + 22)	TMIEp (p = y × 32 + 21)	TMIEp (p = y × 32 + 20)	TMIEp (p = y × 32 + 19)	TMIEp (p = y × 32 + 18)	TMIEp (p = y × 32 + 17)	TMIEp (p = y × 32 + 16)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMIEp (p = y × 32 + 15)	TMIEp (p = y × 32 + 14)	TMIEp (p = y × 32 + 13)	TMIEp (p = y × 32 + 12)	TMIEp (p = y × 32 + 11)	TMIEp (p = y × 32 + 10)	TMIEp (p = y × 32 + 9)	TMIEp (p = y × 32 + 8)	TMIEp (p = y × 32 + 7)	TMIEp (p = y × 32 + 6)	TMIEp (p = y × 32 + 5)	TMIEp (p = y × 32 + 4)	TMIEp (p = y × 32 + 3)	TMIEp (p = y × 32 + 2)	TMIEp (p = y × 32 + 1)	TMIEp (p = y × 32 + 0)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.73 RSCANnTMIECy Register Contents

Bit Position	Bit Name	Function
31 to 0	TMIEp	Transmit Buffer Interrupt Enable bits p (p = y × 32 + 31 to y × 32 + 0) 0: Transmit buffer interrupt is disabled. 1: Transmit buffer interrupt is enabled.

TMIEp Bits (p = see Table 20.6)

When any of these bits is set to 1 and the corresponding transmission has been completed, a transmit buffer interrupt request is generated.

Modify these bits when the TMTRM flag in the corresponding RSCANnTMSTSp register is 0 (no transmit request is present).

Write 0 to the bits corresponding to the transmit buffers linked to transmit/receive FIFO buffers or the transmit buffers allocated to the transmit queue.

Table 20.74 shows the bit allocation.

Table 20.74 TMIEp Bit Allocation

Bit Position	Channel	Transmit Buffer No.
0	0	0
1	0	1
.	.	.
.	.	.
15	0	15
16	1	0
.	.	.
.	.	.
30	1	14
31	1	15
32	2	0
33	2	1
.	.	.
.	.	.
47	2	15
48	3	0
.	.	.
.	.	.
62	3	14
63	3	15

20.3.11 Details of Transmit Buffer Status-Related Registers

20.3.11.1 RSCANnTMTRSTSy — Transmit Buffer Transmit Request Status Register y

Note: For the value of index "y", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTRS TSp (p = y × 32 + 31)	TMTRS TSp (p = y × 32 + 30)	TMTRS TSp (p = y × 32 + 29)	TMTRS TSp (p = y × 32 + 28)	TMTRS TSp (p = y × 32 + 27)	TMTRS TSp (p = y × 32 + 26)	TMTRS TSp (p = y × 32 + 25)	TMTRS TSp (p = y × 32 + 24)	TMTRS TSp (p = y × 32 + 23)	TMTRS TSp (p = y × 32 + 22)	TMTRS TSp (p = y × 32 + 21)	TMTRS TSp (p = y × 32 + 20)	TMTRS TSp (p = y × 32 + 19)	TMTRS TSp (p = y × 32 + 18)	TMTRS TSp (p = y × 32 + 17)	TMTRS TSp (p = y × 32 + 16)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTRS TSp (p = y × 32 + 15)	TMTRS TSp (p = y × 32 + 14)	TMTRS TSp (p = y × 32 + 13)	TMTRS TSp (p = y × 32 + 12)	TMTRS TSp (p = y × 32 + 11)	TMTRS TSp (p = y × 32 + 10)	TMTRS TSp (p = y × 32 + 9)	TMTRS TSp (p = y × 32 + 8)	TMTRS TSp (p = y × 32 + 7)	TMTRS TSp (p = y × 32 + 6)	TMTRS TSp (p = y × 32 + 5)	TMTRS TSp (p = y × 32 + 4)	TMTRS TSp (p = y × 32 + 3)	TMTRS TSp (p = y × 32 + 2)	TMTRS TSp (p = y × 32 + 1)	TMTRS TSp (p = y × 32 + 0)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.75 RSCANnTMTRSTSy Register Contents

Bit Position	Bit Name	Function
31 to 0	TMTRSTSp	Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 31 to y × 32 + 0) 0: No transmit request is present. 1: A transmit request is present.

TMTRSTSp Flags (p = see Table 20.6)

These flags indicate the status of the TMTR bit in the RSCANnTMCp register.

When the TMTR bit is set to 1 (transmission is requested), the corresponding TMTRSTSp flag is set to 1.

The corresponding TMTRSTSp flag is cleared to 0 when the TMTR bit is set to 0 (transmission is not requested) or in channel reset mode.

Table 20.76 shows the bit allocation.

Table 20.76 TMTRSTSp Bit Allocation

Bit Position	Channel	Transmit Buffer No.
0	0	0
1	0	1
.	.	.
.	.	.
15	0	15
16	1	0
.	.	.
.	.	.
30	1	14
31	1	15
32	2	0
33	2	1
.	.	.
.	.	.
47	2	15
48	3	0
.	.	.
.	.	.
62	3	14
63	3	15
64	4	0
65	4	1
.	.	.
.	.	.
78	4	14
79	4	15

20.3.11.2 RSCANnTMTARSTSy — Transmit Buffer Transmit Abort Request Status Register y

Note: For the value of index "y", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTARSTSp (p = y × 32 + 31)	TMTARSTSp (p = y × 32 + 30)	TMTARSTSp (p = y × 32 + 29)	TMTARSTSp (p = y × 32 + 28)	TMTARSTSp (p = y × 32 + 27)	TMTARSTSp (p = y × 32 + 26)	TMTARSTSp (p = y × 32 + 25)	TMTARSTSp (p = y × 32 + 24)	TMTARSTSp (p = y × 32 + 23)	TMTARSTSp (p = y × 32 + 22)	TMTARSTSp (p = y × 32 + 21)	TMTARSTSp (p = y × 32 + 20)	TMTARSTSp (p = y × 32 + 19)	TMTARSTSp (p = y × 32 + 18)	TMTARSTSp (p = y × 32 + 17)	TMTARSTSp (p = y × 32 + 16)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTARSTSp (p = y × 32 + 15)	TMTARSTSp (p = y × 32 + 14)	TMTARSTSp (p = y × 32 + 13)	TMTARSTSp (p = y × 32 + 12)	TMTARSTSp (p = y × 32 + 11)	TMTARSTSp (p = y × 32 + 10)	TMTARSTSp (p = y × 32 + 9)	TMTARSTSp (p = y × 32 + 8)	TMTARSTSp (p = y × 32 + 7)	TMTARSTSp (p = y × 32 + 6)	TMTARSTSp (p = y × 32 + 5)	TMTARSTSp (p = y × 32 + 4)	TMTARSTSp (p = y × 32 + 3)	TMTARSTSp (p = y × 32 + 2)	TMTARSTSp (p = y × 32 + 1)	TMTARSTSp (p = y × 32 + 0)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.77 RSCANnTMTARSTSy Register Contents

Bit Position	Bit Name	Function
31 to 0	TMTARSTSp	Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 31 to y × 32 + 0) 0: No transmit abort request is present. 1: A transmit abort request is present.

TMTARSTSp Flags (p = see Table 20.6)

These flags indicate the status of the TMTAR bit in the RSCANnTMCp register.

When the TMTAR bit is set to 1 (transmit abort is requested), the corresponding TMTARSTSp flag is set to 1.

The corresponding TMTARSTSp flag is cleared to 0 when the TMTAR bit is set to 0 (transmit abort is not requested) or in channel reset mode.

Table 20.78 shows the bit allocation.

Table 20.78 TMTARSTSp Bit Allocation

Bit Position	Channel	Transmit Buffer No.
0	0	0
1	0	1
.	.	.
.	.	.
15	0	15
16	1	0
.	.	.
.	.	.
30	1	14
31	1	15
32	2	0
33	2	1
.	.	.
.	.	.
47	2	15
48	3	0
.	.	.
.	.	.
62	3	14
63	3	15
64	4	0
65	4	1
.	.	.
.	.	.
78	4	14
79	4	15

20.3.11.3 RSCANnTMTCSy — Transmit Buffer Transmit Complete Status Register y

Note: For the value of index "y", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTCS TSp (p = y × 32 + 31)	TMTCS TSp (p = y × 32 + 30)	TMTCS TSp (p = y × 32 + 29)	TMTCS TSp (p = y × 32 + 28)	TMTCS TSp (p = y × 32 + 27)	TMTCS TSp (p = y × 32 + 26)	TMTCS TSp (p = y × 32 + 25)	TMTCS TSp (p = y × 32 + 24)	TMTCS TSp (p = y × 32 + 23)	TMTCS TSp (p = y × 32 + 22)	TMTCS TSp (p = y × 32 + 21)	TMTCS TSp (p = y × 32 + 20)	TMTCS TSp (p = y × 32 + 19)	TMTCS TSp (p = y × 32 + 18)	TMTCS TSp (p = y × 32 + 17)	TMTCS TSp (p = y × 32 + 16)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTCS TSp (p = y × 32 + 15)	TMTCS TSp (p = y × 32 + 14)	TMTCS TSp (p = y × 32 + 13)	TMTCS TSp (p = y × 32 + 12)	TMTCS TSp (p = y × 32 + 11)	TMTCS TSp (p = y × 32 + 10)	TMTCS TSp (p = y × 32 + 9)	TMTCS TSp (p = y × 32 + 8)	TMTCS TSp (p = y × 32 + 7)	TMTCS TSp (p = y × 32 + 6)	TMTCS TSp (p = y × 32 + 5)	TMTCS TSp (p = y × 32 + 4)	TMTCS TSp (p = y × 32 + 3)	TMTCS TSp (p = y × 32 + 2)	TMTCS TSp (p = y × 32 + 1)	TMTCS TSp (p = y × 32 + 0)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.79 RSCANnTMTCSy Register Contents

Bit Position	Bit Name	Function
31 to 0	TMTCSySp	Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 31 to y × 32 + 0) 0: Transmission has not been completed. 1: Transmission has been completed.

TMTCSySp Flags (p = see Table 20.6)

When the TMTRF[1:0] flag in the RSCANnTMSTSp register is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)), the corresponding TMTCSySp flag is set to 1.

These flags are cleared to 0 when the TMTRF[1:0] flag is set to 00_B or in channel reset mode.

Table 20.80 shows the bit allocation.

Table 20.80 TMTcSTSp Bit Allocation

Bit Position	Channel	Transmit Buffer No.
0	0	0
1	0	1
.	.	.
.	.	.
15	0	15
16	1	0
.	.	.
.	.	.
30	1	14
31	1	15
32	2	0
33	2	1
.	.	.
.	.	.
47	2	15
48	3	0
.	.	.
.	.	.
62	3	14
63	3	15
64	4	0
65	4	1
.	.	.
.	.	.
78	4	14
79	4	15

20.3.11.4 RSCANnTMTASTSy — Transmit Buffer Transmit Abort Status Register y

Note: For the value of index "y", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTAS TSp (p = y × 32 + 31)	TMTAS TSp (p = y × 32 + 30)	TMTAS TSp (p = y × 32 + 29)	TMTAS TSp (p = y × 32 + 28)	TMTAS TSp (p = y × 32 + 27)	TMTAS TSp (p = y × 32 + 26)	TMTAS TSp (p = y × 32 + 25)	TMTAS TSp (p = y × 32 + 24)	TMTAS TSp (p = y × 32 + 23)	TMTAS TSp (p = y × 32 + 22)	TMTAS TSp (p = y × 32 + 21)	TMTAS TSp (p = y × 32 + 20)	TMTAS TSp (p = y × 32 + 19)	TMTAS TSp (p = y × 32 + 18)	TMTAS TSp (p = y × 32 + 17)	TMTAS TSp (p = y × 32 + 16)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTAS TSp (p = y × 32 + 15)	TMTAS TSp (p = y × 32 + 14)	TMTAS TSp (p = y × 32 + 13)	TMTAS TSp (p = y × 32 + 12)	TMTAS TSp (p = y × 32 + 11)	TMTAS TSp (p = y × 32 + 10)	TMTAS TSp (p = y × 32 + 9)	TMTAS TSp (p = y × 32 + 8)	TMTAS TSp (p = y × 32 + 7)	TMTAS TSp (p = y × 32 + 6)	TMTAS TSp (p = y × 32 + 5)	TMTAS TSp (p = y × 32 + 4)	TMTAS TSp (p = y × 32 + 3)	TMTAS TSp (p = y × 32 + 2)	TMTAS TSp (p = y × 32 + 1)	TMTAS TSp (p = y × 32 + 0)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.81 RSCANnTMTASTSy Register Contents

Bit Position	Bit Name	Function
31 to 0	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 31 to y × 32 + 0) 0: Transmission is not aborted. 1: Transmission is aborted.

TMTASTSp Flags (p = see Table 20.6)

When the TMTRF[1:0] flag in the RSCANnTMSTSp register is set to 01_B (transmit abort has been completed), the corresponding TMTASTSp flag is set to 1.

These flags are cleared to 0 when the TMTRF[1:0] flag is set to 00_B or in channel reset mode.

Table 20.82 shows the bit allocation.

Table 20.82 TMTASTSp Bit Allocation

Bit Position	Channel	Transmit Buffer No.
0	0	0
1	0	1
.	.	.
.	.	.
15	0	15
16	1	0
.	.	.
.	.	.
30	1	14
31	1	15
32	2	0
33	2	1
.	.	.
.	.	.
47	2	15
48	3	0
.	.	.
.	.	.
62	3	14
63	3	15
64	4	0
65	4	1
.	.	.
.	.	.
78	4	14
79	4	15

20.3.12 Details of Transmit Queue-Related Registers

20.3.12.1 RSCANnTXQCCm — Transmit Queue Configuration and Control Register m

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TXQIM	TXQIE	TXQDC[3:0]			—	—	—	—	—	—	—	—	TXQE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 20.83 RSCANnTXQCCm Register Contents

Bit Position	Bit Name	Function
31 to 14	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
13	TXQIM	Transmit Queue Interrupt Source Select bit 0: When the buffer becomes empty upon completion of message transmission, a transmit queue interrupt source occurs. 1: A transmit queue interrupt source occurs each time a message has been transmitted.
12	TXQIE	Transmit Queue Interrupt Enable 0: Transmit queue interrupt is disabled. 1: Transmit queue interrupt is enabled.
11 to 8	TXQDC [3:0]	Transmit Queue Depth Configuration Setting these bits to g (g = 2 to 15) makes the "g + 1" transmit queue available. Setting these bits to 0 disables the transmit queue. Setting these bits to 1 is prohibited.
7 to 1	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	TXQE	Transmit Queue Enable 0: The transmit queue is not used. 1: The transmit queue is used.

TXQIM Bit

This bit is used to select a transmit queue interrupt source. Modify this bit in channel reset mode.

TXQIE Bit

When the TXQIE bit is set to 1 and the source selected by the TXQIM bit occurs, an interrupt request is generated.

Set the TXQE bit to 0 when modifying the TXQIE bit.

TXQDC[3:0] Bits

These bits are used to set the number of transmit buffers to be allocated to the transmit queue. Transmit buffers are allocated to the transmit queues in the descending order of buffer numbers from $(m \times 16 + 15)$ to $(m \times 16 + 0)$. see

Table 20.20. For an example of buffer allocation, see **Figure 20.9**. Modify these bits only in the channel reset mode.

TXQE Bit

Setting this bit to 1 enables the transmit queue. Modify this bit in channel communication mode or channel halt mode. This bit is cleared to 0 in channel reset mode.

Set the TXQDC[3:0] bits to 0010_B or more before setting the TXQE bit to 1.

20.3.12.2 RSCANnTXQSTSm — Transmit Queue Status Register m

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQIF	TXQFL L	TXQEM P
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the status. Any other writing to the bit results in retention of the status and does not change the value.

Table 20.84 RSCANnTXQSTSm Register Contents

Bit Position	Bit Name	Function
31 to 13	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
12 to 8	—	Reserved When read, the undefined value is returned. When writing to these bits, write the value after reset.
7 to 3	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	TXQIF	Transmit Queue Interrupt Request Flag 0: No transmit queue interrupt request is present. 1: A transmit queue interrupt request is present.
1	TXQFLL	Transmit Queue Full Status Flag 0: The transmit queue is not full. 1: The transmit queue is full.
0	TXQEMP	Transmit Queue Empty Status Flag 0: The transmit queue contains messages. 1: The transmit queue contains no message (transmit queue empty).

TXQIF Flag

The TXQIF flag is set to 1 when the interrupt source set by the TXQIM bit in the RSCANnTXQCCm register has occurred.

The TXQIF flag is cleared to 0 in channel reset mode or by writing 0 to this flag. This flag is not cleared to 0 by setting the TXQE bit in the RSCANnTXQCCm register to 0 (the transmit queue is not used).

TXQFLL Flag

The TXQFLL flag is set to 1 when the number of messages stored in the transmit queue matches the transmit queue depth set by the TXQDC[3:0] bits in the RSCANnTXQCCm register.

This flag is cleared to 0 in any of the following cases.

- The number of messages set for the transmit queue is smaller than the transmit queue depth set by the TXQDC[3:0] bits.

- In channel reset mode

TXQEMP Flag

The TXQEMP flag is cleared to 0 when even a single message is placed in the transmit queue.

This flag is set to 1 in any of the following cases.

- The TXQE bit is set to 0 (the transmit queue is not used).
- The transmit queue becomes empty.
- In channel reset mode

20.3.12.3 RSCANnTXQPCTRM — Transmit Queue Pointer Control Register m

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 20.85 RSCANnTXQPCTRM Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
7 to 0	TXQPC [7:0]	Transmit Queue Pointer Control Writing FF _H to these bits move the write pointer of the transmit queue to the next queue.

TXQPC[7:0] Bits

Writing FF_H to the TXQPC[7:0] bits move the write pointer to the next transmit queue and generates a transmit request for the message. Write transmit messages to the RSCANnTMID_p, RSCANnTMPTR_p, RSCANnTMDF0_p, RSCANnTMDF1_p registers (p = 15, 31, 47, 63, 79) before writing FF_H to the TXQPC[7:0] bits.

When writing FF_H to these bits, make sure that the TXQE bit in the RSCANnTXQCC_m register is set to 1 (the transmit queue is used) and the TXQFLL flag in the RSCANnTXQSTS_m register is 0 (the transmit queue is not full).

20.3.13 Details of Transmit History-Related Registers

20.3.13.1 RSCANnTHLCCm — Transmit History List Configuration and Control Register m

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	THLDT E	THLIM	THLIE	—	—	—	—	—	—	—	THLE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 20.86 RSCANnTHLCCm Register Contents

Bit Position	Bit Name	Function
31 to 11	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
10	THLDTE	Transmit History Target Buffer Select 0: Entries from transmit/receive FIFO buffers and transmit queue 1: Entries from transmit buffers, transmit/receive FIFO buffers, and transmit queue
9	THLIM	Transmit History Interrupt Source Select 0: When 12 items of data have been stored in the transmit history buffer 1: Each time transmit history data is stored in the transmit history buffer
8	THLIE	Transmit History Interrupt Enable 0: Transmit history interrupt is disabled. 1: Transmit history interrupt is enabled.
7 to 1	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	THLE	Transmit History Buffer Enable 0: Transmit history buffer is not used. 1: Transmit history buffer is used.

THLDTE Bit

When this bit is set to 0, the transmit history data of messages transmitted from transmit/receive FIFO buffers and the transmit queue is stored in the transmit history buffer. When this bit is set to 1, the transmit history data of messages transmitted from transmit buffers, transmit/receive FIFO buffers, and the transmit queue is stored in the transmit history buffer.

Modify this bit only in channel reset mode.

THLIM Bit

This bit is used to select a transmit history interrupt source.

Modify this bit only in channel reset mode.

THLIE Bit

When the THLIE bit is set to 1 and the source selected by the THLIM bit has occurred, a transmit history interrupt request is generated. Modify the THLIE bit with the THLE bit is set to 0.

THLE Bit

Setting this bit to 1 enables the transmit history buffer. When data transmission from the buffer selected by the THLDTE bit has been completed, the transmit history data of the transmitted messages is stored in the transmit history buffer.

Modify this bit in channel communication mode or channel halt mode.

This bit is cleared to 0 in channel reset mode.

20.3.13.2 RSCANnTHLSTSm — Transmit History List Status Register m

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THLMC[4:0]				—	—	—	—	THLIF	THLELT	THLFLL	THLEMP	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the status. Any other writing to the bit results in retention of the status and does not change the value.

Table 20.87 RSCANnTHLSTSm Register Contents

Bit Position	Bit Name	Function
31 to 13	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
12 to 8	THLMC [4:0]	Transmit History Buffer Unread Data Counter These bits indicate the number of unread data stored in the transmit history buffer.
7 to 4	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	THLIF	Transmit History Interrupt Request Flag 0: No transmit history interrupt request is present. 1: A transmit history interrupt request is present.
2	THLELT	Transmit History Buffer Overflow Flag 0: Transmit history buffer overflow has not occurred. 1: Transmit history buffer overflow has occurred.
1	THLFLL	Transmit History Buffer Full Status Flag 0: Transmit history buffer is not full. 1: Transmit history buffer is full.
0	THLEMP	Transmit History Buffer Empty Status Flag 0: Transmit history buffer contains unread data. 1: Transmit history buffer contains no unread data (buffer empty).

THLMC[4:0] Bits

These bits indicate the number of unread data stored in the transmit history buffer.

These bits are cleared to 0 in channel reset mode.

THLIF Flag

The THLIF flag is set to 1 when the interrupt source set by the THLIM bit in the RSCANnTHLCCm register has occurred.

This flag is cleared to 0 in channel reset mode or by writing 0 to this flag by the program.

To clear the flag of the register to 0, the program must write 0 to the given flag. When writing 0, use a store instruction to write 0 to the given flag and 1 to the other flags.

THLELT Flag

The THLELT flag is set to 1 when it is attempted to store new transmit history data while the transmit history buffer is full. In this case, the new data is discarded. This flag becomes 0 in channel reset mode or by writing 0 to this flag.

To clear the flag of the register to 0, the program must write 0 to the given flag. When writing 0, use a store instruction to write 0 to the given flag and 1 to the other flags.

THLFLL Flag

The THLFLL flag is set to 1 when 16 items of data have been stored in the transmit history buffer, and is cleared to 0 when the number of data stored in the transmit history buffer has decreased to less than 16. This bit is also cleared to 0 in channel reset mode or when the THLE bit in the RSCANnTHLCCm register is set to 0 (transmit history buffer is not used).

THLEMP Flag

The THLEMP flag is cleared to 0 when even a single item of transmit history data has been stored in the transmit history buffer.

This flag is set to 1 when all the data in the transmit history buffer has been read. This flag is also set to 1 in channel reset mode or when the THLE bit in the RSCANnTHLCCm register is set to 0 (transmit history buffer is not used).

NOTE

To clear the THLIF and THLELT flags to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using a store instruction, set 1 to other flags.

20.3.13.3 RSCANnTHLPCTRm — Transmit History List Pointer Control Register m

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	THLPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 20.88 RSCANnTHLPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When writing to these bits, write the value after reset.
7 to 0	THLPC [7:0]	Transmit History List Pointer Control Writing FF _H to these bits move the read pointer to the next unread data in the transmit history buffer.

THLPC[7:0] Bits

When the THLPC[7:0] bits are set to FF_H, the read pointer moves to the next data in the transmit history buffer. At this time, the THLMC[4:0] (transmit history buffer unread data counter) value in the RSCANnTHLSTSm register is decremented by 1. After reading the RSCANnTHLACCm register, write FF_H to the THLPC[7:0] bits.

When writing FF_H to these bits, make sure that the THLE bit in the RSCANnTHLCCm register is set to 1 (transmit history buffer is used) and the THLEMP flag in the RSCANnTHLSTSm register is 0.

20.3.13.4 RSCANnTHLACCm — Transmit History List Access Register m

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TID[7:0]							—	BN[3:0]			BT[2:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.89 RSCANnTHLACCm Register Contents

Bit Position	Bit Name	Function																
31 to 16	TMTS[15:0]	Timestamp Data The timestamp data of stored data can be read.																
15 to 8	TID[7:0]	Label Data The label information of stored data can be read.																
7	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.																
6 to 3	BN[3:0]	Buffer Number Data The buffer number of transmit source (transmit buffer, transmit/receive FIFO or transmit queue) can be read.																
2 to 0	BT[2:0]	Buffer Type Data <table border="0" style="margin-left: 20px;"> <tr> <td>b2</td><td>b1</td><td>b0</td><td></td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>: Transmit buffer</td> </tr> <tr> <td>0</td><td>1</td><td>0</td><td>: Transmit/receive FIFO buffer</td> </tr> <tr> <td>1</td><td>0</td><td>0</td><td>: Transmit queue</td> </tr> </table>	b2	b1	b0		0	0	1	: Transmit buffer	0	1	0	: Transmit/receive FIFO buffer	1	0	0	: Transmit queue
b2	b1	b0																
0	0	1	: Transmit buffer															
0	1	0	: Transmit/receive FIFO buffer															
1	0	0	: Transmit queue															

TMTS[15:0] Bits

When the TMTSCE bit in the RSCANnGCFG register is 1, timestamp values in transmit history data stored in the transmit history buffer is displayed. When the TMTSCE bit is 0, the TMTS[15:0] bits are always read as 0.

TID[7:0] Bits

These bits indicate the label information of transmit history data stored in the transmit history buffer.

BN[3:0] Bits

These bits indicate the transmit source buffer number of transmit history data stored in the transmit history buffer.

BT[2:0] Bits

These bits indicate the transmit source buffer type of transmit history data stored in the transmit history buffer.

20.3.14 Details of Test-Related Registers

20.3.14.1 RSCANnGTSTCFG — Global Test Configuration Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	RTMPS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	C4ICBCE	C3ICBCE	C2ICBCE	C1ICBCE	C0ICBCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 20.90 RSCANnGTSTCFG Register Contents

Bit Position	Bit Name	Function
31 to 23	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
22 to 16	RTMPS [6:0]	RAM Test Page Configuration Set a value within a range of page 0 (00 _H) to page 49 (31 _H).
15 to 5	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
4	C4ICBCE	CAN4 Inter-Channel Communication Test Enable 0: CAN4 inter-channel communication test is disabled. 1: CAN4 inter-channel communication test is enabled.
3	C3ICBCE	CAN3 Inter-Channel Communication Test Enable 0: CAN3 inter-channel communication test is disabled. 1: CAN3 inter-channel communication test is enabled.
2	C2ICBCE	CAN2 Inter-Channel Communication Test Enable 0: CAN2 inter-channel communication test is disabled. 1: CAN2 inter-channel communication test is enabled.
1	C1ICBCE	CAN1 Inter-Channel Communication Test Enable 0: CAN1 inter-channel communication test is disabled. 1: CAN1 inter-channel communication test is enabled.
0	C0ICBCE	CAN0 Inter-Channel Communication Test Enable 0: CAN0 inter-channel communication test is disabled. 1: CAN0 inter-channel communication test is enabled.

Modify the RSCANnGTSTCFG register in global test mode.

RTMPS[6:0] Bits

These bits are used to set the RAM test target page number for RAM test. Set a value from 00_H to 31_H.

C4ICBCE Bit

Setting this bit to 1 enables channel 4 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C3ICBCE Bit

Setting this bit to 1 enables channel 3 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C2ICBCE Bit

Setting this bit to 1 enables channel 2 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C1ICBCE Bit

Setting this bit to 1 enables channel 1 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C0ICBCE Bit

Setting this bit to 1 enables channel 0 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

20.3.14.2 RSCANnGTSTCTR — Global Test Control Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	ICBCTME
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W

Table 20.91 RSCANnGTSTCTR Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	RTME	RAM Test Enable 0: RAM test is disabled. 1: RAM test is enabled.
1	—	Reserved When read, the value after reset is returned. When writing to this bit, write the value after reset.
0	ICBCTME	Inter-Channel Communication Test Enable 0: Inter-channel communication test is disabled. 1: Inter-channel communication test is enabled.

RTME Bit

Setting this bit to 1 enables RAM test. Modify this bit only in global test mode.

This bit is cleared to 0 in global reset mode.

1. Set the GMDC[1:0] bits in the RSCANnGCTR register to 10_B (global test mode).
2. Set the RTME bit to 1.
3. Check that the RTME bit is set to 1.

For the setting procedure of RAM test, see **Figure 20.37**.

ICBCTME Bit

When this bit is set to 1, a communication test is enabled between the channels for which the CmICBCE bit (m = see **Table 20.6**) in the RSCANnGTSTCFG register has been set to 1. Modify the ICBCTME bit only in global test mode.

This bit is cleared to 0 in global reset mode.

20.3.14.3 RSCANnGLOCKK — Global Lock Key Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOCK[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1

Note 1. Writing to these bits is effective only when the RS-CANFD module is in global test mode.

Table 20.92 RSCANnGLOCKK Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
15 to 0	LOCK[15:0]	Lock Key These bits are key bits to release protection of test mode.

The RSCANnGLOCKK register is a write-only register to release protection of special test bits. For protection released data, **Section 20.11.4.2, Procedure for Releasing the Protection**.

LOCK[15:0] Bits

Writing the protection release data to the LOCK[15:0] bits in succession enables writing 1 to the RTME bit in the GTSTCTR register.

After the protection has been released, writing to the I/O register area of CAN (<RSCFDn_base> + 0000_H to <RSCFDn_base> + 04FF_H), excluding the RAM, enables the protection again.

Reading from the register of the CAN I/O register area or reading from/writing to other areas does not enable the protection.

20.3.14.4 RSCANnRPGACCr — RAM Test Page Access Register r

Note: For the value of index "r", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.93 RSCANnRPGACCr Register Contents

Bit Position	Bit Name	Function
31 to 0	RDTA [31:0]	RAM Data Test Access Data can be read and written in CAN RAM.

Modify the RSCANnRPGACCr register in global test mode with the RTME bit in the RSCANnGTSTCTR register set to 1 (RAM test is enabled).

The RSCANnRPGACCr register can be read and written when the RTME bit is set to 1.

20.4 Registers (CAN FD Mode)

This section describes all registers to be used when the RS-CANFD is used in CAN FD mode.

20.4.1 List of Registers

The following tables list RS-CANFD registers to be used in CAN FD mode.

For details about <RSCFDn_base>, see **Section 20.1.2, Register Base Address**.

For details about registers initialized in Global reset mode or Channel reset mode, see following.

- **Table 20.186, Registers Initialized in Global Reset Mode or Channel Reset Mode**
- **Table 20.187, Registers Initialized Only in Global Reset Mode**

Table 20.94 List of Registers (1/4)

Module Name	Register Name	Symbol	Address	Access	Protection
Interface Mode-Related Registers					
RSCFDn	Global Interface Mode Select Register	RSCFDnCFDGRMCFG	<RSCFDn_base> + 04FC _H	8,16,32	-
Channel-Related Registers					
RSCFDn	Channel m Nominal Bit Rate Configuration Register	RSCFDnCFDCmNCFG	<RSCFDn_base> + 0000 _H + (10 _H × m)	8,16,32	-
RSCFDn	Channel m Control Register	RSCFDnCFDCmCTR	<RSCFDn_base> + 0004 _H + (10 _H × m)	8,16,32	-
RSCFDn	Channel m Status Register	RSCFDnCFDCmSTS	<RSCFDn_base> + 0008 _H + (10 _H × m)	8,16,32	-
RSCFDn	Channel m Error Flag Register	RSCFDnCFDCmERFL	<RSCFDn_base> + 000C _H + (10 _H × m)	8,16,32	-
RSCFDn	Channel m Data Bit Rate Configuration Register	RSCFDnCFDCmDCFG	<RSCFDn_base> + 0500 _H + (20 _H × m)	8,16,32	-
RSCFDn	Channel m CAN FD Configuration Register	RSCFDnCFDCmFDCFG	<RSCFDn_base> + 0504 _H + (20 _H × m)	8,16,32	-
RSCFDn	Channel m CAN FD Control Register	RSCFDnCFDCmFDCTR	<RSCFDn_base> + 0508 _H + (20 _H × m)	8,16,32	-
RSCFDn	Channel m CAN FD Status Register	RSCFDnCFDCmFDSTS	<RSCFDn_base> + 050C _H + (20 _H × m)	8,16,32	-
RSCFDn	Channel m CAN FD CRC Register	RSCFDnCFDCmFDCRC	<RSCFDn_base> + 0510 _H + (20 _H × m)	8,16,32	-
Global-Related Registers					
RSCFDn	Global Configuration Register	RSCFDnCFDGCFG	<RSCFDn_base> + 0084 _H	8,16,32	-
RSCFDn	Global Control Register	RSCFDnCFDGCTR	<RSCFDn_base> + 0088 _H	8,16,32	-
RSCFDn	Global Status Register	RSCFDnCFDGSTS	<RSCFDn_base> + 008C _H	8,16,32	-
RSCFDn	Global Error Flag Register	RSCFDnCFDGERFL	<RSCFDn_base> + 0090 _H	8,16,32	-
RSCFDn	Global Timestamp Counter Register	RSCFDnCFDGTSC	<RSCFDn_base> + 0094 _H	16,32	-
RSCFDn	Global TX Interrupt Status Register 0	RSCFDnCFDGTINTSTS0	<RSCFDn_base> + 0460 _H	8,16,32	-
RSCFDn	Global TX Interrupt Status Register 1	RSCFDnCFDGTINTSTS1	<RSCFDn_base> + 0464 _H	8,16,32	-
RSCFDn	Global FD Configuration Register	RSCFDnCFDGFDCFG	<RSCFDn_base> + 0474 _H	8,16,32	-

Table 20.94 List of Registers (2/4)

Module Name	Register Name	Symbol	Address	Access	Access Protection
Receive Rule-Related Registers					
RSCFDn	Receive Rule Entry Control Register	RSCFDnCFDGAFLECTR	<RSCFDn_base> + 0098 _H	8,16,32	-
RSCFDn	Receive Rule Configuration Register 0	RSCFDnCFDGAFLCFG0	<RSCFDn_base> + 009C _H	8,16,32	-
RSCFDn	Receive Rule Configuration Register 1	RSCFDnCFDGAFLCFG1	<RSCFDn_base> + 00A0 _H	8,16,32	-
RSCFDn	Receive Rule ID Register j	RSCFDnCFDGAFLIDj	<RSCFDn_base> + 1000 _H + (10 _H × j)	8,16,32	-
RSCFDn	Receive Rule Mask Register j	RSCFDnCFDGAFLMj	<RSCFDn_base> + 1004 _H + (10 _H × j)	8,16,32	-
RSCFDn	Receive Rule Pointer 0 Register j	RSCFDnCFDGAFLP0_j	<RSCFDn_base> + 1008 _H + (10 _H × j)	8,16,32	-
RSCFDn	Receive Rule Pointer 1 Register j	RSCFDnCFDGAFLP1_j	<RSCFDn_base> + 100C _H + (10 _H × j)	8,16,32	-
Receive Buffer-Related Registers					
RSCFDn	Receive Buffer Number Register	RSCFDnCFDRMNB	<RSCFDn_base> + 00A4 _H	8,16,32	-
RSCFDn	Receive Buffer New Data Register y	RSCFDnCFDRMNDy	<RSCFDn_base> + 00A8 _H + (04 _H × y)	8,16,32	-
RSCFDn	Receive Buffer ID Register q	RSCFDnCFDRMIDq	<RSCFDn_base> + 2000 _H + (20 _H × q)	8,16,32	-
RSCFDn	Receive Buffer Pointer Register q	RSCFDnCFDRMPTRq	<RSCFDn_base> + 2004 _H + (20 _H × q)	8,16,32	-
RSCFDn	Receive Buffer CAN FD Status Register q	RSCFDnCFDRMFDSTSq	<RSCFDn_base> + 2008 _H + (20 _H × q)	8,16,32	-
RSCFDn	Receive Buffer Data Field b Register q	RSCFDnCFDRMDFb_q	<RSCFDn_base> + 200C _H + (04 _H × b) + (20 _H × q)	8,16,32	-
Receive FIFO Buffer-Related Registers					
RSCFDn	Receive FIFO Buffer Configuration and Control Register x	RSCFDnCFDRFCCx	<RSCFDn_base> + 00B8 _H + (04 _H × x)	8,16,32	-
RSCFDn	Receive FIFO Buffer Status Register x	RSCFDnCFDRFSTSx	<RSCFDn_base> + 00D8 _H + (04 _H × x)	8,16,32	-
RSCFDn	Receive FIFO Buffer Pointer Control Register x	RSCFDnCFDRFPCTRx	<RSCFDn_base> + 00F8 _H + (04 _H × x)	8,16,32	-
RSCFDn	Receive FIFO Buffer Access ID Register x	RSCFDnCFDRFIDx	<RSCFDn_base> + 3000 _H + (80 _H × x)	8,16,32	-
RSCFDn	Receive FIFO Buffer Access Pointer Register x	RSCFDnCFDRFPTRx	<RSCFDn_base> + 3004 _H + (80 _H × x)	8,16,32	-
RSCFDn	Receive FIFO CAN FD Status Register x	RSCFDnCFDRFFDSTSx	<RSCFDn_base> + 3008 _H + (80 _H × x)	8,16,32	-
RSCFDn	Receive FIFO Buffer Access Data Field d Register x	RSCFDnCFDRFDFd_x	<RSCFDn_base> + 300C _H + (04 _H × d) + (80 _H × x)	8,16,32	-

Table 20.94 List of Registers (3/4)

Module Name	Register Name	Symbol	Address	Access	Protection
Transmit/Receive FIFO Buffer Related Registers					
RSCFDn	Transmit/Receive FIFO Buffer Configuration and Control Register k	RSCFDnCFDCFCCK	<RSCFDn_base> + 0118 _H + (04 _H × k)	8,16,32	-
RSCFDn	Transmit/Receive FIFO Buffer Status Register k	RSCFDnCFDCFSTSk	<RSCFDn_base> + 0178 _H + (04 _H × k)	8,16,32	-
RSCFDn	Transmit/Receive FIFO Buffer Pointer Control Register k	RSCFDnCFDCFPCTRk	<RSCFDn_base> + 01D8 _H + (04 _H × k)	8,16,32	-
RSCFDn	Transmit/Receive FIFO Buffer Access ID Register k	RSCFDnCFDCFIDk	<RSCFDn_base> + 3400 _H + (80 _H × k)	8,16,32	-
RSCFDn	Transmit/Receive FIFO Buffer Access Pointer Register k	RSCFDnCFDCFPTRk	<RSCFDn_base> + 3404 _H + (80 _H × k)	8,16,32	-
RSCFDn	Transmit/Receive FIFO CAN FD Configuration/Status Register k	RSCFDnCFDCFFDCSTSk	<RSCFDn_base> + 3408 _H + (80 _H × k)	8,16,32	-
RSCFDn	Transmit/Receive FIFO Buffer Access Data Field d Register k	RSCFDnCFDCFDf_d_k	<RSCFDn_base> + 340C _H + (04 _H × d) + (80 _H × k)	8,16,32	-
FIFO Status-Related Registers					
RSCFDn	FIFO Empty Status Register	RSCFDnCFDFESTS	<RSCFDn_base> + 0238 _H	8,16,32	-
RSCFDn	FIFO Full Status Register	RSCFDnCFDFFSTS	<RSCFDn_base> + 023C _H	8,16,32	-
RSCFDn	FIFO Message Lost Status Register	RSCFDnCFDFMSTS	<RSCFDn_base> + 0240 _H	8,16,32	-
RSCFDn	Receive FIFO Buffer Interrupt Flag Status Register	RSCFDnCFDRFISTS	<RSCFDn_base> + 0244 _H	8,16,32	-
RSCFDn	Transmit/Receive FIFO Buffer Receive Interrupt Flag Status Register	RSCFDnCFDCFRISTS	<RSCFDn_base> + 0248 _H	8,16,32	-
RSCFDn	Transmit/Receive FIFO Buffer Transmit Interrupt Flag Status Register	RSCFDnCFDCFTISTS	<RSCFDn_base> + 024C _H	8,16,32	-
FIFO DMA-Related Registers					
RSCFDn	DMA Enable Register	RSCFDnCFDCDTCT	<RSCFDn_base> + 0490 _H	8,16,32	-
RSCFDn	DMA Status Register	RSCFDnCFDCDTSTS	<RSCFDn_base> + 0494 _H	8,16,32	-
Transmit Buffer-Related Registers					
RSCFDn	Transmit Buffer Control Register p	RSCFDnCFDTMCP	<RSCFDn_base> + 0250 _H + (01 _H × p)	8	-
RSCFDn	Transmit Buffer Status Register p	RSCFDnCFDTMSTSp	<RSCFDn_base> + 02D0 _H + (01 _H × p)	8	-
RSCFDn	Transmit Buffer ID Register p	RSCFDnCFDTMIDp	<RSCFDn_base> + 4000 _H + (20 _H × p)	8,16,32	-
RSCFDn	Transmit Buffer Pointer Register p	RSCFDnCFDTMPTRp	<RSCFDn_base> + 4004 _H + (20 _H × p)	8,16,32	-
RSCFDn	Transmit Buffer CAN FD Configuration Register p	RSCFDnCFDTMFDCTRp	<RSCFDn_base> + 4008 _H + (20 _H × p)	8,16,32	-
RSCFDn	Transmit Buffer Data Field b Register p	RSCFDnCFDTMDFb_p	<RSCFDn_base> + 400C _H + (04 _H × b) + (20 _H × p)	8,16,32	-
RSCFDn	Transmit Buffer Interrupt Enable Configuration Register y	RSCFDnCFDTMIECy	<RSCFDn_base> + 0390 _H + (04 _H × y)	8,16,32	-

Table 20.94 List of Registers (4/4)

Module Name	Register Name	Symbol	Address	Access	Access Protection
Transmit Buffer Status-Related Registers					
RSCFDn	Transmit Buffer Transmit Request Status Register y	RSCFDnCFDTMTRSTSy	<RSCFDn_base> + 0350 _H + (04 _H × y)	8,16,32	-
RSCFDn	Transmit Buffer Transmit Abort Request Status Register y	RSCFDnCFDTMTARSTSy	<RSCFDn_base> + 0360 _H + (04 _H × y)	8,16,32	-
RSCFDn	Transmit Buffer Transmit Complete Status Register y	RSCFDnCFDTMTCSTSy	<RSCFDn_base> + 0370 _H + (04 _H × y)	8,16,32	-
RSCFDn	Transmit Buffer Transmit Abort Status Register y	RSCFDnCFDTMTASTSy	<RSCFDn_base> + 0380 _H + (04 _H × y)	8,16,32	-
Transmit Queue-Related Registers					
RSCFDn	Transmit Queue Configuration and Control Register m	RSCFDnCFDTXQCCm	<RSCFDn_base> + 03A0 _H + (04 _H × m)	8,16,32	-
RSCFDn	Transmit Queue Status Register m	RSCFDnCFDTXQSTSm	<RSCFDn_base> + 03C0 _H + (04 _H × m)	8,16,32	-
RSCFDn	Transmit Queue Pointer Control Register m	RSCFDnCFDTXQPCTRm	<RSCFDn_base> + 03E0 _H + (04 _H × m)	8,16,32	-
Transmit History-Related Registers					
RSCFDn	Transmit History List Configuration and Control Register m	RSCFDnCFDTHLCCm	<RSCFDn_base> + 0400 _H + (04 _H × m)	8,16,32	-
RSCFDn	Transmit History List Status Register m	RSCFDnCFDTHLSTSm	<RSCFDn_base> + 0420 _H + (04 _H × m)	8,16,32	-
RSCFDn	Transmit History List Pointer Control Register m	RSCFDnCFDTHLPCTRm	<RSCFDn_base> + 0440 _H + (04 _H × m)	8,16,32	-
RSCFDn	Transmit History List Access Register m	RSCFDnCFDTHLACCm	<RSCFDn_base> + 6000 _H + (04 _H × m)	8,16,32	-
Test-Related Registers					
RSCFDn	Global Test Configuration Register	RSCFDnCFDGTSTCFG	<RSCFDn_base> + 0468 _H	8,16,32	-
RSCFDn	Global Test Control Register	RSCFDnCFDGTSTCTR	<RSCFDn_base> + 046C _H	8,16,32	RSCFDn CFDGLO CKK
RSCFDn	Global Lock Key Register	RSCFDnCFDGLOCKK	<RSCFDn_base> + 047C _H	16,32	-
RSCFDn	RAM Test Page Access Register r	RSCFDnCFDRPGACCr	<RSCFDn_base> + 6400 _H + (04 _H × r)	8,16,32	RSCFDn CFDGTS TCTR

Table 20.95 Transmit Buffer p Allocated to Each Channel

	CANm
Transmit buffer p	Transmit buffer $16 \times m + 0$
	Transmit buffer $16 \times m + 1$
	Transmit buffer $16 \times m + 2$
	Transmit buffer $16 \times m + 3$
	Transmit buffer $16 \times m + 4$
	Transmit buffer $16 \times m + 5$
	Transmit buffer $16 \times m + 6$
	Transmit buffer $16 \times m + 7$
	Transmit buffer $16 \times m + 8$
	Transmit buffer $16 \times m + 9$
	Transmit buffer $16 \times m + 10$
	Transmit buffer $16 \times m + 11$
	Transmit buffer $16 \times m + 12$
	Transmit buffer $16 \times m + 13$
	Transmit buffer $16 \times m + 14$
	Transmit buffer $16 \times m + 15$

Table 20.96 Transmit/Receive FIFO Buffer k Allocated to Each Channel

	CANm
Transmit/receive FIFO buffer k	Transmit/receive FIFO buffer $3 \times m + 0$
	Transmit/receive FIFO buffer $3 \times m + 1$
	Transmit/receive FIFO buffer $3 \times m + 2$

Table 20.97 Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer by the Setting of Bits CFTML[3:0]

Setting of Bits CFTML[3:0]	Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer
0000 _B	Transmit buffer $16 \times m + 0$
0001 _B	Transmit buffer $16 \times m + 1$
0010 _B	Transmit buffer $16 \times m + 2$
0011 _B	Transmit buffer $16 \times m + 3$
0100 _B	Transmit buffer $16 \times m + 4$
0101 _B	Transmit buffer $16 \times m + 5$
0110 _B	Transmit buffer $16 \times m + 6$
0111 _B	Transmit buffer $16 \times m + 7$
1000 _B	Transmit buffer $16 \times m + 8$
1001 _B	Transmit buffer $16 \times m + 9$
1010 _B	Transmit buffer $16 \times m + 10$
1011 _B	Transmit buffer $16 \times m + 11$
1100 _B	Transmit buffer $16 \times m + 12$
1101 _B	Transmit buffer $16 \times m + 13$
1110 _B	Transmit buffer $16 \times m + 14$
1111 _B	Transmit buffer $16 \times m + 15$

Table 20.98 Transmit Buffer p Allocated to the Transmit Queue of Each Channel

Setting of Bits TXQDC[3:0]	Transmit Buffer p Allocated to the Transmit Queue
0000 _B	Setting prohibited
0001 _B	Setting prohibited
0010 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 13$
0011 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 12$
0100 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 11$
0101 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 10$
0110 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 9$
0111 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 8$
1000 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 7$
1001 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 6$
1010 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 5$
1011 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 4$
1100 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 3$
1101 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 2$
1110 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 1$
1111 _B	Transmit buffer $16 \times m + 15$ to $16 \times m + 0$

20.4.2 Details of Interface Mode-Related Registers

20.4.2.1 RSCFDnCFDGRMCFG — Global Interface Mode Select Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCMC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 20.99 RSCFDnCFDGRMCFG Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
0	RCMC	Interface Mode Select 0: Classical CAN mode 1: CAN FD mode

Note 1. The RSCANnGRMCFG register and the RSCFDnCFDGRMCFG register are the same register. Therefore, set either one of the registers.

Modify the RSCFDnCFDGRMCFG register only in global reset mode. Before setting other RS-CANFD registers, set this register.

RCMC Bit

Setting this bit to 0 makes classical CAN mode available.

Setting this bit to 1 causes the RS-CANFD module to transition to CAN FD mode.

To switch the RS-CAN FD module from classical CAN mode to CAN FD mode, set the values after reset to all respective registers and bits allocated only to the register map in classical CAN mode, and then modify the value of RSCFDnCFDGRMCFG register.

20.4.3 Details of Channel-Related Registers

20.4.3.1 RSCFDnCFDCmNCFG — Channel m Nominal Bit Rate Configuration Register

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	NTSEG2[4:0]				—	NTSEG1[6:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NSJW[4:0]				—	NBRP[9:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.100 RSCFDnCFDCmNCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
28 to 24	NTSEG2[4:0]	Nominal Bit Rate Time Segment 2 Control b28 b27 b26 b25 b24 0 0 0 0 0: Setting is prohibited for channels for CAN communications. 0 0 0 0 1: 2 Tq : : 1 1 1 1 0: 31 Tq 1 1 1 1 1: 32 Tq
23	—	Reserved This bit is read as the value after reset. The write value should be the value after reset.
22 to 16	NTSEG1[6:0]	Nominal Bit Rate Time Segment 1 Control b22 b21 b20 b19 b18 b17 b16 0 0 0 0 0 0 0: is prohibited for channels for CAN communications. 0 0 0 0 0 0 1: is prohibited for channels for CAN communications. 0 0 0 0 0 1 0: is prohibited for channels for CAN communications. 0 0 0 0 0 1 1: 4 Tq : : 1 1 1 1 1 1 0: 127 Tq 1 1 1 1 1 1 1: 128 Tq
15 to 11	NSJW[4:0]	Nominal Bit Rate Resynchronization Jump Width Control b15 b14 b13 b12 b11 0 0 0 0 0: 1 Tq 0 0 0 0 1: 2 Tq 0 0 0 1 0: 3 Tq : : 1 1 1 1 0: 31 Tq 1 1 1 1 1: 32 Tq
10	—	Reserved This bit is read as the value after reset. The write value should be the value after reset.

Table 20.100 RSCFDnCFDCmNCFG Register Contents (2/2)

Bit Position	Bit Name	Function
9 to 0	NBRP[9:0]	Nominal Bit Rate Prescaler Division Ratio Setting When the set value = P (0 to 1023), the nominal bit rate prescaler divides fCAN by (P + 1).

Modify the RSCFDnCFDCmNCFG register in channel reset mode or channel halt mode. Set this register in channel reset mode, and then transition to channel communication mode or channel halt mode. For the description and settings for bit timing parameters, see **Section 20.11.1, Initial Settings**.

NTSEG2[4:0] Bits

These bits specify a Tq value for the length of phase segment 2 (PHASE_SEG2) of nominal bit rate.

Possible values are 2 to 32 Tq.

Set a value smaller than the value of the NTSEG1[6:0] bits.

NTSEG1[6:0] Bits

These bits specify the total length of propagation segment (PROP_SEG) and phase segment 1 (PHASE_SEG1) of nominal bit rate as a Tq value.

Possible values are 4 to 128 Tq.

NSJW[4:0] Bits

These bits specify the resynchronization jump width of nominal bit rate as a Tq value. A value of 1 to 32 Tq is settable. Specify a value equal to or smaller than the NTSEG2[4:0] value.

NBRP[9:0] Bits

The clock obtained by dividing the CAN clock (fCAN) by the nominal bit rate prescaler ((NBRP[9:0]) + 1) becomes CANmTq(N) clock (fCANTQ(N)m). One clock of the CANmTq(N) clock becomes one Time Quantum (Tq).

The NBRP[9:0] value and the DBRP[7:0] value should be equal and the two corresponding bit rate values are different according to the respective segment values.

20.4.3.2 RSCFDnCFDCmCTR — Channel m Control Register

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0005_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ROM	CRCT	—	—	—	CTMS[1:0]	CTME	ERRD	BOM[1:0]	—	TDCVFI E	SOCOIE	EOCOIE	TAIE		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLPR	CHMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 20.101 RSCFDnCFDCmCTR Register Contents (1/2)

Bit Position	Bit Name	Function
31	ROM	Restricted Operation Mode Enable 0: Restricted operation mode is disabled 1: Restricted operation mode is enabled.
30	CRCT	CRC Error Test Enable 0: The first bit of the reception ID field is not inverted. 1: The first bit of the reception ID field is inverted.
29 to 27	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
26, 25	CTMS[1:0]	Communication Test Mode Select b ₂₆ b ₂₅ 0 0: Standard test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback mode) 1 1: Self-test mode 1 (internal loopback mode)
24	CTME	Communication Test Mode Enable 0: Communication test mode is disabled. 1: Communication test mode is enabled.
23	ERRD	Error Display Mode Select 0: Error flags are displayed only for the first error information after bits 14 to 8 in the RSCFDnCFDCmERFL register are all cleared. 1: Error flags for all error information are displayed.
22, 21	BOM[1:0]	Bus Off Recovery Mode Select b ₂₂ b ₂₁ 0 0: ISO11898-1 compliant 0 1: Transitions to channel halt mode automatically at bus-off entry 1 0: Transitions to channel halt mode automatically at bus-off end 1 1: Transitions to channel halt mode (in bus-off state) by program request
20	—	Reserved This bit is read as the value after reset. The write value should be the value after reset.
19	TDCVFI E	Transmitter Delay Compensation Violation Interrupt Enable 0: A transmitter delay compensation violation interrupt is disabled. 1: A transmitter delay compensation violation interrupt is enabled.
18	SOCOIE	Successful Occurrence Counter Overflow Interrupt Enable 0: A successful occurrence counter overflow interrupt is disabled. 1: A successful occurrence counter overflow interrupt is enabled.

Table 20.101 RSCFDnCFDCmCTR Register Contents (2/2)

Bit Position	Bit Name	Function
17	EOCOIE	Error Occurrence Counter Overflow Interrupt Enable 0: An error occurrence counter overflow interrupt is disabled. 1: An error occurrence counter overflow interrupt is enabled.
16	TAIE	Transmit Abort Interrupt Enable 0: Transmit abort interrupt is disabled. 1: Transmit abort interrupt is enabled.
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.
14	BLIE	Bus Lock Interrupt Enable 0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled.
13	OLIE	Overload Frame Transmit Interrupt Enable 0: Overload frame transmit interrupt is disabled. 1: Overload frame transmit interrupt is enabled.
12	BORIE	Bus Off Recovery Interrupt Enable 0: Bus off recovery interrupt is disabled. 1: Bus off recovery interrupt is enabled.
11	BOEIE	Bus Off Entry Interrupt Enable 0: Bus off entry interrupt is disabled. 1: Bus off entry interrupt is enabled.
10	EPIE	Error Passive Interrupt Enable 0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled.
9	EWIE	Error Warning Interrupt Enable 0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled.
8	BEIE	Bus Error Interrupt Enable 0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled.
7 to 4	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
3	RTBO	Forcible Return from Bus-off When this bit is set to 1, forcible return from the bus off state is made. This bit is always read as 0.
2	CSLPR	Channel Stop Mode 0: Other than channel stop mode 1: Channel stop mode
1, 0	CHMDC[1:0]	Mode Select b1 b0 0 0: Channel communication mode 0 1: Channel reset mode 1 0: Channel halt mode 1 1: Setting prohibited

ROM Bit

When the ROM bit and the CTME bit in the RSCFDnCFDCmCTR register are set to 1, restricted operation mode is enabled. Use the restricted operation mode only when the CTMS[1:0] value in the RSCFDnCFDCmCTR register is 00_B (standard test mode). Modify this bit only in channel halt mode. This bit is always 0 in channel reset mode.

CRCT Bit

This bit is used to test the CRC generation circuit in the RS-CANFD module. Setting this bit to 1 inverts the first bit of the ID field when a message is received. With this inversion of bit, the CRC calculation result does not match the normal CRC value of the received frame, which can detect a CRC error (the CERR bit in the RSCFDnCFDCmERFL register is 1). When using this function, note the following.

- This function is available while the CTME bit in the RSCFDnCFDCmCTR register is 1 (communication test mode enabled).
- This function cannot communicate with other CAN nodes. Use this function for inter-channel communication test (the CmICBCE bit in the RSCFDnCFDGTSTCFG register is 1).
- Bit inversion in the ID field may cause bit stuffing rule violation. In that case, no CRC error is detected but a stuff error is detected.

Modify this bit only in channel halt mode. This bit is set to 0 in channel reset mode.

CTMS[1:0] Bits

These bits are used to select a communication test mode. Modify these bits in channel halt mode only. These bits are set to 0 in channel reset mode.

CTME Bit

Setting this bit to 1 enables communication test mode. Modify these bits in channel halt mode. This bit is set to 0 in channel reset mode.

ERRD Bit

This bit is used to control the display mode of bits 14 to 8 in the RSCFDnCFDCmERFL register.

When this bit is clear to 0, if any error is detected while the flags of bits 14 to 8 in the RSCFDnCFDCmERFL register are all 0, only the flags of the first error event are set to 1. If two or more errors occur in the first error event, all the flags of the detected errors are set to 1.

When this bit is set to 1, all the flags of errors that have occurred are set to 1 regardless of the error occurrence order.

Modify this bit only in channel reset mode or channel halt mode.

BOM[1:0] Bits

These bits are used to select the bus off recovery mode of the RS-CANFD module.

When the BOM[1:0] bits are set to 00_B, return from the bus off state to the error active state is compliant with the CAN specifications. That is, the RS-CANFD module reenters the CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. A bus off recovery interrupt request is generated at the time of return from the bus off state. Even if the CHMDC[1:0] bits are set to 10_B (channel halt mode) before recessive bits are detected 128 times, the RS-CANFD module does not transition to channel halt mode until recessive bits are detected 128 times.

When the RS-CANFD module reaches the bus off state when the BOM[1:0] bits are set to 01_B, the CHMDC[1:0] bits in the RSCFDnCFDCmCTR register (m = see **Table 20.6**) are set to 10_B and the RS-CANFD module transitions to channel halt mode. No bus off recovery interrupt request is generated and the TEC[7:0] and REC[7:0] bits in the RSCFDnCFDCmSTS register are cleared to 00_H.

When the RS-CANFD module reaches the bus off state when the BOM[1:0] bits are set to 10_B, the CHMDC[1:0] bits are set to 10_B and the RS-CANFD module transitions to channel halt mode after return from the bus off state (11 consecutive recessive bits are detected 128 times). A bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H.

When the BOM[1:0] bits are set to 11_B and the CHMDC[1:0] bits are set to 10_B while the RS-CANFD module is in the bus off state, the RS-CANFD module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H. However, if 11 consecutive recessive bits are detected 128 times and the RS-CANFD module has recovered to the error active state from the bus off state before the CHMDC[1:0] bits are set to 10_B, a bus off recovery interrupt request is generated.

If a program writes to the CHMDC[1:0] bit at the same time as the RS-CANFD module transition to channel halt mode (at bus off entry when the BOM[1:0] bits are 01_B or at bus off end when the BOM[1:0] bits are 10_B), the program's writing takes precedence. Modify the BOM[1:0] bits only in channel reset mode.

TDCVFIE Bit

When the TDCVF flag in the RSCFDnCFDCmFDSTS register is set to 1 after the TDCVFIE bit is set to 1, an interrupt request occurs. Modify this bit only in channel reset mode.

SOCOIE Bit

When the SOCO flag in the RSCFDnCFDCmFDSTS register is set to 1 after the SOCOIE bit is set to 1, an interrupt request occurs. Modify this bit only in channel reset mode.

EOCOIE Bit

When the EOCO flag in the RSCFDnCFDCmFDSTS register is set to 1 after the EOCOIE bit is set to 1, an interrupt request occurs. Modify this bit only in channel reset mode.

TAIE Bit

When transmit abort of the transmit buffer is completed with the TAIE bit set to 1, an interrupt request is generated. Modify this bit only in channel reset mode.

ALIE Bit

When the ALF flag in the RSCFDnCFDCmERFL register is set to 1 with the ALIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BLIE Bit

When the BLF flag in the RSCFDnCFDCmERFL register is set to 1 with the BLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

OLIE Bit

When the OVLf flag in the RSCFDnCFDCmERFL register is set to 1 with the OLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BORIE Bit

When the BORF flag in the RSCFDnCFDCmERFL register is set to 1 with the BORIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BOEIE Bit

When the BOEF flag in the RSCFDnCFDCmERFL register is set to 1 with the BOEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EPIE Bit

When the EPF flag in the RSCFDnCFDCmERFL register is set to 1 with the EPIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EWIE Bit

When the EWF flag in the RSCFDnCFDCmERFL register is set to 1 with the EWIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BEIE Bit

When the BEF flag in the RSCFDnCFDCmERFL register is set to 1 with the BEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

RTBO Bit

Setting this bit to 1 in the bus off state forcibly returns from the bus off state to the error active state. This bit is automatically cleared to 0. Setting this bit to 1 clears the TEC[7:0] and REC[7:0] bits in the RSCFDnCFDCmSTS register to 00_H and also clears the BOSTS flag in the RSCFDnCFDCmSTS register to 0 (not in bus off state). The other registers remain unchanged. No bus off recovery interrupt request is generated upon return from the bus off state in this case. Use this bit only when the BOM[1:0] bits in the RSCFDnCFDCmCTR register are 00_B (ISO11898-1 compliant).

A delay of up to 1 CAN bit time occurs after the RTBO bit is set to 1 until the RSCFD module transitions to the error active state. Set this bit to 1 in channel communication mode.

CSLPR Bit

Setting this bit to 1 places the channel into channel stop mode.

Clearing this bit to 0 makes the channel exit channel stop mode.

This bit should not be modified in channel communication mode or channel halt mode.

CHMDC[1:0] Bits

These bits are used to select a channel mode (channel communication mode, channel reset mode, or channel halt mode). For details, see **Section 20.6.2, Channel Modes**. Setting the CSLPR bit to 1 in channel reset mode allows transition to channel stop mode. Do not set the CHMDC[1:0] bits to 11_B. When the CAN module has automatically transitioned to channel halt mode based on the setting of the BOM[1:0] bits, the CHMDC[1:0] bits are automatically set to 10_B.

20.4.3.3 RSCFDnCFDCmSTS — Channel m Status Register

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0005_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEC[7:0]								REC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ESIF	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R	R	R/W*1	R	R	R	R	R	R	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 20.102 RSCFDnCFDCmSTS Register Contents

Bit Position	Bit Name	Function
31 to 24	TEC[7:0]	The transmit error counter (TEC) can be read.
23 to 16	REC[7:0]	The receive error counter (REC) can be read.
15 to 9	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
8	ESIF	Error State Indication Flag 0: No CAN FD message whose ESI bit is recessive has been received. 1: At least one CAN FD message whose ESI bit is recessive has been received.
7	COMSTS	Communication Status Flag 0: Communication is not ready. 1: Communication is ready.
6	RECSTS	Receive Status Flag 0: Bus idle, in transmission or bus off state 1: In reception
5	TRMSTS	Transmit Status Flag 0: Bus idle or in reception 1: In transmission or bus off state
4	BOSTS	Bus Off Status Flag 0: Not in bus off state 1: In bus off state
3	EPSTS	Error Passive Status Flag 0: Not in error passive state 1: In error passive state
2	CSLPSTS	Channel Stop Status Flag 0: Not in channel stop mode 1: In channel stop mode
1	CHLTSTS	Channel Halt Status Flag 0: Not in channel halt mode 1: In channel halt mode
0	CRSTSTS	Channel Reset Status Flag 0: Not in channel reset mode 1: In channel reset mode

TEC[7:0] Bits

These bits contain the transmit error counter value. For transmit error counter increment/decrement conditions, see *the CAN specifications (ISO11898-1)*.

These bits are cleared to 0 in channel reset mode.

REC[7:0] Bits

These bits contain the receive error counter value. For receive error counter increment/decrement conditions, see *the CAN specifications (ISO11898-1)*.

These bits are cleared to 0 in channel reset mode.

ESIF Flag

When the recessive ESI bit is detected in a successfully received message, this flag is set to 1. In loopback mode or mirror mode, the own transmission message is regarded as a received message. To clear this flag to 0, write 0 to this bit by the program. This bit cannot be set to 1 by the program. If the flag setting (to 1) timing matches the writing 0 (by the program) timing, this flag is set to 1.

This flag is set to 0 in channel reset mode.

COMSTS Flag

This bit indicates that communication is ready.

This flag is set to 1 when the CAN module has detected 11 consecutive recessive bits after it has transitioned from channel reset mode or channel halt mode to channel communication mode. This flag is cleared to 0 in channel reset mode or channel halt mode.

RECSTS Flag

This flag is set to 1 when reception has started, and is cleared to 0 when the bus has become idle or transmission has started.

TRMSTS Flag

This flag is set to 1 when transmission has started, and is cleared to 0 when the bus has become idle or reception has started. This flag remains 1 in the bus off state.

BOSTS Flag

This flag is set to 1 when the bus off state ($TEC[7:0] > 255$) is entered. It is cleared to 0 when the CAN module has exited the bus off state.

EPSTS Flag

This flag is set to 1 when the RS-CANFD module has entered the error passive state ($(128 \leq TEC[7:0] \leq 255)$ or $(128 \leq REC[7:0])$), It is cleared to 0 when the RS-CANFD module has exited the error passive state or has entered channel reset mode.

CSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel stop mode, and is cleared to 0 when the CAN module has returned from channel stop mode.

CHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel halt mode, and is cleared to 0 when the CAN module has returned from channel halt mode.

CRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel reset mode, and is cleared to 0 when the CAN module has transitioned to channel communication mode or channel halt mode. This flag remains 1 when the CAN module transitions from channel reset mode to channel stop mode.

20.4.3.4 RSCFDnCFDCmERFL — Channel m Error Flag Register

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CRCREG[14:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLFF	BORF	BOEF	EPF	EWFF	BEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 20.103 RSCFDnCFDCmERFL Register Contents (1/2)

Bit Position	Bit Name	Function
31	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
30 to 16	CRCREG[14:0]	CRC Calculation Data (CRC length:15 bits) A CRC value calculated based on the transmit message or receive message is indicated.
15	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
14	ADERR	ACK Delimiter Error Flag 0: No ACK delimiter error is detected. 1: ACK delimiter error is detected.
13	B0ERR	Dominant Bit Error Flag 0: No dominant bit error is detected. 1: Dominant bit error is detected.
12	B1ERR	Recessive Bit Error Flag 0: No recessive bit error is detected. 1: Recessive bit error is detected.
11	CERR	CRC Error Flag 0: No CRC error is detected. 1: CRC error is detected.
10	AERR	ACK Error Flag 0: No ACK error is detected. 1: ACK error is detected.
9	FERR	Form Error Flag 0: No form error is detected. 1: Form error is detected.
8	SERR	Stuff Error Flag 0: No stuff error is detected. 1: Stuff error is detected.
7	ALF	Arbitration-lost Flag 0: No arbitration-lost is detected. 1: Arbitration-lost is detected.

Table 20.103 RSCFDnCFDCmERFL Register Contents (2/2)

Bit Position	Bit Name	Function
6	BLF	Bus Lock Flag 0: No channel bus lock is detected. 1: Channel bus lock is detected.
5	OVLf	Overload Flag 0: No overload is detected. 1: Overload is detected.
4	BORF	Bus Off Recovery Flag 0: No bus off recovery is detected. 1: Bus off recovery is detected.
3	BOEF	Bus Off Entry Flag 0: No bus off entry is detected. 1: Bus off entry is detected.
2	EPF	Error Passive Flag 0: No error passive is detected. 1: Error passive is detected.
1	EWf	Error Warning Flag 0: No error warning is detected. 1: Error warning is detected.
0	BEF	Bus Error Flag 0: No channel bus error is detected. 1: Channel bus error is detected.

See the *CAN specification (ISO11898-1)* for a description of error occurrence conditions. To clear each flag of this register, the program must write a 0 to the corresponding bit. These flags cannot be set to 1 by the program. If any of these flags is set to 1 at the same time that the program writes 0 to the flag, the flag is set to 1. Transition to channel reset mode resets these flags to 0.

If the ERRD bit in the RSCFDnCFDCmCTR register is set to 0 (i.e., only the flags for the first error event are displayed) and an error related to bits 14 to 8 of RSCFDnCFDCmERFL is detected, the flag bits are only set to 1 by the error event if bits 14 to 8 were all 0 at the time when the error occurred.

CRCREG[14:0] Flag

When the CTME bit in the RSCFDnCFDCmCTR register is set to 1 (communication test mode is enabled), if transmit or receive message is a classical CAN frame (CRC length = 15 bits), this flag is updated and the CRC value calculated based on the message can be read. When a CAN FD frame is sent or received, the value of CRCREG[20:0] bits in the RSCFDnCFDCmFDCRC register is updated. When the CTME bit is set to 0 (communication test mode is disabled), these bits are always read as 0. These bits are always 0 in channel reset mode.

ADERR Flag

This flag is set to 1 when a form error has been detected in the ACK delimiter during transmission.

B0ERR Flag

This flag is set to 1 when a recessive bit has been detected even though a dominant bit was transmitted.

B1ERR Flag

This flag is set to 1 when a dominant bit has been detected even though a recessive bit was transmitted.

CERR Flag

This flag is set to 1 when a CRC error has been detected.

AERR Flag

This flag is set to 1 when an ACK error has been detected.

FERR Flag

This flag is set to 1 when a form error has been detected.

SERR Flag

This flag is set to 1 when a stuff error has been detected.

ALF Flag

This flag is set to 1 when an arbitration-lost has been detected.

BLF Flag

This flag is set to 1 when 32 consecutive dominant bits have been detected on the CAN bus in channel communication mode. After that, detection of a dominant lock is restarted when either of the following conditions is met.

- A recessive bit is detected after the BLF bit has been cleared from 1 to 0.
- The CAN module transitions to channel reset mode and returns to channel communication mode after the BLF bit has been cleared from 1 to 0.

OVLf Flag

This flag is set to 1 when the overload frame transmit condition has been detected when performing reception or transmission.

BORF Flag

This flag is set to 1 when 11 consecutive recessive bits have been detected 128 times and the CAN module returns from the bus off state. However, this flag is not set to 1 if the CAN module returns from the bus off state in any of the following ways before 11 consecutive recessive bits are detected 128 times.

- The CHMDC[1:0] bits in the RSCFDnCFDCmCTR register are set to 01_B (channel reset mode).
- The RTBO bit in the RSCFDnCFDCmCTR register is set to 1 (forcible return from the bus off state).
- The BOM[1:0] bits in the RSCFDnCFDCmCTR register are set to 01_B (transition to channel halt mode at bus off entry).
- The CHMDC[1:0] bits in the RSCFDnCFDCmCTR register are set to 10_B (channel halt mode) before 11 consecutive recessive bits are detected 128 times with the BOM[1:0] bits set to 11_B (transition to channel halt mode upon a request from the program during bus off).

BOEF Flag

This flag is set to 1 when the bus off state is entered (TEC[7:0] value > 255). This flag is also set to 1 if the bus off state is entered when the BOM[1:0] bits in the RSCFDnCFDCmCTR register (m = see **Table 20.6**) are set to 01_B (transition to channel halt mode at bus off entry).

EPF Flag

This flag is set to 1 when the error passive state is entered (REC[7:0] or TEC[7:0] value > 127).

This flag is set to 1 only when the REC[7:0] or TEC[7:0] value first exceeds 127. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 127, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 127 or less and then the REC[7:0] or TEC[7:0] value exceeds 127 again.

EWf Flag

This flag is set to 1 only when the REC[7:0] or TEC[7:0] value first exceeds 95. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 95, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 95 or less and then the REC[7:0] or TEC[7:0] value exceeds 95 again.

BEF Flag

This flag is set to 1 when any one of the ADERR, B0ERR, B1ERR, CERR, AERR, FERR, and SERR flags in the RSCFDnCFDCmERFL register is set to 1.

NOTE

To clear the flag of this register to 0, use a store instruction to write "0" to the given flag and "1" to the other flags.

20.4.3.5 RSCFDnCFDCmDCFG — Channel m Data Bit Rate Configuration Register

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	DSJW[2:0]			—	DTSEG2[2:0]			DTSEG1[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DBRP[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.104 RSCFDnCFDCmDCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
26 to 24	DSJW[2:0]	Data Bit Rate Resynchronization Jump Width Control b26 b25 b24 0 0 0: 1 Tq 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq
23	—	Reserved This bit is read as the value after reset. The write value should be the value after reset.
22 to 20	DTSEG2[2:0]	Data Bit Rate Time Segment 2 Control b22 b21 b20 0 0 0: Setting is prohibited for channels for CAN communications. 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq

Table 20.104 RSCFDnCFDCmDCFG Register Contents (2/2)

Bit Position	Bit Name	Function
19 to 16	DTSEG1[3:0]	Data Bit Rate Time Segment 1 Control b19 b18 b17 b16 0 0 0 0: Setting is prohibited for channels for CAN communications. 0 0 0 1: 2 Tq 0 0 1 0: 3 Tq 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq
15 to 8	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
7 to 0	DBRP[7:0]	Data Bit Rate Prescaler Division Ratio Setting When the set value = P (0 to 255), the data bit rate prescaler divides fCAN by (P + 1).

Modify the RSCFDnCFDCmDCFG register in channel reset mode or channel halt mode. Set this register in channel reset mode, and then transition to channel communication mode or channel halt mode. When only classical CAN frames are used in CAN FD mode, set the RSCFDnCFDCmDCFG register to the value equal to the set RSCFDnCFDCmNCFG register value. For the description and settings of bit timing parameters, see **Section 20.11.1, Initial Settings**.

DSJW[2:0] Bits

These bits specify the resynchronization jump width of data bit rate as a Tq value. Possible values are 1 to 8 Tq. Specify a value equal to or smaller than the DTSEG2[2:0] bits value.

DTSEG2[2:0] Bits

These bits are used to specify a Tq value for the length of phase segment 2 (PHASE_SEG2) of data bit rate.

Possible values are 2 to 8 Tq.

Specify a value equal to or smaller than the DTSEG1[3:0] bits value.

DTSEG1[3:0] Bits

These bits specify the total length of propagation segment (PROP_SEG) and phase segment 1 (PHASE_SEG1) of data bit rate as a Tq value.

Possible values are 2 to 16 Tq.

DBRP[7:0] Bits

The clock obtained by dividing the CAN clock (fCAN) by the data bit rate prescaler ((DBRP[7:0]) + 1) becomes CANmTq(D) clock (fCANTQ(D)m). One clock of the CANmTq(D) clock becomes one Time Quantum (Tq).

Be sure to specify the same value for both NBRP[9:0] and DBRP[7:0].

To specify different values for the nominal bit rate and the data bit rate, change the values of the RSCFDnCFDCmNCFG.NTSEG1 and NTSEG2 bits and RSCFDnCFDCmDCFG.DTSEG1 and DTSEG2 bits, respectively.

When the TDCE bit is set to 1 (Transmitter delay compensation is enabled) in the RSCFDnCFDCmFDCFG register, set the equal value of 1 or less to the bits NBRP[9:0] and DBRP[7:0].

20.4.3.6 RSCFDnCFDCmFDCFG — Channel m CAN FD Configuration Register

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	REFE	FDOE	TMME	GWBR S	GWDFD	GWEN	—	TDCO[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ESIC	TDCE	TDCOC	—	—	—	—	—	EOCCFG[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 20.105 RSCFDnCFDCmFDCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 30	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
29	REFE	Reception data edge filter enable 0: Reception data edge filter is disabled 1: Reception data edge filter is enabled
28	FDOE	FD only enable 0: FD-only mode is disabled 1: FD-only mode is enabled
27	TMME	Transmit Buffer Merge Mode Enable 0: Transmit buffer merge mode is disabled. 1: Transmit buffer merge mode is enabled.
26	GWBR S	Gateway BRS 0: A frame is transmitted with the BRS bit in the received frame set to 0. 1: A frame is transmitted with the BRS bit in the received frame set to 1.
25	GWDFD	Gateway FDF 0: A frame is transmitted regarding the received frame as a classical CAN frame. 1: A frame is transmitted regarding the received frame as a CAN FD frame.
24	GWEN	CAN-CAN FD Gateway Enable 0: The CAN-CAN FD gateway is disabled. 1: The CAN-CAN FD gateway is enabled.
23	—	Reserved This bit is read as the value after reset. The write value should be the value after reset.
22 to 16	TDCO[6:0]	Transmitter Delay Compensation Offset These bits are set to the transmitter delay compensation offset value.
15 to 11	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
10	ESIC	Error State Display Mode Select 0: Always displays the node error state. 1: When the node is not in the error passive state: Displays the message buffer error state. When the node is in the error passive state: Displays the node error state.

Table 20.105 RSCFDnCFDCmFDCFG Register Contents (2/2)

Bit Position	Bit Name	Function																																				
9	TDCE	Transmitter Delay Compensation Enable 0: Transmitter delay compensation is disabled. 1: Transmitter delay compensation is enabled.																																				
8	TDCOC	Transmitter Delay Compensation Measurement Select 0: Measurement and offset 1: Only offset																																				
7 to 3	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.																																				
2 to 0	EOCCFG[2:0]	Error Occurrence Counting Method Select <table border="0"> <tr> <td>b2</td> <td>b1</td> <td>b0</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0: All transmit messages and receive messages</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1: All transmit messages</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0: All receive messages</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1: Setting prohibited</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0: Only data phase of transmitted or received CAN FD message</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1: Only data phase of transmitted CAN FD message</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0: Only data phase of received CAN FD message</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1: Setting prohibited</td> </tr> </table>	b2	b1	b0		0	0	0	0: All transmit messages and receive messages	0	0	1	1: All transmit messages	0	1	0	0: All receive messages	0	1	1	1: Setting prohibited	1	0	0	0: Only data phase of transmitted or received CAN FD message	1	0	1	1: Only data phase of transmitted CAN FD message	1	1	0	0: Only data phase of received CAN FD message	1	1	1	1: Setting prohibited
b2	b1	b0																																				
0	0	0	0: All transmit messages and receive messages																																			
0	0	1	1: All transmit messages																																			
0	1	0	0: All receive messages																																			
0	1	1	1: Setting prohibited																																			
1	0	0	0: Only data phase of transmitted or received CAN FD message																																			
1	0	1	1: Only data phase of transmitted CAN FD message																																			
1	1	0	0: Only data phase of received CAN FD message																																			
1	1	1	1: Setting prohibited																																			

REFE Bit

Setting this bit to 1 enables reception data edge filtering when the idle condition is detected, and a dominant level with less than 2 time quanta is ignored. A dominant level with more than or equal to 2 time quanta is detected as an edge.

Modify this bit only in channel reset mode.

FDOE Bit

Setting this bit to 1 enables FD-only mode. When data is transmitted, a CAN FD frame will be sent regardless of the settings to the CFFDF bit in the RSCFDnCFDCFFDCSTSk register or the TMFDF bit in the RSCFDnCFDTMFDCTRp register. When a Classical CAN frame is received, a form error is detected. Modify this bit only in channel reset mode.

TMME Bit

Setting this bit to 1 enables transmit buffer merge mode. Modify this bit only in channel reset mode or channel halt mode.

GWBRs Bit

When the GWEN bit is 1, the BRS bit in a CAN FD frame to be transmitted by the gateway function is set. When the GWFDF bit is set to 0, write 0 to this bit. Modify this bit only in channel reset mode.

GWFDF Bit

When the GWEN bit is 1, the FDF bit in a CAN FD frame to be transmitted by the gateway function is set. Modify this bit only in channel reset mode.

GWEN Bit

This bit is used to control the operation of the transmit/receive FIFO buffer with the CFM[1:0] bits in the RSCFDnCFDCFCCK register set to 10_B (gateway mode).

Setting this bit to 1 enables the CAN-CAN FD gateway, enabling transmission in a format different from that of frames received by the gateway function. Received frames are replaced in accordance with the settings of the GWFDF bit and the GWBRs bit. When the DLC value in the received classical CAN frame is 1001_B or more and the GWFDF bit is set to 1 (CAN FD frame), the DLC value is replaced with 1000_B.

While this bit is set to 1, do not perform routing the following frames by using the gateway function.

- CAN FD frames with a payload length of more than 8 bytes
- Remote frames

While this bit is set to 1, the following frame should be transmitted from the channel according to the setting of GWDFD.

- When GWDFD bit is set to 0, only classical CAN frame should be transmitted.
- When GWDFD bit is set to 1, only CAN FD frame should be transmitted.

Modify this bit only in channel reset mode.

Table 20.106 shows the settings and formats of transmit frame and receive frame while the CAN-CAN FD gateway is enabled.

Table 20.106 Operation when the CAN-CAN FD Gateway is Enabled

Receive Frame			GWDFD Bit	Transmit Frame		
Format	BRS Bit	Received DLC Value		Format	BRS Bit	DLC Value to be Transmitted
Classical CAN	None	DLC ≤ 1000 _B	0	Classical CAN	None	Not replaced
		DLC > 1000 _B				
CAN FD	Arbitrary	DLC ≤ 1000 _B				
Classical CAN	None	DLC ≤ 1000 _B				
		DLC >1000 _B	Replaced with 1000 _B			
CAN FD	Arbitrary	DLC ≤ 1000 _B	Not replaced			

TDCO[6:0] Bits

These bits set the SSP offset value. How to use this value depends on the TDCOC bit in the RSCFDnCFDCmFDCFG register. These bits are based on CAN clock frequency(fCAN).

When the TDCOC bit is set to 0, the transmitter delay compensation result equals to the total value of the measured delay value and the TDCO[6:0] value (rounded down to the nearest integer Tq).

When the TDCOC bit is set to 1, the transmitter delay compensation result equals to the TDCO[6:0] value.

The SSP offset value = (set value of TDCO[6:0] bits + 1).

Modify these bits only in channel reset mode or channel halt mode.

ESIC Bit

When the ESIC bit is set to 1, if the channel is in the error active state, the ESI bit value (CFESI bit in the RSCFDnCFDCFFDCSTSk register or TMESI bit in the RSCFDnCFDTMFDCTRp register) set in the transmit/receive FIFO buffer or transmit buffer is transmitted as an ESI bit value of the transmit message. When the channel is in the error passive state or the ESIC bit is set to 0, the channel status is transmitted as an ESI bit value. Modify this bit only in channel reset mode or channel halt mode.

Table 20.107 ESI Value to Be Transmitted

ESIC Bit	Channel Status	ESI Value to be Transmitted
0	Error active	0 (error active node)
	Error passive	1 (error passive node)
1	Error active	ESI value set in the transmit/receive FIFO buffer or transmit buffer (CFESI bit in the RSCFDnCFDCFFDCSTSk register or TMESI bit in the RSCFDnCFDTMFDCTRp register)
	Error passive	1 (error passive node)

TDCE Bit

Setting this bit to 1 enables transmitter delay compensation. Modify this bit only in channel reset mode or channel halt mode.

TDCOC Bit

When this bit is set to 0, the SSP position is defined by the total of the measured delay value and the SSP offset value (fixed value).

When this bit is set to 1, the SSP position is defined only by the SSP offset value.

Modify this bit only in channel reset mode or channel halt mode.

EOCCFG[2:0] Bits

These bits are used to select a frame format and a transmission/reception direction when the error occurrence counter counts CAN bus errors.

Modify these bits only in channel reset mode or channel halt mode.

20.4.3.7 RSCFDnCFDCmFDCTR — Channel m CAN FD Control Register

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOCCLR	EOCCLR
															R	R
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 20.108 RSCFDnCFDCmFDCTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
1	SOCCLR	Successful Occurrence Counter Clear Setting the SOCCLR bit to 1 clears the successful occurrence counter. This bit is always read as 0.
0	EOCCLR	Error Occurrence Counter Clear Setting the EOCCLR bit to 1 clears the error occurrence counter. This bit is always read as 0.

SOCCLR Bit

Setting this bit to 1 clears the successful occurrence counter (SOC[7:0] bits in the RSCFDnCFDCmFDSTS register). This bit is automatically cleared to 0.

EOCCLR Bit

Setting this bit to 1 clears the error occurrence counter (EOC[7:0] bits in the RSCFDnCFDCmFDSTS register). This bit is automatically cleared to 0.

20.4.3.8 RSCFDnCFDCmFDSTS — Channel m CAN FD Status Register

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SOC[7:0]								EOC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SOCO	EOCO	TDCVF	TDCR[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R	R	R	R	R	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 20.109 RSCFDnCFDCmFDSTS Register Contents

Bit Position	Bit Name	Function
31 to 24	SOC[7:0]	Successful Occurrence Counter The successful occurrence counter value can be read.
23 to 16	EOC[7:0]	Error Occurrence Counter The error occurrence counter value can be read.
15 to 10	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
9	SOCO	Successful Occurrence Counter Overflow Flag 0: The successful occurrence counter does not overflow. 1: The successful occurrence counter has overflowed.
8	EOCO	Error Occurrence Counter Overflow Flag 0: The error occurrence counter does not overflow. 1: The error occurrence counter has overflowed.
7	TDCVF	Transmitter Delay Compensation Violation Flag 0: No transmitter delay compensation violation is present. 1: A transmitter delay compensation violation is present.
6 to 0	TDCR[6:0]	Transmitter Delay Compensation Result Status The transmitter delay compensation result can be read.

SOC[7:0] Bits

These bits show the successful occurrence counter value. The successful occurrence counter is incremented upon completion of message reception or transmission without an error. This counter stops counting when it reaches FF_H. In loopback mode, this counter is incremented twice.

These bits are cleared to 0 by writing 1 to the SOCCLR bit in the RSCFDnCFDCmCTR register. These bits are set to 0 in channel reset mode.

EOC[7:0] Bits

These bits show the error occurrence counter value. The error occurrence counter is incremented each time an error occurs according to the condition specified by the EOCCFG[2:0] bits in the RSCFDnCFDCmFDCFG register. This counter stops counting when it reaches FF_H.

These bits are cleared to 0 by writing 1 to the EOCCLR bit in the RSCFDnCFDCmCTR register. These bits are set to 0 in channel reset mode.

SOCO Flag

This bit indicates that successful occurrence counter overflow has occurred.

This flag is set to 1 when message reception or transmission is completed while the SOC[7:0] value has reached FF_H. This flag is set to 0 in channel reset mode.

EOCO Flag

This bit indicates that error occurrence counter overflow has occurred.

This flag is set to 1 when a CAN bus error is detected under the condition specified by the EOCCFG[2:0] bits in the RSCFDnCFDCmFDCFG register when the EOC[7:0] value has reached FF_H. This flag is set to 0 in channel reset mode.

TDCVF Flag

This bit indicates violation of transmitter delay compensation.

The transmit data is compared with the reception CAN bus level delayed due to the transceiver's loop delay. This delay changes due to physical factors such as temperature. Because the TDCR[6:0] flags are updated for each message, temporary maximum delay cannot be confirmed.

This bit is set to 1 when the transmitter delay compensation exceeds the maximum compensation 3 CANm bit times - 2 fCAN (CANm bit time is the value of data bit rate).

This flag is set to 0 in channel reset mode.

TDCR[6:0] Flags

These bits indicate the transmitter delay compensation result as a multiple of CAN clock frequency (fCAN).

This result depends on the settings of the TDCOC bit and TDCO[6:0] bits in the RSCFDnCFDCmFDCFG register.

These flags are updated at a falling edge between the FDF bit and res bit when the TDCE bit in the RSCFDnCFDCmFDCFG register is set to 1 (transmitter delay compensation enable) and also the TDCOC bit in the RSCFDnCFDCmFDCFG register is set to 0 (measurement and offset).

This flag is set to 0 in channel reset mode.

NOTE

To clear the flag of this register to 0, use a store instruction to write 0 to the given flag and 1 to the other flags.

20.4.3.9 RSCFDnCFDCmFDCRC — Channel m CAN FD CRC Register

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	SCNT[3:0]			—	—	—	CRCREG[20:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRCREG[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.110 RSCFDnCFDCmFDCRC Register Contents

Bit Position	Bit Name	Function
31 to 28	—	Reserved These bits are read as the value after reset.
27 to 24	SCNT[3:0]	Stuff count bit Indicate a value of the stuff count in a CAN FD frame. Bits 25 to 27 indicates the Gray-coded value of the stuff bit count modulo 8 in the transmitted/received frames. Bit 24 indicates an even parity value of bits 25 to 27.
23 to 21	—	Reserved These bits are read as 0. The write value should be always 0.
20 to 0	CRCREG[20:0]	CRC Calculation Data (CRC Length:17 Bit or 21 Bit) These bits show the CRC value calculated based on the transmit message or receive message. When the CRC length is 17 bits, bits b20 to b17 are read as 0.

SCNT[3:0] Flags

When the CTME bit in the RSCFDnCFDCmCTR register is set to 1 (communication test mode enabled), a stuff count bit value of the CAN FD frame can be read if a message transmitted/received is a CAN FD frame. When the CTME bit is 0 (communication test mode disabled), this flag is always read as 0. These flags are updated at the first bit in the CRC field of the CAN FD frame. These bits are cleared to 0 in channel reset mode.

CRCREG[20:0] Flags

When the CTME bit in the RSCFDnCFDCmCTR register is 1 (communication test mode enabled), if transmit or receive message is a CAN FD frame (CRC length = 17 or 21 bits), these flags are updated and the CRC value calculated based on the message can be read. When the CRC length of the message is 17 bits, bits b20 to b17 are always read as 0. When a classical CAN frame is transmitted or received, the CRCREG[14:0] value in the RSCFDnCFDCmERFL register is updated. When the CTME bit is 0 (communication test mode disabled), these bits are always read as 0.

These bits are cleared to 0 in channel reset mode.

20.4.4 Details of Global-Related Registers

20.4.4.1 RSCFDnCFDGCFG — Global Configuration Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ITRCP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSBTCS[2:0]		TSSS	TSP[3:0]			—	—	CMPOC	DCS	MME	DRE	DCE	TPRI		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.111 RSCFDnCFDGCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	ITRCP[15:0]	Interval Timer Prescaler Set When these bits are set to M, the pclk is divided by M. Setting 0000 _H is prohibited when the interval timer is in use.
15 to 13	TSBTCS[2:0]	Timestamp Clock Source Select b15 b14 b13 0 0 0: Channel 0 nominal bit time clock 0 0 1: Channel 1 nominal bit time clock 0 1 0: Channel 2 nominal bit time clock 0 1 1: Channel 3 nominal bit time clock 1 0 0: Channel 4 nominal bit time clock 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited
12	TSSS	Timestamp Source Select 0: pclk/2 ^{*1} 1: Nominal bit time clock
11 to 8	TSP[3:0]	Timestamp Clock Source Division b11 b10 b9 b8 0 0 0 0: Not divided 0 0 0 1: Divided by 2 0 0 1 0: Divided by 4 0 0 1 1: Divided by 8 0 1 0 0: Divided by 16 0 1 0 1: Divided by 32 0 1 1 0: Divided by 64 0 1 1 1: Divided by 128 1 0 0 0: Divided by 256 1 0 0 1: Divided by 512 1 0 1 0: Divided by 1024 1 0 1 1: Divided by 2048 1 1 0 0: Divided by 4096 1 1 0 1: Divided by 8192 1 1 1 0: Divided by 16384 1 1 1 1: Divided by 32768

Table 20.111 RSCFDnCFDGCFCFG Register Contents (2/2)

Bit Position	Bit Name	Function
7, 6	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
5	CMPOC	Payload Overflow Mode Select 0: No message is stored. 1: Messages are stored and payloads exceeding the buffer size are discarded.
4	DCS	CAN Clock Source Select 0: clkc 1: clk_xincan
3	MME	Mirror Function Enable 0: Mirror function is disabled. 1: Mirror function is enabled.
2	DRE	DLC Replacement Enable 0: DLC replacement is disabled. 1: DLC replacement is enabled.
1	DCE	DLC Check Enable 0: DLC check is disabled. 1: DLC check is enabled.
0	TPRI	Transmit Priority Select 0: ID priority 1: Transmit buffer number priority

Note 1. When specifying pclk/2 as the timestamp counter count source, set bits TSBTCS[2:0] to 000_B.

Modify the RSCFDnCFDGCFCFG register only in global reset mode.

ITRCP[15:0] Bits

These bits are used to set a clock source division value of the interval timer for FIFO buffers. See **Section 20.8.3.1, Interval Transmission Function**.

TSBTCS[2:0] Bits

When the TSSS bit is 1, these bits are used to select the channel of the nominal bit time clock that will be the clock source of the timestamp counter. However, do not select the channel that handles the CAN FD frames.

TSSS Bit

This bit is used to select a clock source of the timestamp counter. Select pclk if there is no channel that handles only classical CAN frames.

TSP[3:0] Bits

A clock obtained by dividing the clock source selected with the TSBTCS[2:0] bits and TSSS bit according to the TSP[3:0] bits is used as the timestamp counter count source.

CMPOC Bit

This bit is used to select operation in case the payload length of received message exceeds the payload storage size of the storage buffer.

When this bit is 0, the received message in which the payload overflows is not stored in the buffer.

When this bit is 1, the received message in which the payload overflows is stored in the buffer, and depending on the DRE bit the received DLC value or the DLC value of the receive rule is stored in the buffer. At this time, payloads exceeding the buffer's payload storage size are discarded.

The buffer's payload storage size is set by the following bits.

- Receive buffer: RMPLS[1:0] bits in the RSCFDnCFDRMNB register
- Receive FIFO buffer: RFPLS[2:0] bits in the RSCFDnCFDRFCCx register
- Transmit/receive FIFO buffer: CFPLS[2:0] bits in the RSCFDnCFDCFCCK register

DCS Bit

When this bit is set to 0, clk_c is used as the clock source of the CAN clock (fCAN).

When this bit is set to 1, clk_xincan is used as the clock source of the CAN clock (fCAN).

MME Bit

Setting this bit to 1 makes the mirror function available.

DRE Bit

When the DRE bit is set to 1, the DLC value of the receive rule is stored in the buffer instead of the DLC value of the received message after the DLC value has passed through the DLC filter. In this case, a value of 00_H is stored in each data byte that exceeds the DLC value of the receive rule.

The DLC replacement function is only available when the DCE bit is set to 1 (DLC check is enabled).

DCE Bit

Setting this bit to 1 makes the DLC check function available. When disabling the DLC check function, set the GAFLDLC[3:0] bits in the RSCFDnCFDGAFLP0_j register to 0000_B before clearing the DCE bit in the RSCFDnCFDGCFCFG register to 0.

TPRI Bit

This bit is used to set the transmit priority.

When this bit is set to 0, ID priority is selected and the transmit priority complies with the CAN bus arbitration rule (ISO11898-1 specifications). When this bit is set to 1, transmit buffer number priority is selected and the transmit buffer with the smallest number has the highest priority.

When using the transmit queue, this bit should be set to 0.

20.4.4.2 RSCFDnCFDGCTR — Global Control Register

Value after reset: 0000 0005_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CMPOF IE	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 20.112 RSCFDnCFDGCTR Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
16	TSRST	Timestamp Counter Reset Setting the TSRST bit to 1 resets the timestamp counter. This bit is always read as 0.
15 to 12	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
11	CMPOFIE	Payload Overflow Interrupt Enable 0: A payload overflow interrupt is disabled. 1: A payload overflow interrupt is enabled.
10	THLEIE	Transmit History Buffer Overflow Interrupt Enable 0: Transmit history buffer overflow interrupt is disabled. 1: Transmit history buffer overflow interrupt is enabled.
9	MEIE	FIFO Message Lost Interrupt Enable 0: FIFO message lost interrupt is disabled. 1: FIFO message lost interrupt is enabled.
8	DEIE	DLC Error Interrupt Enable 0: DLC error interrupt is disabled. 1: DLC error interrupt is enabled.
7 to 3	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	GSLPR	Global Stop Mode 0: Other than global stop mode 1: Global stop mode
1, 0	GMDC[1:0]	Global Mode Select b1 b0 0 0: Global operating mode 0 1: Global reset mode 1 0: Global test mode 1 1: Setting prohibited

TSRST Bit

This bit is used to reset the timestamp counter. When this bit is set to 1, the RSCFDnCFDGTSC register is cleared to 0000_H.

CMPOFIE Bit

When the CMPOF flag in the RSCFDnCFDGERFL register is set to 1 after the CMPOFIE bit is set to 1, an interrupt request occurs. Modify this bit only in global reset mode.

THLEIE Bit

When the THLEIE bit is set to 1 and the THLES flag in the RSCFDnCFDGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

MEIE Bit

When the MEIE bit is set to 1 and the MES flag in the RSCFDnCFDGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

DEIE Bit

When the DEIE bit is set to 1 and the DEF flag in the RSCFDnCFDGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

GSLPR Bit

Setting this bit to 1 places the RS-CANFD module into global stop mode.

Clearing this bit to 0 releases the RS-CANFD module from global stop mode.

This bit should not be modified in global operating mode or global test mode.

GMDC[1:0] Bits

These bits are used to select the mode of entire RS-CANFD module (global operating mode, global reset mode, or global test mode). For details, see **Section 20.6.1, Global Modes**. Setting the GSLPR bit to 1 when in global reset mode places the RS-CANFD module into global stop mode.

20.4.4.3 RSCFDnCFDGSTS — Global Status Register

Value after reset: 0000 000D_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	GRAMI NIT	GSLPS TS	GHLT TS	GRSTS TS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.113 RSCFDnCFDGSTS Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned.
3	GRAMINIT	CAN RAM Initialization Status Flag 0: CAN RAM initialization is completed. 1: CAN RAM initialization is ongoing.
2	GSLPSTS	Global Stop Status Flag 0: Not in global stop mode 1: In global stop mode
1	GHLTSTS	Global Test Status Flag 0: Not in global test mode 1: In global test mode
0	GRSTSTS	Global Reset Status Flag 0: Not in global reset mode 1: In global reset mode

GRAMINIT Flag

This flag indicates the initialization status of the CAN RAM.

This flag is set to 1 after the MCU has been reset, and is cleared to 0 when CAN RAM initialization is completed.

GSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to global stop mode, and is cleared to 0 when the CAN module has returned from global stop mode.

GHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global test mode, and is cleared to 0 when the CAN module has exited global test mode.

GRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global reset mode, and is cleared to 0 when the CAN module has exited global reset mode. This flag remains 1 even when the CAN module has transitioned from global reset mode to global stop mode.

20.4.4.4 RSCFDnCFDGERFL — Global Error Flag Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	EEF4	EEF3	EEF2	EEF1	EEF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CMPOF	THLES	MES	DEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R	R	R/W*1

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 20.114 RSCFDnCFDGERFL Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 21	—	Reserved When read, the value after reset is returned. When writing these bits, write the value after reset.
20	EEF4	ECC Error Flag for Unit Channel 4 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
19	EEF3	ECC Error Flag for Unit Channel 3 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
18	EEF2	ECC Error Flag for Unit Channel 2 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
17	EEF1	ECC Error Flag for Unit Channel 1 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
16	EEF0	ECC Error Flag for Unit Channel 0 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
15 to 14, 7 to 6, 4	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
13 to 8, 5	—	Reserved When read, the undefined value is returned. When writing these bits, write the value after reset.
3	CMPOF	Payload Overflow Flag 0: No payload overflow has occurred. 1: A payload overflow has occurred.
2	THLES	Transmit History Buffer Overflow Status Flag 0: No transmit history buffer overflow has occurred. 1: A transmit history buffer overflow has occurred.
1	MES	FIFO Message Lost Status Flag 0: No FIFO message lost error has occurred. 1: A FIFO message lost error has occurred.

Table 20.114 RSCFDnCFDGERFL Register Contents (2/2)

Bit Position	Bit Name	Function
0	DEF	DLC Error Flag 0: No DLC error has occurred. 1: A DLC error has occurred.

All flags in the RSCFDnCFDGERFL register are cleared to 0 in global reset mode.

EEFm Flag

When a 2-bit ECC error is detected during the transmission priority determination of unit channel m (m = see **Table 20.6**), the EEFm flag is set to 1, disabling message transmission. This flag can be cleared to 0 by writing 0 by the program.

CMPOF Flag

When a payload overflow occurs in any of channel m (m = see **Table 20.6**), the CMPOF flag is set to 1.

This flag can be cleared to 0 by writing 0 to this bit by the program.

THLES Flag
The THLES flag is set to 1 when any one of the THLELT flags in the RSCFDnCFDTHLSTSm register (m = see **Table 20.6**) is set to 1.

This flag is cleared to 0 when the THLELT flags of all channels are set to 0.

MES Flag

The MES flag is set to 1 when any one of the RFMLT flags in the RSCFDnCFDRFSTStx register (x = see **Table 20.6**) or the CFMLT flags in the RSCFDnCFDCFSTStk register (k = see **Table 20.6**) is set to 1.

This flag is cleared to 0 when all RFMLT flags and CFMLT flags are set to 0.

DEF Flag

The DEF flag is set to 1 when an error has been detected during the DLC check. The program can clear this flag by writing 0 to this bit.

NOTE

To clear the flag of this register to 0, use a store instruction to write 0 to the given flag and 1 to the other flags.

20.4.4.5 RSCFDnCFDGTSC — Global Timestamp Counter Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.115 RSCFDnCFDGTSC Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved These bits are read as the value after reset.
15 to 0	TS[15:0]	Timestamp Value The timestamp counter value can be read. Counter Value: 0000 _H to FFFF _H

TS[15:0] Bits

When the TS[15:0] bits are read, the read value shows the timestamp counter (16-bit free-running counter) value at that time. When the SOF is detected, the TS[15:0] value is captured and later stored in the receive buffer or the FIFO buffer. Furthermore, the TS[15:0] value is stored in the transmit history buffer. The timestamp counter is initialized in global reset mode.

The timestamp counter starts and stops counting differently, depending on the count source.

- When the TSSS bit in the RSCFDnCFDGCFG register is 0 (pclk):
The timestamp counter starts counting when the RS-CANFD module has transitioned to global operating mode.
This counter stops counting when the RS-CANFD module has transitioned to global stop mode or global test mode.
- When the TSSS bit is 1 (CANm nominal bit time clock):
The timestamp counter starts counting when the corresponding channel has transitioned to channel communication mode.
This counter stops counting when the corresponding channel has transitioned to channel reset mode or channel halt mode.

20.4.4.6 RSCFDnCFDGTINTSTS0 — Global TX Interrupt Status Register 0

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	THIF3	CFTIF3	TQIF3	TAIF3	TSIF3	—	—	—	THIF2	CFTIF2	TQIF2	TAIF2	TSIF2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}	R	R	R	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THIF1	CFTIF1	TQIF1	TAIF1	TSIF1	—	—	—	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}	R	R	R	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}	R ^{*1}

Note 1. This bit is automatically cleared in the global reset or channel reset mode.

Table 20.116 RSCFDnCFDGTINTSTS0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	—	Reserved These bits are read as the value after reset. When writing to these bits, write the value after reset.
28	THIF3	Unit Channel 3 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
27	CFTIF3	Unit Channel 3 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
26	TQIF3	Unit Channel 3 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
25	TAIF3	Unit Channel 3 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
24	TSIF3	Unit Channel 3 Transmit Buffer Transmit Complete Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
23 to 21	—	Reserved These bits are read as the value after reset. When writing to these bits, write the value after reset.
20	THIF2	Unit Channel 2 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
19	CFTIF2	Unit Channel 2 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
18	TQIF2	Unit Channel 2 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
17	TAIF2	Unit Channel 2 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.

Table 20.116 RSCFDnCFDGTINTSTS0 Register Contents (2/2)

Bit Position	Bit Name	Function
16	TSIF2	Unit Channel 2 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
15 to 13	—	Reserved These bits are read as the value after reset.
12	THIF1	Unit Channel 1 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
11	CFTIF1	Unit Channel 1 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
10	TQIF1	Unit Channel 1 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
9	TAIF1	Unit Channel 1 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
8	TSIF1	Unit Channel 1 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
7 to 5	—	Reserved These bits are read as the value after reset.
4	THIF0	Unit Channel 0 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
3	CFTIF0	Unit Channel 0 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
2	TQIF0	Unit Channel 0 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
1	TAIF0	Unit Channel 0 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
0	TSIF0	Unit Channel 0 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.

TSIFm Bits

The TSIFm bit is set to 1 when the TMIEp bit in the RSCFDnCFDTMIECy register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the corresponding RSCFDnCFDTMSTSp register are set to 10_B (transmit completed without abort request) or 11_B (transmit completed with abort request).

When the TMTRF[1:0] flags are cleared to 00_B under the condition that the TSIFm bit can be set to 1, this flag is cleared to 0. In addition, clearing the TMIEp bit to 0 also clears this flag to 0.

TAIFm Bits

The TAIFm bit is set to 1 when the TAIE bit in the RSCFDnCFDCmCTR register is 1 (transmit abort interrupt enabled) and the TMTRF[1:0] flags in the RSCFDnCFDTMSTSp register are set to 01_B (transmit abort completed).

This flag is cleared to 0 when the TMTRF[1:0] flags are cleared to 00_B after the transmit abort is completed.

TQIFm Bits

When the TXQIE bit in the RSCFDnCFDTXQCCm register is set to 1 (transmit queue interrupt enabled) and the TXQIF bit in the RSCFDnCFDTXQSTSm register is set to 1 (transmit queue interrupt request), the TQIFm bit is set to 1.

When the TXQIF bit (transmit queue interrupt request) in the RSCFDnCFDTXQSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the TXQIE bit is cleared to 0.

CFTIFm Bits

When the CFTXIE bit in the RSCFDnCFDCFCCK register is set to 1 (transmit/receive FIFO transmit interrupt enabled) and the CFTXIF bit in the RSCFDnCFDCFSTSk register is set to 1 (transmit/receive FIFO transmit interrupt request), the CFTIFm bit is set to 1.

When the CFTXIF bit is cleared to 0 under the conditions that the CFTIFm bit can be set to 1, this bit is cleared to 0. This flag is also cleared to 0 when the CFTXIE bit is cleared to 0.

THIFm Bits

When the THLIE bit in the RSCFDnCFDTHLCCm register is set to 1 (transmit history interrupt enabled) and the THLIF bit in the RSCFDnCFDTHLSTSm register is set to 1 (transmit history interrupt request), the THIFm bit is set to 1.

When the THLIF bit in the RSCFDnCFDTHLSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the THLIE bit is cleared to 0.

20.4.4.7 RSCFDnCFDGTINTSTS1 — Global TX Interrupt Status Register 1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	THIF4	CFTIF4	TQIF4	TAIF4	TSIF4
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R*	R*	R*	R*	R*

Note 1. This bit is automatically cleared in the global reset or channel reset mode.

Table 20.117 RSCFDnCFDGTINTSTS1 Register Contents

Bit Position	Bit Name	Function
31 to 5	—	Reserved These bits are read as the value after reset. When writing to these bits, write the value after reset.
4	THIF4	Unit Channel 4 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
3	CFTIF4	Unit Channel 4 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
2	TQIF4	Unit Channel 4 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
1	TAIF4	Unit Channel 4 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
0	TSIF4	Unit Channel 4 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.

TSIFm Bits

The TSIFm bit is set to 1 when the TMIEp bit in the RSCFDnCFDTMIECy register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the RSCFDnCFDTMSTSp register are set to 10_B (transmit completed without abort request) or 11_B (transmit completed with abort request).

When the TMTRF[1:0] flags are cleared to 00_B under the condition that the TSIFm bit can be set to 1, this flag is cleared to 0. In addition, clearing the TMIEp bit to 0 also clears this flag to 0.

TAIFm Bits

The TAIFm bit is set to 1 when the TAIE bit in the RSCFDnCFDCmCTR register is 1 (transmit abort interrupt enabled) and the TMTRF[1:0] flags in the RSCFDnCFDTMSTSp register are set to 01_B (transmit abort completed).

This flag is cleared to 0 when the TMTRF[1:0] flags are cleared to 00_B after the transmit abort is completed.

TQIFm Bits

When the TXQIE bit in the RSCFDnCFDTXQCCm register is set to 1 (transmit queue interrupt enabled) and the TXQIF bit in the RSCFDnCFDTXQSTSm register is set to 1 (transmit queue interrupt request), the TQIFm bit is set to 1.

When the TXQIF bit (transmit queue interrupt request) in the RSCFDnCFDTXQSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the TXQIE bit is cleared to 0.

CFTIFm Bits

When the CFTXIE bit in the RSCFDnCFDCFCCK register is set to 1 (transmit/receive FIFO transmit interrupt enabled) and the CFTXIF bit in the RSCFDnCFDCFSTSk register is set to 1 (transmit/receive FIFO transmit interrupt request), the CFTIFm bit is set to 1.

When the CFTXIF bit is cleared to 0 under the conditions that the CFTIFm bit can be set to 1, this bit is cleared to 0. This flag is also cleared to 0 when the CFTXIE bit is cleared to 0.

THIFm Bits

When the THLIE bit in the RSCFDnCFDTHLCCm register is set to 1 (transmit history interrupt enabled) and the THLIF bit in the RSCFDnCFDTHLSTSm register is set to 1 (transmit history interrupt request), the THIFm bit is set to 1.

When the THLIF bit in the RSCFDnCFDTHLSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the THLIE bit is cleared to 0.

20.4.4.8 RSCFDnCFDGFDCFG — Global FD Configuration Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TSCCFG[1:0]		—	—	—	—	—	—	—	RPED
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 20.118 RSCFDnCFDGFDCFG Register Contents

Bit Position	Bit Name	Function
31 to 10	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
9 to 8	TSCCFG[1:0]	Timestamp capture setting b9 b8 0 0: Timestamp capture at the sample point of SOF(start of frame) 0 1: Timestamp capture at frame valid indication 1 0: Timestamp capture at the sample point of RES bit*1 1 1: Setting prohibited
7 to 1	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	RPED	Protocol exception event detection disable bit 0: Protocol exception event detection is enabled 1: Protocol exception event detection is disabled

Note 1. When a Classical CAN frame is transmitted/received, a time-stamp value will be captured at the sample point in the SOF bit.

TSCCFG Bit

Select a point where a time-stamp value is captured. Modify this bit only in global reset mode.

RPED Bit

Setting this bit to 1 disables the protocol exception event detection. When a protocol exception event is detected while this bit is set to 1, the event is regarded as a form error and an error frame will be output.

Modify this bit only in global reset mode.

20.4.5 Details of Receive Rule-Related Registers

20.4.5.1 RSCFDnCFDGAFLECTR — Receive Rule Entry Control Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	AFLDA E	—	—	—	AFLPN[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 20.119 RSCFDnCFDGAFLECTR Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
8	AFLDAE	Receive Rule Table Write Enable 0: Receive rule table write is disabled. 1: Receive rule table write is enabled.
7 to 5	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
4 to 0	AFLPN[4:0]	Receive Rule Table Page Number Configuration A page number can be selected from a range of page 0 (00000 _B) to page 19 (10011 _B).

AFLDAE Bit

Setting this bit to 0 disables the write to the receive rule table. After writes to the receive rule table are completed, set this bit to 0 to disable the write to the table. The receive rule table can be read regardless of the value of this bit.

Set the AFLDAE bit to 1 only in global reset mode.

AFLPN[4:0] Bits

These bits are used to set the page number of the receive rule table. Sixteen receive rules can be set per page.

Set these bits to a value within a range of 00000_B to 10011_B.

For details about the receive rule table, see **Section 20.7.1, Data Processing Using the Receive Rule Table**.

20.4.5.2 RSCFDnCFDGAFLCFG0 — Receive Rule Configuration Register 0

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RNC0[7:0]								RNC1[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RNC2[7:0]								RNC3[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.120 RSCFDnCFDGAFLCFG0 Register Contents

Bit Position	Bit Name	Function
31 to 24	RNC0[7:0]	Number of Rules for Unit Channel 0 Set the number of receive rules exclusively used for channel 0.
23 to 16	RNC1[7:0]	Number of Rules for Unit Channel 1 Set the number of receive rules exclusively used for channel 1.
15 to 8	RNC2[7:0]	Number of Rules for Unit Channel 2 Set the number of receive rules exclusively used for channel 2.
7 to 0	RNC3[7:0]	Number of Rules for Unit Channel 3 Set the number of receive rules exclusively used for channel 3.

Modify the RSCFDnCFDGAFLCFG0 register only in global reset mode.

Up to $64 \times$ (number of channels) rules can be registered in the receive rule table for the entire unit. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 128.
- The total number of rules allocated to each channel does not exceed the number of rules that can be registered for the entire unit.

RNC0[7:0] Bits

These bits are used to set the number of rules to be registered in the unit channel 0 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC1[7:0] Bits

These bits are used to set the number of rules to be registered in the unit channel 1 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC2[7:0] Bits

These bits are used to set the number of rules to be registered in the unit channel 2 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC3[7:0] Bits

These bits are used to set the number of rules to be registered in the unit channel 3 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

20.4.5.3 RSCFDnCFDGAFLCFG1 — Receive Rule Configuration Register 1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RNC4[7:0]								—							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.121 RSCFDnCFDGAFLCFG1 Register Contents

Bit Position	Bit Name	Function
31 to 24	RNC4[7:0]	Number of Rules for Unit Channel 4 Set the number of receive rules exclusively used for channel 4.
23 to 0	—	Reserved These bits are always read as 0. The write value should always be 0.

Modify the RSCFDnGAFLCFG1 register only in global reset mode.

Up to “64 × (number of channels)” rules can be registered to the receive rule tables in the entire module. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 128.
- The total of the number of rules allocated to each channel is not larger than the number of rules permitted to the entire module.

RNC4[7:0] Bits

These bits are used to set the number of rules to be registered in the unit channel 4 receive rule table.

Set these bits to a value within a range of 00_H to 80_H.

20.4.5.4 RSCFDnCFDGAFLIDj — Receive Rule ID Register j

Note: For the value of index "j", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLID E	GAFLR TR	GAFL LB	GAFLID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.122 RSCFDnCFDGAFLIDj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDE	IDE Select 0: Standard ID 1: Extended ID
30	GAFLRTR	RTR Select 0: Data frame 1: Remote frame
29	GAFLLB	Receive Rule Target Message Select 0: When a message transmitted from another CAN node is received 1: When the own transmitted message is received
28 to 0	GAFLID[28:0]	ID Set the ID of the receive rule. For the standard ID, set the ID in bits b10 to b0 and set bits b28 to b11 to 0.

Modify the RSCFDnCFDGAFLIDj register when the AFLDAE bit in the RSCFDnCFDGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDE Bit

This bit is used to select the ID format (standard ID or extended ID) of the receive rule. This bit is compared with the IDE bit in the received message during the acceptance filter processing.

GAFLRTR Bit

This bit is used to select the frame format (data frame or remote frame) of the receive rule. This bit is compared with the RTR bit in the received message during the acceptance filter processing.

GAFLLB Bit

When this bit is set to 0, data processing using the receive rule is performed when receiving messages transmitted from another CAN node.

When this bit is set to 1 when the mirror function is used, data processing using the receive rule is performed when the CAN node is receiving its own transmitted messages.

GAFLID[28:0] Bits

These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID of the received message during the acceptance filter processing.

20.4.5.5 RSCFDnCFDGAFLMj — Receive Rule Mask Register j

Note: For the value of index "j", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLID EM	GAFLR TRM	—	GAFLIDM[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLIDM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.123 RSCFDnCFDGAFLMj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDEM	IDE Mask 0: The IDE bit is not compared. 1: The IDE bit is compared.
30	GAFLRTRM	RTR Mask 0: The RTR bit is not compared. 1: The RTR bit is compared.
29	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
28 to 0	GAFLIDM[28:0]	ID Mask 0: The corresponding ID bit is not compared. 1: The corresponding ID bit is compared.

Modify the RSCFDnCFDGAFLMj register when the AFLDAE bit in the RSCFDnCFDGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDEM Bit

When this bit is set to 1, filter processing is performed only for messages of the ID format specified by the GAFLIDE bit in the RSCFDnCFDGAFLIDj register.

When this bit is cleared to 0, the IDs of all the receive messages and the specified IDs are regarded as matched. To set the GAFLIDEM bit to 0, set all the GAFLIDM[28:0] bits to 0 at the same time.

GAFLRTRM Bit

This bit is used to mask the RTR bit of the receive rule.

GAFLIDM[28:0] Bits

These bits are used to mask the corresponding ID bit of the receive rule.

20.4.5.6 RSCFDnCFDGAFLP0_j — Receive Rule Pointer 0 Register j

Note: For the value of index "j", see **Table 20.6**.

Value after reset: 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLDLC[3:0]				GAFLPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLRMV	GAFLRMDP[6:0]						—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 20.124 RSCFDnCFDGAFLP0_j Register Contents

Bit Position	Bit Name	Function
31 to 28	GAFLDLC[3:0]	Receive Rule DLC
		b31 b30 b29 b28 Classical CAN Frame CAN FD Frame
		0 0 0 0 DLC check is disabled
		0 0 0 1 1 data bytes
		0 0 1 0 2 data bytes
		0 0 1 1 3 data bytes
		0 1 0 0 4 data bytes
		0 1 0 1 5 data bytes
		0 1 1 0 6 data bytes
		0 1 1 1 7 data bytes
		1 0 0 0 8 data bytes
		1 0 0 1 8 data bytes 12 data bytes
		1 0 1 0 16 data bytes
		1 0 1 1 20 data bytes
		1 1 0 0 24 data bytes
		1 1 0 1 32 data bytes
1 1 1 0 48 data bytes		
1 1 1 1 64 data bytes		
27 to 16	GAFLPTR[11:0]	Receive Rule Label Set the 12-bit label information.
15	GAFLRMV	Receive Buffer Enable 0: No receive buffer is used. 1: A receive buffer is used.
14 to 8	GAFLRMDP[6:0]	Receive Buffer Number Select Set the receive buffer number to store receive messages.
7 to 0	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.

Modify the RSCFDnCFDGAFLP0_j register when the AFLDAE bit in the RSCFDnCFDGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLDLC[3:0] Bits

These bits are used to set the minimum data length necessary for receiving messages. If the data length of a message that is being filtered is equal to or larger than the value set by the GAFLDLC[3:0] bits, the message passes the DLC check. Setting these bits to 0000_B disables the DLC check function allowing messages with any data length to pass the DLC check.

GAFLPTR[11:0] Bits

These bits are used to set a 12-bit label to be attached to messages that have passed through the filter. A label is attached when a message is stored in the receive buffer or the FIFO buffer.

GAFLRMV Bit

When this bit is set to 1, receive messages that have passed through the filter are stored in the receive buffer selected by the GAFLRMDP[6:0] bits.

GAFLRMDP[6:0] Bits

These bits are used to select the number of the receive buffer that stores receive messages that have passed through the filter when the GAFLRMV bit is set to 1. Set these bits to a value smaller than the value set by the NRXMB[7:0] bits in the RSCFDnCFDRMNB register.

20.4.5.7 RSCFDnCFDGAFLP1_j — Receive Rule Pointer 1 Register j

Note: For the value of index "j", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	GAFLFDP[22:16]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLFDP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.125 RSCFDnCFDGAFLP1_j Register Contents

Bit Position	Bit Name	Function
31 to 23	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
22 to 8	GAFLFDP[22:8]	Transmit/Receive FIFO Buffer k Select (Bit position – 8 = target transmit/receive FIFO buffer number k) 0: Transmit/receive FIFO buffer is not selected. 1: Transmit/receive FIFO buffer is selected.
7 to 0	GAFLFDP[7:0]	Receive FIFO Buffer x Select (Bit position = target receive FIFO buffer number x) 0: Receive FIFO buffer is not selected. 1: Receive FIFO buffer is selected.

Modify the RSCFDnCFDGAFLP1_j register when the AFLDAE bit in the RSCFDnCFDGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLFDP Bits

These bits are used to specify FIFO buffers that store receive messages that have passed through the filter. Up to eight FIFO buffers can be selected. However, when the GAFLRMV bit in the RSCFDnCFDGAFLP0_j register is set to 1 (messages are stored in the receive buffer), up to seven FIFO buffers can be selected. Only receive FIFO buffers and the transmit/receive FIFO buffer for which the CFM[1:0] bits in the RSCFDnCFDCFCCK register are set to 00_B (receive mode) or 10_B (gateway mode) can be selected.

20.4.6 Details of Receive Buffer-Related Registers

20.4.6.1 RSCFDnCFDRMNB — Receive Buffer Number Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RMPLS[1:0]		NRXMB[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.126 RSCFDnCFDRMNB Register Contents

Bit Position	Bit Name	Function
31 to 10	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
9, 8	RMPLS[1:0]	Receive Buffer Payload Storage Size Select b9 b8 0 0: 8 bytes 0 1: 12 bytes 1 0: 16 bytes 1 1: 20 bytes
7 to 0	NRXMB[7:0]	Receive Buffer Number Configuration Set the number of receive buffers. Set a value of 0 to 80.

Modify the RSCFDnCFDRMNB register only in global reset mode.

RMPLS[1:0] Bits

These bits are used to select the maximum payload size that can be stored in the receive buffer.

NRXMB[7:0] Bits

These bits are used to set the total number of receive buffers of the RS-CANFD module. The maximum value is $16 \times$ (number of channels).

Setting all of these bits to 0 makes receive buffers unavailable.

20.4.6.2 RSCFDnCFDRMNDy — Receive Buffer New Data Register y

Note: For the value of index "y", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMNSq (q = y × 32 + 31)	RMNSq (q = y × 32 + 30)	RMNSq (q = y × 32 + 29)	RMNSq (q = y × 32 + 28)	RMNSq (q = y × 32 + 27)	RMNSq (q = y × 32 + 26)	RMNSq (q = y × 32 + 25)	RMNSq (q = y × 32 + 24)	RMNSq (q = y × 32 + 23)	RMNSq (q = y × 32 + 22)	RMNSq (q = y × 32 + 21)	RMNSq (q = y × 32 + 20)	RMNSq (q = y × 32 + 19)	RMNSq (q = y × 32 + 18)	RMNSq (q = y × 32 + 17)	RMNSq (q = y × 32 + 16)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMNSq (q = y × 32 + 15)	RMNSq (q = y × 32 + 14)	RMNSq (q = y × 32 + 13)	RMNSq (q = y × 32 + 12)	RMNSq (q = y × 32 + 11)	RMNSq (q = y × 32 + 10)	RMNSq (q = y × 32 + 9)	RMNSq (q = y × 32 + 8)	RMNSq (q = y × 32 + 7)	RMNSq (q = y × 32 + 6)	RMNSq (q = y × 32 + 5)	RMNSq (q = y × 32 + 4)	RMNSq (q = y × 32 + 3)	RMNSq (q = y × 32 + 2)	RMNSq (q = y × 32 + 1)	RMNSq (q = y × 32 + 0)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.127 RSCFDnCFDRMNDy Register Contents

Bit Position	Bit Name	Function
31 to 0	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 31 to y × 32 + 0) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q.

Write 0 to the RSCFDnCFDRMNDy register in global operating mode or global test mode.

RMNSq Flags (q = see Table 20.6)

Each RMNSq flag is set to 1 when the processing for storing a message in the corresponding receive buffer starts.

To clear a flag to 0, the program must write 0 to the flag. Use a store instruction to write "0" to the flag and "1" to other flags. These bits cannot be set to 0 while a message is being stored. The message storing time depends on the storage payload size of the receive buffer. When the RMPLS[1:0] value in the RSCFDnCFDRMNB register is 00_B (8 bytes), the message storing time is 12 pclk clock cycles. When the RMPLS[1:0] value is 11_B (20 bytes), the message storing time is 18 pclk clock cycles. (2 pclk clock cycles per 4 bytes of storage payload size).

These flags are cleared to 0 in global reset mode.

20.4.6.3 RSCFDnCFDRMIDq — Receive Buffer ID Register q

Note: For the value of index "q", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMIDE	RMRTR	—	RMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.128 RSCFDnCFDRMIDq Register Contents

Bit Position	Bit Name	Function
31	RMIDE	Receive Buffer IDE 0: Standard ID 1: Extended ID
30	RMRTR	Receive Buffer RTR/RRS <ul style="list-style-type: none"> When the received message is a classical CAN frame 0: Data frame 1: Remote frame When the received message is a CAN FD frame The RRS bit value of the received message can be read.
29	—	Reserved These bits are read as the value after reset.
28 to 0	RMID[28:0]	Receive Buffer ID Data These bits contain the standard ID or extended ID of the received message. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

RMIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive buffer.

RMRTR Bit

When the received message is a classical CAN frame, this bit indicates the frame format (data frame or remote frame) of the message stored in the receive buffer. When the received message is a CAN FD frame, this bit indicates the RRS bit value in the message.

RMID[28:0] Bits

These bits contain the ID of the message stored in the receive buffer.

20.4.6.4 RSCFDnCFDRMPTRq — Receive Buffer Pointer Register q

Note: For the value of index "q", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDLC[3:0]				RMPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.129 RSCFDnCFDRMPTRq Register Contents

Bit Position	Bit Name	Function	
31 to 28	RMDLC[3:0]	Receive Buffer DLC Data	
		b31 b30 b29 b28 Classical CAN Frame CAN FD Frame	
		0 0 0 0 0 data bytes	
		0 0 0 1 1 data bytes	
		0 0 1 0 2 data bytes	
		0 0 1 1 3 data bytes	
		0 1 0 0 4 data bytes	
		0 1 0 1 5 data bytes	
		0 1 1 0 6 data bytes	
		0 1 1 1 7 data bytes	
		1 0 0 0 8 data bytes	
		1 0 0 1 8 data bytes	12 data bytes
		1 0 1 0	16 data bytes
		1 0 1 1	20 data bytes
		1 1 0 0	24 data bytes
		1 1 0 1	32 data bytes
1 1 1 0	48 data bytes		
1 1 1 1	64 data bytes		
27 to 16	RMPTR[11:0]	Receive Buffer Label Data Label information of the received message.	
15 to 0	RMTS[15:0]	Receive Buffer Timestamp Data Timestamp value of the received message.	

RMDLC[3:0] Bits

These bits indicate the data length of the message stored in the receive buffer. The number of bytes of the payload to be stored in the receive buffer is determined by the RMPLS[1:0] bits in the RSCFDnCFDRMNB register.

RMPTR[11:0] Bits

These bits indicate the label information of the message stored in the receive buffer.

RMTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the receive buffer.

20.4.6.5 RSCFDnCFDRMFDSTSq — Receive Buffer CAN FD Status Register q

Note: For the value of index "q", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RMFDF	RMBRS	RMESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.130 RSCFDnCFDRMFDSTSq Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved These bits are read as the value after reset.
2	RMFDF	FDF 0: Classical CAN frame 1: CAN FD frame
1	RMBRS	BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes.
0	RMESI	ESI 0: Error active node 1: Error passive node

RMFDF Bit

This bit indicates the FD format (classical CAN frame or CAN FD frame) of the message stored in the receive buffer.

RMBRS Bit

When the RMFDF bit is set to 1, this bit indicates the BRS bit value of the message stored in the receive buffer. When the RMFDF bit is set to 0, this bit is always read as 0.

RMESI Bit

When the RMFDF bit is set to 1, this bit indicates the ESI bit value of the message stored in the receive buffer. When the RMFDF bit is set to 0, this bit is always read as 0.

20.4.6.6 RSCFDnCFDRMDFb_q — Receive Buffer Data Field b Register q

Note: For the value of index "b" and "q", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB4 × b + 3 [7:0]								RMDB4 × b + 2 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB4 × b + 1 [7:0]								RMDB4 × b + 0 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.131 RSCFDnCFDRMDFb_q Register Contents

Bit Position	Bit Name	Function
31 to 24	RMDB4 × b + 3 [7:0]	Receive Buffer Data Byte 4 × b + 3
23 to 16	RMDB4 × b + 2 [7:0]	Receive Buffer Data Byte 4 × b + 2
15 to 8	RMDB4 × b + 1 [7:0]	Receive Buffer Data Byte 4 × b + 1
7 to 0	RMDB4 × b + 0 [7:0]	Receive Buffer Data Byte 4 × b + 0

Data for a message stored in the receive buffer can be read.

When the RMDLC[3:0] value in the RSCFDnCFDRMPTRq register is smaller than the payload storage size of the receive buffer, data bytes for which no data is set are read as 00_H.

Specify the payload storage size of the receive buffer by the RMPLS[1:0] bits in the RSCFDnCFDRMNB register. Do not read or write the RSCFDnCFDRMDFb_q register corresponding to an area that exceeds the specified size.

20.4.7 Details of Receive FIFO Buffer-Related Registers

20.4.7.1 RSCFDnCFDRFCCx — Receive FIFO Buffer Configuration and Control Register x

Note: For the value of index "x", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	RFPLS[2:0]			—	—	RFIE	RFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

Table 20.132 RSCFDnCFDRFCCx Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
15 to 13	RFIGCV[2:0]	Receive FIFO Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.
12	RFIM	Receive FIFO Interrupt Source Select 0: An interrupt occurs when the condition set by the RFIGCV[2:0] bits is met. 1: An interrupt occurs each time a message has been received.
11	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
10 to 8	RFDC[2:0]	Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 20.132 RSCFDnCFDRFCCx Register Contents (2/2)

Bit Position	Bit Name	Function
6 to 4	RFPLS[2:0]	Receive FIFO Buffer Payload Storage Size Select b6 b5 b4 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes
3, 2	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
1	RFIE	Receive FIFO Interrupt Enable 0: Receive FIFO interrupt is disabled. 1: Receive FIFO interrupt is enabled.
0	RFE	Receive FIFO Buffer Enable 0: No receive FIFO buffer is used. 1: Receive FIFO buffers are used.

RFIGCV[2:0] Bits

These bits are used to specify the number of received messages required to generate a receive FIFO interrupt request when the RFIM bit is set to 0 with a fraction for the total number of buffers (the setting of RFDC[2:0]).

When the RFDC[2:0] bits are set to 001_B (4 messages), set the RFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B. Modify these bits only in global reset mode.

RFIM Bit

This bit is used to select a FIFO interrupt source. Modify this bit only in global reset mode.

RFDC[2:0] Bits

These bits are used to select the number of messages that can be stored in a single receive FIFO buffer. When these bits are set to 000_B, no receive FIFO buffer should be used. Modify these bits only in global reset mode.

RFPLS[2:0] Bits

These bits are used to select the maximum payload size that can be stored in the receive FIFO buffer. Modify these bits only in global reset mode.

RFIE Bit

Setting the RFIE bit to 1 enables receive FIFO interrupts. Modify this bit when the RFE bit is set to 0 (no receive FIFO buffer is used).

RFE Bit

Setting the RFE bit to 1 enables the use of FIFO buffers. Clearing this bit to 0 sets the RFEMP flag in the RSCFDnCFDRFSTSx register to 1 (buffer empty). Modify this bit in global operating mode or global test mode.

Set this bit to 1 with another instruction after the settings to all bits in the RSCFDnCFDRFCCx register have been done. This bit is cleared to 0 in global reset mode.

20.4.7.2 RSCFDnCFDRFSTSx — Receive FIFO Buffer Status Register x

Note: For the value of index "x", see **Table 20.6**.

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMC[7:0]							—	—	—	—	RFIF	RFMLT	RFFLL	RFEMP	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 20.133 RSCFDnCFDRFSTSx Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved These bits are read as the value after reset. When writing to these bits, write the value after reset.
15 to 8	RFMC[7:0]	Receive FIFO Unread Message Counter The number of unread messages stored in the receive FIFO buffer is displayed.
7 to 4	—	Reserved These bits are read as the value after reset. When writing to these bits, write the value after reset.
3	RFIF	Receive FIFO Interrupt Request Flag 0: No receive FIFO interrupt request is present. 1: A receive FIFO interrupt request is present.
2	RFMLT	Receive FIFO Message Lost Flag 0: No receive FIFO message is lost. 1: A receive FIFO message is lost.
1	RFFLL	Receive FIFO Buffer Full Status Flag 0: The receive FIFO buffer is not full. 1: The receive FIFO buffer is full.
0	RFEMP	Receive FIFO Buffer Empty Status Flag 0: The receive FIFO buffer contains unread message. 1: The receive FIFO buffer contains no unread message (buffer empty).

RFMC[7:0] Flag

This flag indicates the number of unread messages in the receive FIFO buffer. This flag becomes 00_H when the RFE bit in the RSCFDnCFDRFCCx register is set to 0.

This flag is 00_H in global reset mode.

RFIF Flag

This flag is set to 1 when the receive FIFO interrupt request generation conditions set by the RFIGCV[2:0] bits and the RFIM bit in the RSCFDnCFDRFCCx register are met. This flag is cleared to 0 in global reset mode or by writing 0 to this flag. Modify this bit in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flags to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFMLT Flag

This flag is set to 1 when an attempt is made to store a new message while the receive FIFO buffer is full. In this case, the new message is discarded.

This flag is cleared to 0 in global reset mode or by writing 0 to this flag.

Modify this bit in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flags to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFLL Flag

This flag is set to 1 when the number of messages stored in the receive FIFO buffer matches the FIFO buffer depth set by the RFDC[2:0] bits in the RSCFDnCFDRFCCx register.

If the number of messages stored in the receive FIFO buffer becomes smaller than the FIFO buffer depth set by the RFDC[2:0] bits, this flag is cleared to 0. This flag is also cleared to 0 when the RFE bit in the RSCFDnCFDRFCCx register is set to 0 (no receive FIFO buffer is used) or in global reset mode.

RFEMP Flag

This flag is set to 1 when all messages in the receive FIFO buffer have been read. This flag is also set to 1 when the RFE bit in the RSCFDnCFDRFCCx register is 0 or in global reset mode.

This flag is cleared to 0 when a received message is in the receive FIFO buffer.

NOTE

To clear the RFMLT or RFIF flag to 0, use a store instruction to write “0” to the given flag and “1” to the other flags.

20.4.7.3 RSCFDnCFDRFPCTR_x — Receive FIFO Buffer Pointer Control Register x

Note: For the value of index "x", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 20.134 RSCFDnCFDRFPCTR_x Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved The write value should be the value after reset.
7 to 0	RFPC[7:0]	Receive FIFO Pointer Control When these bits are set to FF _H , the read pointer moves to the next unread message in the receive FIFO buffer.

When the RFDMAE_x value in the RSCFDnCFDCDTCT register is 1 (DMA transfer request enabled), do not write a value to this register.

RFPC[7:0] Bits

When the RFPC[7:0] bits are set to FF_H, the read pointer moves to the next unread message in the receive FIFO buffer. At this time, the RFMC[7:0] (receive FIFO unread message counter) value in the RSCFDnCFDRFSTS_x register is decremented by 1. Read the RSCFDnCFDRFID_x, RSCFDnCFDRFPTR_x, RSCFDnCFDRFFDSTS_x, and RSCFDnCFDRFDFd_x registers to read messages in the receive FIFO buffer, and then write FF_H to the RFPC[7:0] bits.

When writing FF_H to these bits, make sure that the RFE bit in the RSCFDnCFDRFCC_x register is set to 1 (receive FIFO buffers are used) and the RFEMP flag in the RSCFDnCFDRFSTS_x register is 0 (the receive FIFO buffer contains unread messages).

20.4.7.4 RSCFDnCFDRFIDx — Receive FIFO Buffer Access ID Register x

Note: For the value of index "x", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFIDE	RFRTTR	—	RFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.135 RSCFDnCFDRFIDx Register Contents

Bit Position	Bit Name	Function
31	RFIDE	Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	RFRTTR	Receive FIFO Buffer RTR/RRS <ul style="list-style-type: none"> When the received message is a classical CAN frame 0: Data frame 1: Remote frame When the received message is a CAN FD frame The RRS bit value of the received message can be read.
29	—	Reserved These bits are read as the value after reset.
28 to 0	RFID[28:0]	Receive FIFO Buffer ID Data The standard ID or extended ID of received message can be read. Read bits 10 to 0 for standard ID. Bits 28 to 11 are read as 0.

RFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive FIFO buffer.

RFRTTR Bit

When the received message is a classical CAN frame, this bit indicates the frame format (data frame or remote frame) of the message stored in the receive FIFO buffer. When the received message is a CAN FD frame, this bit indicates the RRS bit value in the message.

RFID[28:0] Bits

These bits indicate the ID of the message stored in the receive FIFO buffer.

20.4.7.5 RSCFDnCFDRFPTRx — Receive FIFO Buffer Access Pointer Register x

Note: For the value of index "x", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDLC[3:0]				RFPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.136 RSCFDnCFDRFPTRx Register Contents

Bit Position	Bit Name	Function	
31 to 28	RFDLC[3:0]	Receive FIFO Buffer DLC Data	
		b31 b30 b29 b28 Classical CAN Frame CAN FD Frame	
		0 0 0 0 0 data bytes	
		0 0 0 1 1 data bytes	
		0 0 1 0 2 data bytes	
		0 0 1 1 3 data bytes	
		0 1 0 0 4 data bytes	
		0 1 0 1 5 data bytes	
		0 1 1 0 6 data bytes	
		0 1 1 1 7 data bytes	
		1 0 0 0 8 data bytes	
		1 0 0 1 8 data bytes	12 data bytes
		1 0 1 0	16 data bytes
		1 0 1 1	20 data bytes
		1 1 0 0	24 data bytes
		1 1 0 1	32 data bytes
1 1 1 0	48 data bytes		
1 1 1 1	64 data bytes		
27 to 16	RFPTR[11:0]	Receive FIFO Buffer Label Data Label information of the received message can be read.	
15 to 0	RFTS[15:0]	Receive FIFO Buffer Timestamp Data Timestamp value of the received message can be read.	

RFDLC[3:0] Bits

These bits contain the data length of the message stored in the receive FIFO buffer.

RFPTR[11:0] Bits

These bits contain the label information of the message stored in the receive FIFO buffer.

RFTS[15:0] Bits

These bits contain the timestamp value of the message stored in the receive FIFO buffer.

20.4.7.6 RSCFDnCFDRFFDSTSx — Receive FIFO CAN FD Status Register x

Note: For the value of index "x", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFDF	RFBRS	RFESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.137 RSCFDnCFDRFFDSTSx Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved These bits are read as the value after reset.
2	RFFDF	FDF 0: Classical CAN frame 1: CAN FD frame
1	RFBRS	BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes.
0	RFESI	ESI 0: Error active node 1: Error passive node

RFFDF Bit

This bit indicates the FD format (classical CAN frame or CAN FD frame) of the message stored in the receive FIFO buffer.

RFBRS Bit

When the RFFDF bit is set to 1, this bit indicates the BRS bit value of the message stored in the receive FIFO buffer. When the RFFDF bit is set to 0, this bit is always read as 0.

RFESI Bit

When the RFFDF bit is set to 1, this bit indicates the ESI bit value of the message stored in the receive FIFO buffer. When the RFFDF bit is set to 0, this bit is always read as 0.

20.4.7.7 RSCFDnCFDRFDFd_x — Receive FIFO Buffer Access Data Field d Register x

Note: For the value of index "d" and "x", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB4 × d + 3 [7:0]								RFDB4 × d + 2 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFDB4 × d + 1 [7:0]								RFDB4 × d + 0 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.138 RSCFDnCFDRFDFd_x Register Contents

Bit Position	Bit Name	Function
31 to 24	RFDB4 × d + 3 [7:0]	Receive Buffer Data Byte 4 × d + 3
23 to 16	RFDB4 × d + 2 [7:0]	Receive Buffer Data Byte 4 × d + 2
15 to 8	RFDB4 × d + 1 [7:0]	Receive Buffer Data Byte 4 × d + 1
7 to 0	RFDB4 × d + 0 [7:0]	Receive Buffer Data Byte 4 × d + 0

Data for a message stored in the receive buffer can be read.

When the RFDLC[3:0] value in the RSCFDnCFDRFPTRx register is smaller than the payload storage size of the receive FIFO buffer, data bytes for which no data is set are read as 00_H.

Specify the payload storage size of the receive FIFO buffer by the RFPLS[2:0] bits in the RSCFDnCFDRFCCx register. Do not read or write the RSCFDnCFDRFDFd_x register corresponding to an area that exceeds the specified size.

20.4.8 Details of Transmit/Receive FIFO Buffer Related Registers

20.4.8.1 RSCFDnCFDCFCCK — Transmit/Receive FIFO Buffer Configuration and Control Register k

Note: For the value of index "k", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFITT[7:0]							CFTML[3:0]				CFITR	CFITSS	CFM[1:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFIGCV[2:0]			CFIM	—	CFDC[2:0]		—	CFPLS[2:0]			—	CFIXIE	CFRXIE	CFE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 20.139 RSCFDnCFDCFCCK Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	CFITT[7:0]	Set a message transmission interval. Set Value: 00 _H to FF _H
23 to 20	CFTML[3:0]	Transmit Buffer Link Configuration Set the transmit buffer number to be linked to the transmit/receive FIFO buffer.
19	CFITR	Transmit/Receive FIFO Interval Timer Resolution 0: Clock obtained by dividing pclk/2 by the value of the ITRCP [15:0] bits 1: Clock obtained by dividing pclk/2 by the "value of ITRCP [15:0] bits x 10"
18	CFITSS	Transmit/Receive FIFO Interval Timer Clock Source Select 0: Interval timer clock source selected by the CFITR bit 1: Interval timer clock source is the nominal bit time clock for the channel to which the FIFO is linked.
17, 16	CFM[1:0]	Transmit/Receive FIFO Mode Select b17 b16 0 0: Receive mode 0 1: Transmit mode 1 0: Gateway mode 1 1: Setting prohibited
15 to 13	CFIGCV[2:0]	Transmit/Receive FIFO Receive Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.

Table 20.139 RSCFDnCFDCFCCK Register Contents (2/2)

Bit Position	Bit Name	Function
12	CFIM	Transmit/Receive FIFO Interrupt Source Select 0: <ul style="list-style-type: none"> • Receive mode/gateway mode When the number of received messages meets the condition set by the CFIGCV[2:0] bits, a FIFO receive interrupt request is generated. • Transmit mode/gateway mode When the buffer becomes empty upon completion of message transmission, a FIFO transmit interrupt request is generated. 1: <ul style="list-style-type: none"> • Receive mode/gateway mode A FIFO receive interrupt request is generated each time a message has been received. • Transmit mode/gateway mode A FIFO transmit interrupt request is generated each time a message has been transmitted.
11	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
10 to 8	CFDC[2:0]	Transmit/Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
7	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
6 to 4	CFPLS[2:0]	Transmit/Receive FIFO Buffer Payload Storage Size Select b6 b5 b4 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes
3	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
2	CFTXIE	Transmit/Receive FIFO Transmit Interrupt Enable 0: Transmit/receive FIFO transmit interrupt is disabled. 1: Transmit/receive FIFO transmit interrupt is enabled.
1	CFRXIE	Transmit/Receive FIFO Receive Interrupt Enable 0: Transmit/receive FIFO receive interrupt is disabled. 1: Transmit/receive FIFO receive interrupt is enabled.
0	CFE	Transmit/Receive FIFO Buffer Enable 0: No transmit/receive FIFO buffer is used. 1: Transmit/receive FIFO buffers are used.

CFITT[7:0] Bits

These bits are used to set a message transmission interval when transmitting messages continuously from a transmit/receive FIFO buffer whose CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode).

Clear the CFE bit to 0 (no transmit/receive FIFO buffer is used) before modifying the CFITT[7:0] bits.

CFTML[3:0] Bits

These bits are used to set the number of transmit buffer on the channel which will be linked to transmit/receive FIFO buffer k when the CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode). There are three transmit/receive FIFO buffers per channel, so channel number m of FIFO buffer k is calculated as $m = k/3$ (integral quotient). The actual assigned transmit buffer number p linked to FIFO buffer k will be $((16 \times m) + CFTML[3:0])$. See **Table 20.97**.

See **Table 20.95** and **Table 20.96**, as for the relationship between transmit/receive FIFO buffer k and transmit buffer p.

Setting the CFDC[2:0] bits to 001_B or more enables the setting of the CFTML[3:0] bits.

Do not link to any transmit buffer which is already allocated to a transmit queue on the same channel or to another transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFITR Bit

This bit is enabled when the CFITSS bit is 0.

When this bit is 0, the interval timer clock source is the pclk/2 clock divided by the value of the ITRCP[15:0] bits in the RSCFDnCFDGCFCFG register.

When this bit is 1, the interval timer clock source is the pclk/2 clock divided by (the value of the ITRCP[15:0] bits in the RSCFDnCFDGCFCFG register × 10).

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFITSS Bit

When this bit is 0, the clock selected by the CFITR bit is the count source of the interval timer.

When this bit is 1, the nominal bit time clock of the channel to which the FIFO is linked is the count source of the interval timer. Use this count source only for the channel does not handle the CAN FD frames.

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFM[1:0] Bits

These bits are used to select transmit/receive FIFO mode. Modify these bits only in global reset mode.

CFIGCV[2:0] Bits

These bits are used to specify the number of received messages required to generate a transmit/receive FIFO receive interrupt request when the CFM[1:0] bits are set to 00_B (receive mode) or 10_B (gateway mode) and the CFIM bit is set to 0 with a fraction for the total number of buffers (the setting of CFDC[2:0]).

When the CFDC[2:0] bits are set to 001_B (4 messages), set the CFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B.

Modify these bits only in global reset mode.

CFIM Bit

This bit is used to select a transmit/receive FIFO interrupt source. Modify this bit only in global reset mode.

CFDC[2:0] Bits

These bits are used to set the number of messages that can be stored in a single transmit/receive FIFO buffer. When these bits are set to 000_B, do not use a transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFPLS[2:0] Bits

These bits are used to select the maximum payload size that can be stored in the transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFTXIE Bit

When this bit is set to 1 and the CFTXIF flag in the RSCFDnCFDCFSTSk register is set to 1, a transmit/receive FIFO transmit interrupt request is generated.

Modify this bit with the CFE bit set to 0 (no transmit/receive FIFO buffer is used).

CFRXIE Bit

When this bit is set to 1 and the CFRXIF flag in the RSCFDnCFDCFSTSk register is set to 1, a transmit/receive FIFO receive interrupt request is generated.

Modify this bit with the CFE bit set to 0.

CFE Bit

Setting this bit to 1 enables the transmit/receive FIFO buffers.

When this bit is set to 0 in transmit mode or gateway mode, if a message in the transmit/receive FIFO buffer is being transmitted or will be transmitted next, the transmit/receive FIFO buffer becomes empty after completion of transmission of that message, or upon detection of a CAN bus error, or arbitration-lost. In other cases or in receive mode, the transmit/receive FIFO buffer becomes empty immediately.

This bit is cleared to 0 when the following conditions are met.

- Receive mode: Global reset mode
- Transmit mode or gateway mode: Channel reset mode

Modify this bit in the following mode.

- Receive mode: Global operating mode or global test mode
- Transmit mode or gateway mode: Channel communication mode or channel halt mode

After all other bits in the RSCFDnCFDCFCCk register have been set, set this bit to 1 by using another instruction.

20.4.8.2 RSCFDnCFDCFSTSk — Transmit/Receive FIFO Buffer Status Register k

Note: For the value of index "k", see **Table 20.6**.

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFMC[7:0]							—	—	—	CFTXIF	CFRXIF	CFMLT	CFLL	CFEMP	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 20.140 RSCFDnCFDCFSTSk Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved These bits are read as the value after reset. When writing to these bits, write the value after reset.
15 to 8	CFMC[7:0]	Transmit/Receive FIFO Message Counter The number of messages stored in the transmit/receive FIFO buffer.
7 to 5	—	Reserved These bits are read as the value after reset. When writing to these bits, write the value after reset.
4	CFTXIF	Transmit/Receive FIFO Transmit Interrupt Request Flag 0: No transmit/receive FIFO transmit interrupt request is present. 1: A transmit/receive FIFO transmit interrupt request is present.
3	CFRXIF	Transmit/Receive FIFO Receive Interrupt Request Flag 0: No transmit/receive FIFO receive interrupt request is present. 1: A transmit/receive FIFO receive interrupt request is present.
2	CFMLT	Transmit/Receive FIFO Message Lost Flag 0: No transmit/receive FIFO message is lost. 1: A transmit/receive FIFO message is lost.
1	CFLL	Transmit/Receive FIFO Buffer Full Status Flag 0: The transmit/receive FIFO buffer is not full. 1: The transmit/receive FIFO buffer is full.
0	CFEMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: The transmit/receive FIFO buffer contains messages. 1: The transmit/receive FIFO buffer contains no message (buffer empty).

CFMC[7:0] Bits

The CFMC[7:0] bits indicate the following values according to the setting of the CFM[1:0] bits in the RSCFDnCFDCFCCK register.

- When CFM[1:0] value is 01_B (transmit mode): Number of untransmitted messages in the buffer
- When CFM[1:0] value is 00_B (receive mode): Number of unread received messages in the buffer
- When CFM[1:0] value is 10_B (gateway mode): Number of untransmitted received messages in the buffer

These bits are cleared to 0 when any of the following conditions is met.

- When CFM[1:0] value is 00_B: In global reset mode
- When CFM[1:0] value is 01_B or 10_B: In channel reset mode
- When the CFE bit in the RSCFDnCFDCFCCK register is cleared to 0.

CFTXIF Flag

The CFTXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 01_B or 10_B, and the interrupt source selected by the CFIM bit in the RSCFDnCFDCFCCK register occurs

The CFTXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFTXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFRXIF Flag

The CFRXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B, and the factor selected by the CFIM bit in the RSCFDnCFDCFCCK register occurs

The CFRXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFRXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFMLT Flag

The CFMLT flag is set to 1 when any of the following conditions is met.

- When an attempt is made to store a new message while the transmit/receive FIFO buffer is full. In this case, the new message is discarded.

The CFMLT flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFMLT flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFLL Flag

The CFLL flag is set to 1 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer matches the FIFO buffer depth set by the CFDC[2:0] bits in the RSCFDnCFDCFCCK register.

The CFLL flag is cleared to 0 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer becomes smaller than the FIFO buffer depth set by the CFDC[2:0] bits.
- When the CFE bit in the RSCFDnCFDCFCCK register is 0 (no transmit/receive FIFO buffer is used): When not in the transmit abort
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

CFEMP Flag

The CFEMP flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B: All messages have been read, or in global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: All messages have been transmitted, or in channel reset mode
- When the CFE bit is 0 (no transmit/receive FIFO buffer is used): Not in the transmit abort

The CFEMP flag is cleared to 0 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B: At least one received message has been stored in the transmit/receive FIFO buffer.
- When the CFM[1:0] bits are set to 01_B: A value of FF_H has been written to the RSCFDnCFDCFPCTRk register after data was written to the RSCFDnCFDCFIDk, RSCFDnCFDCFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDnCFDCFDf_k registers.

NOTE

To clear CFTXIF, CFRXIF, or CFMLT flag to 0, the program must write 0. When writing, use a store instruction to write "0" to the given flag and "1" to other flags.

20.4.8.3 RSCFDnCFDCFPCTRk — Transmit/Receive FIFO Buffer Pointer Control Register k

Note: For the value of index "k", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 20.141 RSCFDnCFDCFPCTRk Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved The write value should be the value after reset.
7 to 0	CFPC[7:0]	Transmit/Receive FIFO Pointer Control <ul style="list-style-type: none"> • Receive mode: Writing FF_H to these bits move the read pointer to the next unread message in the transmit/receive FIFO buffer. • Transmit mode: Writing FF_H to these bits move the write pointer to the next stage of the transmit/receive FIFO buffer. • Gateway mode: Setting prohibited

When the corresponding transmit/receive FIFO buffer is the first transmit/receive FIFO buffer ($k = 3 \times m$) allocated to channel m and when the CFDMAEm bit in the RSCFDnCFDCDTCT register is 1 (DMA transfer request enabled), do not write a value to this register.

CFPC[7:0] Bits

- Receive mode (CFM[1:0] value in the RSCFDnCFDCFCCK register is 00_B):
Writing FF_H to the CFPC[7:0] bits move the read pointer to the next unread message in the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value (transmit/receive FIFO message counter) in the RSCFDnCFDCFSTSk register is decremented by 1. Read the RSCFDnCFDCFDIDk, RSCFDnCFDCFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDnCFDCFDf_d_k registers to read messages from the transmit/receive FIFO buffer, and then write FF_H to the CFPC[7:0] bits.
When writing FF_H to these bits, make sure that the CFE bit in the RSCFDnCFDCFCCK register is set to 1 (transmit/receive FIFO buffers are used) and the CFEMP flag in the RSCFDnCFDCFSTSk register is 0 (the transmit/receive FIFO buffer contains messages).
- Transmit mode (CFM[1:0] value in the RSCFDnCFDCFCCK register is 01_B):
Writing FF_H to the CFPC[7:0] bits store the data written to the RSCFDnCFDCFDIDk, RSCFDnCFDCFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDnCFDCFDf_d_k registers in the transmit/receive FIFO buffer and move the write pointer to the next stage of the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value is incremented by 1. Write transmit messages to the RSCFDnCFDCFDIDk, RSCFDnCFDCFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDnCFDCFDf_d_k registers before writing FF_H to the CFPC[7:0] bits.

When writing FF_H to these bits, make sure that the CFE bit in the RSCFDnCFDCFCCK register is set to 1 and the CFFLL flag in the RSCFDnCFDCFSTSk register is 0 (the transmit/receive FIFO buffer is not full).

- Gateway mode (CFM[1:0] value in the RSCFDnCFDCFCCK register is 10_B):
Setting prohibited

20.4.8.4 RSCFDnCFDCFDk — Transmit/Receive FIFO Buffer Access ID Register k

Note: For the value of index "k", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFIDE	CFRTR	THLEN	CFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.142 RSCFDnCFDCFDk Register Contents

Bit Position	Bit Name	Function
31	CFIDE	Transmit/Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	CFRTR	Transmit/Receive FIFO Buffer RTR/RRS <ul style="list-style-type: none"> When the CFM[1:0] value is 01_B (transmit mode) <ul style="list-style-type: none"> When the transmit message is a classical CAN frame 0: Data frame 1: Remote frame When the transmit message is a CAN FD frame Write 0 to this bit. When the CFM[1:0] value is 00_B (receive mode) <ul style="list-style-type: none"> When the received message is a classical CAN frame 0: Data frame 1: Remote frame When the received message is a CAN FD frame The RRS bit value of the received message can be read.
29	THLEN	Transmit History Data Store Enable This bit is valid only when the CFM[1:0] value is 01 _B (transmit mode). 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	CFID[28:0]	Transmit/Receive FIFO Buffer ID Data <ul style="list-style-type: none"> When CFM[1:0] value is 01_B (transmit mode): Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11. When CFM[1:0] value is 00_B (receive mode): Standard ID or extended ID in the received message can be read. For standard ID, read bits 10 to 0. Bits 28 to 11 are read as 0.

This register can be written only when the CFM[1:0] value in the RSCFDnCFDCFCCK register is 01_B (transmit mode). This register is can be read only when the CFM[1:0] value is 00_B (receive mode). This RSCFDnCFDCFDk register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, this bit is used to set the ID format of the message to be transmitted from the transmit/receive FIFO buffer.

CFRTR Bit

If the received message is a classical CAN frame, this bit indicates the data format (data frame or remote frame) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. If the received message is a CAN FD frame, this bit indicates the RRS bit value of the received message.

When the CFM[1:0] value is 01_B, this bit is used to set the data format of the message to be transmitted from the transmit/receive FIFO buffer.

When the CFFDF bit in the RSCFDnCFDCFFDCSTSk register is 1 (CAN FD frame), set this bit to 0.

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, buffer type, and timestamp) of transmit messages is stored in the transmit history buffer after transmission is completed.

This bit is enabled when the CFM[1:0] value is 01_B (transmit mode).

CFID[28:0] Bits

These bits contain the ID of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B.

When the CFM[1:0] value is 01_B, these bits are used to set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

20.4.8.5 RSCFDnCFDCFPTRk — Transmit/Receive FIFO Buffer Access Pointer Register

Note: For the value of index "k", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDLC[3:0]				CFPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.143 RSCFDnCFDCFPTRk Register Contents

Bit Position	Bit Name	Function																																																																																																						
31 to 28	CFDLC[3:0]	Transmit/Receive FIFO Buffer DLC Data																																																																																																						
		<table border="1"> <thead> <tr> <th>b31</th> <th>b30</th> <th>b29</th> <th>b28</th> <th>Classical CAN Frame</th> <th>CAN FD Frame</th> </tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0 data bytes</td><td></td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td><td>1 data bytes</td><td></td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>0</td><td>2 data bytes</td><td></td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>1</td><td>3 data bytes</td><td></td> </tr> <tr> <td>0</td><td>1</td><td>0</td><td>0</td><td>4 data bytes</td><td></td> </tr> <tr> <td>0</td><td>1</td><td>0</td><td>1</td><td>5 data bytes</td><td></td> </tr> <tr> <td>0</td><td>1</td><td>1</td><td>0</td><td>6 data bytes</td><td></td> </tr> <tr> <td>0</td><td>1</td><td>1</td><td>1</td><td>7 data bytes</td><td></td> </tr> <tr> <td>1</td><td>0</td><td>0</td><td>0</td><td>8 data bytes</td><td></td> </tr> <tr> <td>1</td><td>0</td><td>0</td><td>1</td><td>8 data bytes</td><td>12 data bytes</td> </tr> <tr> <td>1</td><td>0</td><td>1</td><td>0</td><td></td><td>16 data bytes</td> </tr> <tr> <td>1</td><td>0</td><td>1</td><td>1</td><td></td><td>20 data bytes</td> </tr> <tr> <td>1</td><td>1</td><td>0</td><td>0</td><td></td><td>24 data bytes</td> </tr> <tr> <td>1</td><td>1</td><td>0</td><td>1</td><td></td><td>32 data bytes</td> </tr> <tr> <td>1</td><td>1</td><td>1</td><td>0</td><td></td><td>48 data bytes</td> </tr> <tr> <td>1</td><td>1</td><td>1</td><td>1</td><td></td><td>64 data bytes</td> </tr> </tbody> </table>	b31	b30	b29	b28	Classical CAN Frame	CAN FD Frame	0	0	0	0	0 data bytes		0	0	0	1	1 data bytes		0	0	1	0	2 data bytes		0	0	1	1	3 data bytes		0	1	0	0	4 data bytes		0	1	0	1	5 data bytes		0	1	1	0	6 data bytes		0	1	1	1	7 data bytes		1	0	0	0	8 data bytes		1	0	0	1	8 data bytes	12 data bytes	1	0	1	0		16 data bytes	1	0	1	1		20 data bytes	1	1	0	0		24 data bytes	1	1	0	1		32 data bytes	1	1	1	0		48 data bytes	1	1	1	1		64 data bytes
		b31	b30	b29	b28	Classical CAN Frame	CAN FD Frame																																																																																																	
		0	0	0	0	0 data bytes																																																																																																		
		0	0	0	1	1 data bytes																																																																																																		
		0	0	1	0	2 data bytes																																																																																																		
		0	0	1	1	3 data bytes																																																																																																		
		0	1	0	0	4 data bytes																																																																																																		
		0	1	0	1	5 data bytes																																																																																																		
		0	1	1	0	6 data bytes																																																																																																		
		0	1	1	1	7 data bytes																																																																																																		
		1	0	0	0	8 data bytes																																																																																																		
		1	0	0	1	8 data bytes	12 data bytes																																																																																																	
		1	0	1	0		16 data bytes																																																																																																	
		1	0	1	1		20 data bytes																																																																																																	
		1	1	0	0		24 data bytes																																																																																																	
		1	1	0	1		32 data bytes																																																																																																	
1	1	1	0		48 data bytes																																																																																																			
1	1	1	1		64 data bytes																																																																																																			
27 to 16	CFPTR[11:0]	Transmit/Receive FIFO Buffer Label Data <ul style="list-style-type: none"> When CFM[1:0] value is 01_B (transmit mode): <ul style="list-style-type: none"> Set the label information to be stored in the transmit history buffer. Only bits CFPTR[7:0] are valid. When CFM[1:0] value is 00_B (receive mode): <ul style="list-style-type: none"> The label information of the received message can be read. 																																																																																																						
15 to 0	CFTS[15:0]	Transmit/Receive FIFO Buffer Timestamp Data <p>These bits are valid only when the CFM[1:0] value is 00_B (receive mode).</p> <p>The timestamp value of the received message can be read.</p>																																																																																																						

This register can be written only when the CFM[1:0] value in the RSCFDnCFDCFCCK register is 01_B (transmit mode). This register can be read only when the CFM[1:0] value is 00_B (receive mode). This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFDLC[3:0] Bits

These bits indicate the data length of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B.

When the CFM[1:0] value is 01_B, these bits are used to set the data length of the message to be transmitted from the transmit/receive FIFO buffer.

When the CFDLC[3:0] bits are set to 1001_B or more while the CFFDF bit in the RSCFDnCFDCFFDCSTSk register is 0 (CAN frame), 8-byte data is transmitted actually. When the CFFDF bit is 1 (CAN FD frame), the settable value range varies depending on the settings of the TMME bit in the RSCFDnCFDCmFDCFG register and the CFPLS[2:0] bits in the RSCFDnCFDCFCCK register.

- When TMME bit = 0 (transmit buffer merge mode disabled):
A value of 0000_B to 1111_B is settable. If the specified data length exceeds the payload storage size specified by the CFPLS[2:0] bits, excessive payloads are padded by CC_H.
- When TMME bit = 1 (transmit buffer merge mode enabled):
Set the data length within the payload storage size specified by the CFPLS[2:0] bits.

CFPTR[11:0] Bits

These bits indicate the label information attached to the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, the CFPTR[7:0] value is stored in the transmit history buffer when message transmission has been completed.

CFTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the transmit/receive FIFO buffer.

These bits are valid when the CFM[1:0] value is 00_B.

20.4.8.6 RSCFDnCFDCFFDCSTSk — Transmit/Receive FIFO CAN FD Configuration/Status Register k

Note: For the value of index "k", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CFFDF	CFBRS	CFESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 20.144 RSCFDnCFDCFFDCSTSk Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
2	CFFDF	FD format 0: Classical CAN frame 1: CAN FD frame
1	CFBRS	BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes.
0	CFESI	ESI 0: Error active node 1: Error passive node

This register can be written only when the CFM[1:0] value in the RSCFDnCFDCFCCK register is 01_B (transmit mode). This register can be read only when the CFM[1:0] value is 00_B (receive mode). Do not read or write this register when the CFM[1:0] value is 10_B (gateway mode).

CFFDF Bit

When the CFM[1:0] value is 00_B, this bit indicates the FD format (classical CAN frame or CAN FD frame) of the message stored in the transmit/receive FIFO buffer. When the CFM[1:0] value is 01_B, this bit is used to set the FD format of the message to be transmitted from the transmit/receive FIFO buffer.

CFBRS Bit

When the CFM[1:0] value is 00_B, if the CFFDF bit is set to 1, this bit indicates the BRS bit value of the message stored in the transmit/receive FIFO buffer. If the CFFDF bit is set to 0, this bit is always read as 0.

When the CFM[1:0] value is 01_B, if the CFFDF bit is set to 1, this bit is used to set the BRS bit value of the message to be transmitted from the transmit/receive FIFO buffer. If the CFFDF bit is set to 0, write 0 to this bit.

CFESI Bit

When the CFM[1:0] value is 00_B, if the CFFDF bit is set to 1, this bit indicates the ESI bit value of the message stored in the transmit/receive FIFO buffer. If the CFFDF bit is set to 0, this bit is always read as 0.

When the CFM[1:0] value is 01_B, if the CFFDF bit is set to 1, this bit is used to set the ESI bit value of the message to be transmitted from the transmit/receive FIFO buffer. The set value is transmitted when the ESIC bit in the RSCFD_nCFDCmFD CFG register is set to 1 and the channel is in the error active state. When the channel is in the error passive state, the ESI bit value that shows an error passive node is transmitted regardless of this bit value. When the CFFDF bit is set to 0, write 0 to this bit.

20.4.8.7 RSCFDnCFDCFDf_d_k — Transmit/Receive FIFO Buffer Access Data Field d Register k

Note: For the value of index "d" and "k", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB4 × d + 3 [7:0]								CFDB4 × d + 2 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB4 × d + 1 [7:0]								CFDB4 × d + 0 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.145 RSCFDnCFDCFDf_d_k Register Contents

Bit Position	Bit Name	Function
31 to 24	CFDB4 × d + 3 [7:0]	Transmit/Receive FIFO Buffer Data Byte 4 × d + 3
23 to 16	CFDB4 × d + 2 [7:0]	Transmit/Receive FIFO Buffer Data Byte 4 × d + 2
15 to 8	CFDB4 × d + 1 [7:0]	Transmit/Receive FIFO Buffer Data Byte 4 × d + 1
7 to 0	CFDB4 × d + 0 [7:0]	Transmit/Receive FIFO Buffer Data Byte 4 × d + 0 <ul style="list-style-type: none"> When CFM[1:0] value is 01_B (transmit mode): Set the transmit/receive FIFO buffer data. When CFM[1:0] value is 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.

This register can be written only when the CFM[1:0] value in the RSCFDnCFDCFCCK register is 01_B (transmit mode).

This register can be read only when the CFM[1:0] value is 00_B (receive mode). When the CFDLc[3:0] value in the RSCFDnCFDCFPTRk register is smaller than the payload storage size of the transmit/receive FIFO buffer, data bytes for which no data is set are read as 00_H.

Specify the payload storage size of the transmit/receive FIFO buffer by the CFPLS[2:0] bits in the RSCFDnCFDCFCCK register. Do not read or write the RSCFDnCFDCFDf_d_k register corresponding to an area that exceeds the specified size.

This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

20.4.9 Details of FIFO Status-Related Registers

20.4.9.1 RSCFDnCFDFESTS — FIFO Empty Status Register

Value after reset: 03FF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CF14EMP	CF13EMP	CF12EMP	CF11EMP	CF10EMP	CF9EMP	CF8EMP
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7EMP	CF6EMP	CF5EMP	CF4EMP	CF3EMP	CF2EMP	CF1EMP	CF0EMP	RF7EMP	RF6EMP	RF5EMP	RF4EMP	RF3EMP	RF2EMP	RF1EMP	RF0EMP
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.146 RSCFDnCFDFESTS Register Contents

Bit Position	Bit Name	Function
31 to 23	—	Reserved These bits are read as the value after reset.
22	CF14EMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: Transmit/receive FIFO buffer k contains a message. 1: Transmit/receive FIFO buffer k contains no message. (k = see Table 20.6)
21	CF13EMP	
20	CF12EMP	
19	CF11EMP	
18	CF10EMP	
17	CF9EMP	
16	CF8EMP	
15	CF7EMP	
14	CF6EMP	Receive FIFO Buffer Empty Status Flag 0: Receive FIFO buffer x contains an unread message. 1: Receive FIFO buffer x contains no unread message. (x = see Table 20.6)
13	CF5EMP	
12	CF4EMP	
11	CF3EMP	
10	CF2EMP	
9	CF1EMP	
8	CF0EMP	
7	RF7EMP	
6	RF6EMP	Receive FIFO Buffer Empty Status Flag 0: Receive FIFO buffer x contains an unread message. 1: Receive FIFO buffer x contains no unread message. (x = see Table 20.6)
5	RF5EMP	
4	RF4EMP	
3	RF3EMP	
2	RF2EMP	
1	RF1EMP	
0	RF0EMP	

The RSCFDnCFDFESTS register is set to 03FF FFFF_H in global reset mode.

CFkEMP Flag (k = see Table 20.6)

The CFkEMP flag is set to 1 when the CFEMP flag in the RSCFDnCFDCFSTSk register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)). When the CFEMP flag is cleared to 0 (the transmit/receive FIFO buffer contains messages), the CFkEMP flag is cleared to 0.

RFxEMP Flag (x = see Table 20.6)

The RFxEMP flag is set to 1 when the RFEMP flag in the RSCFDnCFDRFSTSt register is set to 1 (the receive FIFO buffer contains no unread message). When the RFEMP flag is cleared to 0 (the receive FIFO buffer contains unread messages), the RFxEMP flag is cleared to 0.

20.4.9.2 RSCFDnCFDFFSTS — FIFO Full Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CF14FL L	CF13FL L	CF12FL L	CF11FL L	CF10FL L	CF9FLL	CF8FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7FLL	CF6FLL	CF5FLL	CF4FLL	CF3FLL	CF2FLL	CF1FLL	CF0FLL	RF7FLL	RF6FLL	RF5FLL	RF4FLL	RF3FLL	RF2FLL	RF1FLL	RF0FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.147 RSCFDnCFDFFSTS Register Contents

Bit Position	Bit Name	Function
31 to 23	—	Reserved These bits are read as the value after reset.
22	CF14FLL	Transmit/Receive FIFO Buffer Full Status Flag 0: Transmit/receive buffer k is not full. 1: Transmit/receive buffer k is full. (k = see Table 20.6)
21	CF13FLL	
20	CF12FLL	
19	CF11FLL	
18	CF10FLL	
17	CF9FLL	
16	CF8FLL	
15	CF7FLL	
14	CF6FLL	Receive FIFO Buffer Full Status Flag 0: Receive FIFO buffer x is not full. 1: Receive FIFO buffer x is full. (x = see Table 20.6)
13	CF5FLL	
12	CF4FLL	
11	CF3FLL	
10	CF2FLL	
9	CF1FLL	
8	CF0FLL	
7	RF7FLL	
6	RF6FLL	Receive FIFO Buffer Full Status Flag 0: Receive FIFO buffer x is not full. 1: Receive FIFO buffer x is full. (x = see Table 20.6)
5	RF5FLL	
4	RF4FLL	
3	RF3FLL	
2	RF2FLL	
1	RF1FLL	
0	RF0FLL	

The RSCFDnCFDFSTSt register is cleared to 0000 0000_H in global reset mode.

CFkFLL Flag (k = see Table 20.6)

The CFkFLL flag is set to 1 when the CFFLL flag in the RSCFDnCFDCFSTSk register is set to 1 (the transmit/receive FIFO buffer is full).

When the CFFLL flag is cleared to 0 (the transmit/receive FIFO buffer is not full), the CFkFLL flag is cleared to 0.

RFxFLL Flag (x = see Table 20.6)

The RFxFLL flag is set to 1 when the RFFLL flag in the RSCFDnCFDRFSTSt register is set to 1 (the receive FIFO buffer is full). When the RFFLL flag is cleared to 0 (the receive FIFO buffer is not full), the RFxFLL flag is cleared to 0.

20.4.9.3 RSCFDnCFDFMSTS — FIFO Message Lost Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CF14MLT	CF13MLT	CF12MLT	CF11MLT	CF10MLT	CF9MLT	CF8MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7MLT	CF6MLT	CF5MLT	CF4MLT	CF3MLT	CF2MLT	CF1MLT	CF0MLT	RF7MLT	RF6MLT	RF5MLT	RF4MLT	RF3MLT	RF2MLT	RF1MLT	RF0MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.148 RSCFDnCFDFMSTS Register Contents

Bit Position	Bit Name	Function
31 to 23	—	Reserved These bits are read as the value after reset.
22	CF14MLT	Transmit/Receive FIFO Buffer Message Lost Status Flag 0: No transmit/receive FIFO buffer k message is lost. 1: A transmit/receive FIFO buffer k message is lost. (k = see Table 20.6)
21	CF13MLT	
20	CF12MLT	
19	CF11MLT	
18	CF10MLT	
17	CF9MLT	
16	CF8MLT	
15	CF7MLT	
14	CF6MLT	Receive FIFO Buffer Message Lost Status Flag 0: No receive FIFO buffer x message is lost. 1: A receive FIFO buffer x message is lost. (x = see Table 20.6)
13	CF5MLT	
12	CF4MLT	
11	CF3MLT	
10	CF2MLT	
9	CF1MLT	
8	CF0MLT	
7	RF7MLT	
6	RF6MLT	Receive FIFO Buffer Message Lost Status Flag 0: No receive FIFO buffer x message is lost. 1: A receive FIFO buffer x message is lost. (x = see Table 20.6)
5	RF5MLT	
4	RF4MLT	
3	RF3MLT	
2	RF2MLT	
1	RF1MLT	
0	RF0MLT	

The RSCFDnCFDFMSTS register is cleared to 0000 0000_H in global reset mode.

CFkMLT Flag (k = see Table 20.6)

The CFkMLT flag is set to 1 when the CFMLT flag in the RSCFDnCFDCFSTSk register is set to 1 (a transmit/receive FIFO message is lost). When the CFMLT flag is cleared to 0, the CFkMLT flag is cleared to 0.

RFxMLT Flag (x = see Table 20.6)

The RFxMLT flag is set to 1 when the RFMLT flag in the RSCFDnCFDRFSTSt register is set to 1 (a receive FIFO message is lost). When the RFMLT flag is cleared to 0, the RFxMLT flag is cleared to 0.

20.4.9.4 RSCFDnCFDRFISTS — Receive FIFO Buffer Interrupt Flag Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RF7IF	RF6IF	RF5IF	RF4IF	RF3IF	RF2IF	RF1IF	RF0IF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.149 RSCFDnCFDRFISTS Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are read as the value after reset.
7	RF7IF	Receive FIFO Buffer Interrupt Request Status Flag
6	RF6IF	0: No receive FIFO buffer x interrupt request is present.
5	RF5IF	1: A receive FIFO buffer x interrupt request is present.
4	RF4IF	(x = see Table 20.6)
3	RF3IF	
2	RF2IF	
1	RF1IF	
0	RF0IF	

The RSCFDnCFDRFISTS register is cleared to 0000 0000_H in global reset mode.

RFxIF Flag (x = see Table 20.6)

The RFxIF flag is set to 1 when the RFIF flag in the RSCFDnCFDRFISTS_x register is set to 1 (a receive FIFO interrupt request is present). When the RFIF flag is cleared to 0, the RFxIF flag is cleared to 0.

20.4.9.5 RSCFDnCFDCFRISTS — Transmit/Receive FIFO Buffer Receive Interrupt Flag Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	CF14RXIF	CF13RXIF	CF12RXIF	CF11RXIF	CF10RXIF	CF9RXIF	CF8RXIF	CF7RXIF	CF6RXIF	CF5RXIF	CF4RXIF	CF3RXIF	CF2RXIF	CF1RXIF	CF0RXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.150 RSCFDnCFDCFRISTS Register Contents

Bit Position	Bit Name	Function
31 to 15	—	Reserved These bits are read as the value after reset.
14	CF14RXIF	Transmit/Receive FIFO Buffer Receive Interrupt Request Status Flag
13	CF13RXIF	0: No transmit/receive FIFO buffer k receive interrupt request is present.
12	CF12RXIF	1: A transmit/receive FIFO buffer k receive interrupt request is present.
11	CF11RXIF	(k = see Table 20.6)
10	CF10RXIF	
9	CF9RXIF	
8	CF8RXIF	
7	CF7RXIF	
6	CF6RXIF	
5	CF5RXIF	
4	CF4RXIF	
3	CF3RXIF	
2	CF2RXIF	
1	CF1RXIF	
0	CF0RXIF	

The RSCFDnCFDCFRISTS register is cleared to 0000 0000_H in global reset mode.

CFkRXIF Flag (k = see Table 20.6)

The CFkRXIF flag is set to 1 when the CFRXIF flag in the RSCFDnCFDCFRISTS_k register is set to 1 (a transmit/receive FIFO receive interrupt request is present). When the CFRXIF flag is cleared to 0, the CFkRXIF flag is cleared to 0.

20.4.9.6 RSCFDnCFDCFTISTS — Transmit/Receive FIFO Buffer Transmit Interrupt Flag Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	CF14TXIF	CF13TXIF	CF12TXIF	CF11TXIF	CF10TXIF	CF9TXIF	CF8TXIF	CF7TXIF	CF6TXIF	CF5TXIF	CF4TXIF	CF3TXIF	CF2TXIF	CF1TXIF	CF0TXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.151 RSCFDnCFDCFTISTS Register Contents

Bit Position	Bit Name	Function
31 to 15	—	Reserved These bits are read as the value after reset.
14	CF14TXIF	Transmit/Receive FIFO Buffer Transmit Interrupt Request Status Flag
13	CF13TXIF	0: No transmit/receive FIFO buffer k transmit interrupt request is present.
12	CF12TXIF	1: A transmit/receive FIFO buffer k transmit interrupt request is present.
11	CF11TXIF	(k = see Table 20.6)
10	CF10TXIF	
9	CF9TXIF	
8	CF8TXIF	
7	CF7TXIF	
6	CF6TXIF	
5	CF5TXIF	
4	CF4TXIF	
3	CF3TXIF	
2	CF2TXIF	
1	CF1TXIF	
0	CF0TXIF	

The RSCFDnCFDCFTISTS register is cleared to 0000 0000_H in global reset mode.

CFkTXIF Flag (k = see Table 20.6)

The CFkTXIF flag is set to 1 when the CFTXIF flag in the RSCFDnCFDCFSTSk register is set to 1 (a transmit/receive FIFO transmit interrupt request is present). When the CFTXIF flag is cleared to 0, the CFkTXIF flag is cleared to 0.

20.4.10 Details of FIFO DMA-Related Registers

20.4.10.1 RSCFDnCFDCTCT – DMA Enable Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	CFDMAE4	CFDMAE3	CFDMAE2	CFDMAE1	CFDMAE0	RFDMAE7	RFDMAE6	RFDMAE5	RFDMAE4	RFDMAE3	RFDMAE2	RFDMAE1	RFDMAE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.152 RSCFDnCFDCTCT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 13	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
12	CFDMAE4	DMA Transfer Enable for Transmit/Receive FIFO buffer 12 0: DMA Transfer Request disabled for Transmit/Receive FIFO buffer 12 1: DMA Transfer Request enabled for Transmit/Receive FIFO buffer 12
11	CFDMAE3	DMA Transfer Enable for Transmit/Receive FIFO buffer 9 0: DMA Transfer Request disabled for Transmit/Receive FIFO buffer 9 1: DMA Transfer Request enabled for Transmit/Receive FIFO buffer 9
10	CFDMAE2	DMA Transfer Enable for Transmit/Receive FIFO buffer 6 0: DMA Transfer Request disabled for Transmit/Receive FIFO buffer 6 1: DMA Transfer Request enabled for Transmit/Receive FIFO buffer 6
9	CFDMAE1	DMA Transfer Enable for Transmit/Receive FIFO buffer 3 0: DMA Transfer Request disabled for Transmit/Receive FIFO buffer 3 1: DMA Transfer Request enabled for Transmit/Receive FIFO buffer 3
8	CFDMAE0	DMA Transfer Enable for Transmit/Receive FIFO buffer 0 0: DMA Transfer Request disabled for Transmit/Receive FIFO buffer 0 1: DMA Transfer Request enabled for Transmit/Receive FIFO buffer 0
7	RFDMAE7	DMA Transfer Enable for Receive FIFO buffer 7 0: DMA Transfer Request disabled for Receive FIFO buffer 7 1: DMA Transfer Request enabled for Receive FIFO buffer 7
6	RFDMAE6	DMA Transfer Enable for Receive FIFO buffer 6 0: DMA Transfer Request disabled for Receive FIFO buffer 6 1: DMA Transfer Request enabled for Receive FIFO buffer 6
5	RFDMAE5	DMA Transfer Enable for Receive FIFO buffer 5 0: DMA Transfer Request disabled for Receive FIFO buffer 5 1: DMA Transfer Request enabled for Receive FIFO buffer 5
4	RFDMAE4	DMA Transfer Enable for Receive FIFO buffer 4 0: DMA Transfer Request disabled for Receive FIFO buffer 4 1: DMA Transfer Request enabled for Receive FIFO buffer 4
3	RFDMAE3	DMA Transfer Enable for Receive FIFO buffer 3 0: DMA Transfer Request disabled for Receive FIFO buffer 3 1: DMA Transfer Request enabled for Receive FIFO buffer 3

Table 20.152 RSCFDnCFDCDTCT Register Contents (2/2)

Bit Position	Bit Name	Function
2	RFDMAE2	DMA Transfer Enable for Receive FIFO buffer 2 0: DMA Transfer Request disabled for Receive FIFO buffer 2 1: DMA Transfer Request enabled for Receive FIFO buffer 2
1	RFDMAE1	DMA Transfer Enable for Receive FIFO buffer 1 0: DMA Transfer Request disabled for Receive FIFO buffer 1 1: DMA Transfer Request enabled for Receive FIFO buffer 1
0	RFDMAE0	DMA Transfer Enable for Receive FIFO buffer 0 0: DMA Transfer Request disabled for Receive FIFO buffer 0 1: DMA Transfer Request enabled for Receive FIFO buffer 0

Modify the RSCFDnCFDCDTCT register in global operating mode or global test mode.

CFDMAEi Bit

This bit is used to enable DMA transfer for transmit/receive FIFO buffer $3 \times m$ (the first transmit/receive FIFO buffer allocated to channel m). DMA transfer is enabled only for transmit/receive FIFO buffers for which the CFM[1:0] bits in the RSCFDnCFDCFCCK register is set to 00_B (receive mode).

Set this bit to 0 when the CFM[1:0] value is 01_B (transmit mode) or 10_B (gateway mode).

RFDMAEe Bit

This bit is used to enable DMA transfer for receive FIFO buffer x. (x = see **Table 20.6**)

20.4.10.2 RSCFDnCFDCDTSTS – DMA Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	CFDMA STS4	CFDMA STS3	CFDMA STS2	CFDMA STS1	CFDMA STS0	RFDMA STS7	RFDMA STS6	RFDMA STS5	RFDMA STS4	RFDMA STS3	RFDMA STS2	RFDMA STS1	RFDMA STS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.153 RSCFDnCFDCDTSTS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 13	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
12	CFDMASTS4	DMA Transfer Status for Transmit/Receive FIFO Buffer 12 0: DMA transfer of transmit/receive FIFO buffer 12 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 12 is in progress.
11	CFDMASTS3	DMA Transfer Status for Transmit/Receive FIFO Buffer 9 0: DMA transfer of transmit/receive FIFO buffer 9 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 9 is in progress.
10	CFDMASTS2	DMA Transfer Status for Transmit/Receive FIFO Buffer 6 0: DMA transfer of transmit/receive FIFO buffer 6 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 6 is in progress.
9	CFDMASTS1	DMA Transfer Status for Transmit/Receive FIFO Buffer 3 0: DMA transfer of transmit/receive FIFO buffer 3 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 3 is in progress.
8	CFDMASTS0	DMA Transfer Status for Transmit/Receive FIFO Buffer 0 0: DMA transfer of transmit/receive FIFO buffer 0 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 0 is in progress.
7	RFDMASTS7	DMA Transfer Status for Receive FIFO Buffer 7 0: DMA transfer of receive FIFO buffer 7 is not in progress. 1: DMA transfer of receive FIFO buffer 7 is in progress.
6	RFDMASTS6	DMA Transfer Status for Receive FIFO Buffer 6 0: DMA transfer of receive FIFO buffer 6 is not in progress. 1: DMA transfer of receive FIFO buffer 6 is in progress.
5	RFDMASTS5	DMA Transfer Status for Receive FIFO Buffer 5 0: DMA transfer of receive FIFO buffer 5 is not in progress. 1: DMA transfer of receive FIFO buffer 5 is in progress.
4	RFDMASTS4	DMA Transfer Status for Receive FIFO Buffer 4 0: DMA transfer of receive FIFO buffer 4 is not in progress. 1: DMA transfer of receive FIFO buffer 4 is in progress.
3	RFDMASTS3	DMA Transfer Status for Receive FIFO Buffer 3 0: DMA transfer of receive FIFO buffer 3 is not in progress. 1: DMA transfer of receive FIFO buffer 3 is in progress.
2	RFDMASTS2	DMA Transfer Status for Receive FIFO Buffer 2 0: DMA transfer of receive FIFO buffer 2 is not in progress. 1: DMA transfer of receive FIFO buffer 2 is in progress.

Table 20.153 RSCFDnCFDCDTSTS Register Contents (2/2)

Bit Position	Bit Name	Function
1	RFDMASTS1	DMA Transfer Status for Receive FIFO Buffer 1 0: DMA transfer of receive FIFO buffer 1 is not in progress. 1: DMA transfer of receive FIFO buffer 1 is in progress.
0	RFDMASTS0	DMA Transfer Status for Receive FIFO Buffer 0 0: DMA transfer of receive FIFO buffer 0 is not in progress. 1: DMA transfer of receive FIFO buffer 0 is in progress.

CFDMASTS_m Bit

When DMA transfer is enabled (CFDMAEm bit in the RSCFDnCFDCDTCT register is 1) for the transmit/receive FIFO buffer $3 \times m$ (the first transmit/receive FIFO buffer allocated to channel m) while the transmit/receive FIFO buffer contains one of more messages, the CFDMASTS_m bit is set to 1 indicating that DMA transfer is in progress.

When all messages in the transmit/receive FIFO buffer have been transferred or DMA transfer is disabled (CFDMAEm bit is 0), the CFDMASTS_m bit is cleared to 0 indicating that DMA transfer has been completed. If the CFDMAEm bit is set to 0 during DMA transfer, the CFDMASTS_m bit is cleared to 0 after the ongoing DMA transfer has been completed (when the message that is being transferred has been transferred to the last byte in the payload storage area).

These bits are cleared to 0 in global reset mode.

RFDMASTS_x Bit

When DMA transfer is enabled (corresponding RFDMAEx bit in the RSCFDnCFDCDTCT register is 1) for the receive FIFO buffer x and the receive FIFO buffer contains one of more messages, the RFDMASTS_x bit is set to 1 indicating that DMA transfer is in progress.

When all messages in the receive FIFO buffer x have been transferred or DMA transfer is disabled (RFDMAEx bit = 0), the RFDMASTS_x bit is cleared to 0 indicating that DMA transfer has been completed. If the RFDMAEx bit is set to 0 during DMA transfer, the RFDMASTS_x bit is cleared to 0 after the ongoing DMA transfer has been completed (when the message that is being transferred has been transferred to the last byte in the payload storage area)

These bits are cleared to 0 in global reset mode.

20.4.11 Details of Transmit Buffer-Related Registers

20.4.11.1 RSCFDnCFDTMCp — Transmit Buffer Control Register p

Note: For the value of index "p", see **Table 20.6**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	TMOM	TMTAR	TMTR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W*1	R/W*1

Note 1. The only effective value for writing to this bit is 1, which sets the bit. Otherwise writing to the bit results in retention of its state.

Table 20.154 RSCFDnCFDTMCp Register Contents

Bit Position	Bit Name	Function
7 to 3	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
2	TMOM	One-Shot Transmission Enable 0: One-shot transmission is disabled. 1: One-shot transmission is enabled.
1	TMTAR	Transmit Abort Request 0: Transmit abort is not requested. 1: Transmit abort is requested.
0	TMTR	Transmit Request 0: Transmission is not requested. 1: Transmission is requested.

When the RSCFDnCFDTMCp register meets any of the following conditions, set it to 00_H.

- The RSCFDnCFDTMCp register corresponds to the transmit buffer number selected by the CFTML[3:0] bits in the RSCFDnCFDCFCCk register ($p = m \times 16 + \text{the value of CFTML}[3:0] \text{ bits}$).
- The RSCFDnCFDTMCp register corresponds to the transmit buffer allocated to the transmit queue by the TXQDC[3:0] bits in the RSCFDnCFDTXQCCm ($m = \text{see Table 20.6}$) register ($p = (m \times 16 + 15) \text{ to } (m \times 16 + 15 + \text{the value of TXQDC}[3:0] \text{ bits})$).
- RSCFDnCFDTMCp register ($p = (m \times 16) + 1, (m \times 16) + 2, (m \times 16) + 4, \text{ or } (m \times 16) + 5$) corresponding to the transmit buffer allocated as a payload storage area when the TMME bit in the RSCFDnCFDCmFDCFG register is 1 (transmit buffer merge mode)

All of the bits in the RSCFDnCFDTMCp register are cleared to 0 in channel reset mode. Modify the RSCFDnCFDTMCp register in channel communication mode or channel halt mode.

TMOM Bit

Setting this bit to 1 enables one-shot transmission. If transmission fails, retransmission defined in the CAN protocol is not performed.

Modify the TMOM bit when the TMTRM flag in the RSCFDnCFDTMSTSp register is set to 0. Set the TMOM bit to 1 together with the TMTR bit.

TMTAR Bit

Setting this bit to 1 generates a transmit abort request for the message stored in the transmit buffer. However, a message that is being transmitted or one that will be transmitted next cannot be aborted.

The TMTAR bit can be set to 1 when TMTR bit is 1.

The TMTAR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed.
- An error or arbitration-lost has been detected.

If this bit becomes 0 at the same time as the program writes 1 to this bit, this bit becomes 0.

TMTR Bit

Setting this bit to 1 transmits the message stored in the transmit buffer.

The TMTR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed after the TMTAR bit was set to 1.
- An error or arbitration-lost has been detected with the TMOM bit set to 1.

Set the TMTR bit to 1 when the value of TMTRF[1:0] in the RSCFDnCFDTMSTSp register is 00_B.

20.4.11.2 RSCFDnCFDTMSTSp — Transmit Buffer Status Register p

Note: For the value of index "p", see **Table 20.6**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	TMTARM	TMTRM	TMTRF[1:0]		TMTSTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R

Table 20.155 RSCFDnCFDTMSTSp Register Contents

Bit Position	Bit Name	Function
7 to 5	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
4	TMTARM	Transmit Buffer Transmit Abort Request Status Flag 0: No transmit abort request is present. 1: A transmit abort request is present.
3	TMTRM	Transmit Buffer Transmit Request Status Flag 0: No transmit request is present. 1: A transmit request is present.
2, 1	TMTRF[1:0]	Transmit Buffer Transmit Result Status Flag b2 b1 0 0: Transmission is in progress or no transmit request is present. 0 1: Transmit abort has been completed. 1 0: Transmission has been completed (without transmit abort request). 1 1: Transmission has been completed (with transmit abort request).
0	TMTSTS	Transmit Buffer Transmit Status Flag 0: Transmission is not in progress. 1: Transmission is in progress.

All of the bits in the RSCFDnCFDTMSTSp register are cleared to 0 in channel reset mode.

TMTARM Flag

The TMTARM flag is set to 1 when the TMTAR bit in the RSCFDnCFDTMCp register is set to 1.

The TMTARM flag is set to 0 when the TMTAR bit in the RSCFDnCFDTMCp register is set to 0.

TMTRM Flag

The TMTRM flag is set to 1 when the TMTR bit in the RSCFDnCFDTMCp register is set to 1.

The TMTRM flag is set to 0 when the TMTR bit in the RSCFDnCFDTMCp register is set to 0.

TMTRF[1:0] Flag

This flag indicates the result of transmission from the transmit buffer.

00_B: Transmission is in progress or no transmit request is present.

01_B: Transmission from the transmit buffer was aborted.

10_B: Transmission has been completed with the TMTAR bit in the RSCFDnCFDTMCp register set to 0 (transmit abort is not requested).

11_B: Transmission has been completed with the TMTAR bit in the RSCFDnCFDTMCp register set to 1 (transmit abort is requested).

Write 00_B to the TMTRF[1:0] flag in channel communication mode or channel halt mode. Do not write any value other than 00_B to this flag.

TMTSTS Flag

This flag is set to 1 when transmission from the transmit buffer starts, and is cleared to 0 when transmission from the transmit buffer has been completed or terminated due to a bus error or arbitration lost.

20.4.11.3 RSCFDnCFDTMIDp — Transmit Buffer ID Register p

Note: For the value of index "p", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIDE	TMRTR	THLEN	TMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.156 RSCFDnCFDTMIDp Register Contents

Bit Position	Bit Name	Function
31	TMIDE	Transmit Buffer IDE 0: Standard ID 1: Extended ID
30	TMRTR	Transmit Buffer RTR/RRS <ul style="list-style-type: none"> When the transmit message is a classical CAN frame 0: Data frame 1: Remote frame When the transmit message is a CAN FD frame Write 0 to this bit.
29	THLEN	Transmit History Data Store Enable 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	TMID[28:0]	Transmit Buffer ID Data Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11.

Modify this register when the TMTRM bit in the corresponding RSCFDnCFDTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write data to this register. If this register is allocated to the transmit queue, only write data to a transmit buffer p ($p = m \times 16 + 15$) for the corresponding channel.

TMIDE Bit

This bit is used to set the ID format of the message to be transmitted from the transmit buffer.

TMRTR Bit

This bit is used to set the data format of the message to be transmitted from the transmit buffer.

Set this bit to 0 when the TMFDF bit in the RSCFDnCFDTMFDCTRp register is 1 (CAN FD frame).

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, buffer type, and timestamp) of transmit messages is stored in the transmit history buffer after transmission is completed.

TMID[28:0] Bits

These bits are used to set the ID of the message to be transmitted from the transmit buffer.

20.4.11.4 RSCFDnCFDTMPTRp — Transmit Buffer Pointer Register p

Note: For the value of index "p", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDLC[3:0]				—	—	—	—	TMPTR[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.157 RSCFDnCFDTMPTRp Register Contents

Bit Position	Bit Name	Function	
31 to 28	TMDLC[3:0]	Transmit Buffer DLC Data	
		b31 b30 b29 b28 Classical CAN Frame CAN FD Frame	
		0 0 0 0 0 data bytes	
		0 0 0 1 1 data bytes	
		0 0 1 0 2 data bytes	
		0 0 1 1 3 data bytes	
		0 1 0 0 4 data bytes	
		0 1 0 1 5 data bytes	
		0 1 1 0 6 data bytes	
		0 1 1 1 7 data bytes	
		1 0 0 0 8 data bytes	
		1 0 0 1 8 data bytes	12 data bytes
		1 0 1 0	16 data bytes
		1 0 1 1	20 data bytes
		1 1 0 0	24 data bytes
		1 1 0 1	32 data bytes
1 1 1 0	48 data bytes		
1 1 1 1	64 data bytes		
27 to 24	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.	
23 to 16	TMPTR[7:0]	Transmit Buffer Label Data Set the label information to be stored in the transmit history buffer.	
15 to 0	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.	

Modify this register when the TMTRM bit in the corresponding RSCFDnCFDTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p ($p = m \times 16 + 15$) for the corresponding channel.

TMDLC[3:0] Bits

These bits are used to set the data length of the message to be transmitted from the transmit buffer when the TMRTR bit in the RSCFDnCFDTMIDp register is set to 0 (data frame).

When the TMDLC[3:0] bits are set to 1001_B or more while the TMFDF bit in the RSCFDnCFDTMFDCTRp register is 0 (classical CAN frame), 8-byte data is transmitted actually. When the TMFDF bit is 1 (CAN FD frame), the settable value range varies depending on the setting of the TMME bit in the RSCFDnCFDCmFDCFG register.

- When the TMME bit = 0 (transmit buffer merge mode disabled):
A value of 0000_B to 1111_B can be set. If a value larger than 1100_B is set, payloads exceeding 20 bytes are padded by CC_H.
- When the TMME bit = 1 (transmit buffer merge mode enabled):
When the corresponding transmit buffer number $p = (m \times 16) + 0$ or $(m \times 16) + 3$, a value of 0000_B to 1111_B can be set. In other cases, set a value of 0000_B to 1011_B (20 data bytes).

When the TMRTR bit is 1 (remote frame), these bits set the length of the message to be requested.

TMPTR[7:0] Bits

When message transmission has been completed, the TMPTR[7:0] value is stored in the transmit history buffer.

20.4.11.5 RSCFDnCFDTMFDCTR_p — Transmit Buffer CAN FD Configuration Register _p

Note: For the value of index "p", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TMFDF	TMBRS	TMESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 20.158 RSCFDnCFDTMFDCTR_p Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved These bits are read as the value after reset. The write value should be the value after reset.
2	TMFDF	FDF 0: Classical CAN frame 1: CAN FD frame
1	TMBRS	BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes.
0	TMESI	ESI 0: Error active node 1: Error passive node

Modify this register when the TMTRM bit in the corresponding RSCFDnCFDTMST_p register is set to 0 (transmission not requested). When this register is linked to the transmit/receive FIFO buffer, do not write data to this register. When this register is allocated to the transmit queue, write data only to transmit buffer p ($p = m \times 16 + 15$) of the corresponding channel.

TMFDF Bit

This bit is used to set the FD format of the message to be transmitted from the transmit buffer.

TMBRS Bit

When this bit is set to 1 while the TMFDF bit is set to 1, the data area of a transmit message is transmitted at the data bit rate. When the TMFDF bit is 0, write set to 0 to this bit.

TMESI Bit

This bit is used to set the ESI bit value of the message to be transmitted from the transmit buffer when the TMFDF bit is set to 1. The set value is transmitted when the ESIC bit in the RSCFDnCFDCmFDCFG register is set to 1 and the channel is in the error active state. When the channel is in the error passive state, the ESI bit value that shows an error passive node is transmitted regardless of this bit value. When the TMFDF bit is set to 0, write 0 to this bit.

20.4.11.6 RSCFDnCFDTMDFb_p — Transmit Buffer Data Field b Register p

Note: For the value of index "b" and "p", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB4 × b + 3 [7:0]								TMDB4 × b + 2 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB4 × b + 1 [7:0]								TMDB4 × b + 0 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.159 RSCFDnCFDTMDFb_p Register Contents

Bit Position	Bit Name	Function
31 to 24	TMDB4 × b + 3 [7:0]	Transmit Buffer Data Byte 4 × b + 3
23 to 16	TMDB4 × b + 2 [7:0]	Transmit Buffer Data Byte 4 × b + 2
15 to 8	TMDB4 × b + 1 [7:0]	Transmit Buffer Data Byte 4 × b + 1
7 to 0	TMDB4 × b + 0 [7:0]	Transmit Buffer Data Byte 4 × b + 0

Set the transmit buffer data.

Modify this register when the TMTRM bit in the corresponding RSCFDnCFDTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p ($p = m \times 16 + 15$) for the corresponding channel.

20.4.11.7 RSCFDnCFDTMIECy — Transmit Buffer Interrupt Enable Configuration Register y

Note: For the value of index "y", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIEp (p = y × 32 + 31)	TMIEp (p = y × 32 + 30)	TMIEp (p = y × 32 + 29)	TMIEp (p = y × 32 + 28)	TMIEp (p = y × 32 + 27)	TMIEp (p = y × 32 + 26)	TMIEp (p = y × 32 + 25)	TMIEp (p = y × 32 + 24)	TMIEp (p = y × 32 + 23)	TMIEp (p = y × 32 + 22)	TMIEp (p = y × 32 + 21)	TMIEp (p = y × 32 + 20)	TMIEp (p = y × 32 + 19)	TMIEp (p = y × 32 + 18)	TMIEp (p = y × 32 + 17)	TMIEp (p = y × 32 + 16)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMIEp (p = y × 32 + 15)	TMIEp (p = y × 32 + 14)	TMIEp (p = y × 32 + 13)	TMIEp (p = y × 32 + 12)	TMIEp (p = y × 32 + 11)	TMIEp (p = y × 32 + 10)	TMIEp (p = y × 32 + 9)	TMIEp (p = y × 32 + 8)	TMIEp (p = y × 32 + 7)	TMIEp (p = y × 32 + 6)	TMIEp (p = y × 32 + 5)	TMIEp (p = y × 32 + 4)	TMIEp (p = y × 32 + 3)	TMIEp (p = y × 32 + 2)	TMIEp (p = y × 32 + 1)	TMIEp (p = y × 32 + 0)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.160 RSCFDnCFDTMIECy Register Contents

Bit Position	Bit Name	Function
31 to 0	TMIEp	Transmit Buffer Interrupt Enable p (p = y × 32 + 31 to y × 32 + 0) 0: Transmit buffer interrupt is disabled 1: Transmit buffer interrupt is enabled

TMIEp Bits (p = see Table 20.6)

When any of these bits is set to 1 and the corresponding transmission has been completed, a transmit buffer interrupt request is generated.

Modify these bits when the TMTRM flag in the corresponding RSCFDnCFDTMSTSp register is 0 (no transmit request is present).

Write 0 to bits corresponding to transmit buffers linked to transmit/receive FIFO buffers or transmit buffers allocated to the transmit queue.

When the TMME bit in the RSCFDnCFDCmFDCFG register is 1 (transmit buffer merge mode enable), set the bit corresponding to the transmit buffer allocated as a payload storage area to 0.

Table 20.161 shows the bit assignment.

Table 20.161 TMIEp Bit Assignment

Bit Position	Channel	Transmit Buffer Number
0	0	0
1	0	1
.	.	.
.	.	.
15	0	15
16	1	0
.	.	.
.	.	.
30	1	14
31	1	15
32	2	0
33	2	1
.	.	.
.	.	.
47	2	15
48	3	0
.	.	.
.	.	.
62	3	14
63	3	15
64	4	0
65	4	1
.	.	.
.	.	.
78	4	14
79	4	15

20.4.12 Details of Transmit Buffer Status-Related Registers

20.4.12.1 RSCFDnCFDTMTRSTSy — Transmit Buffer Transmit Request Status Register y

Note: For the value of index "y", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTRS TSp (p = y × 32 + 31)	TMTRS TSp (p = y × 32 + 30)	TMTRS TSp (p = y × 32 + 29)	TMTRS TSp (p = y × 32 + 28)	TMTRS TSp (p = y × 32 + 27)	TMTRS TSp (p = y × 32 + 26)	TMTRS TSp (p = y × 32 + 25)	TMTRS TSp (p = y × 32 + 24)	TMTRS TSp (p = y × 32 + 23)	TMTRS TSp (p = y × 32 + 22)	TMTRS TSp (p = y × 32 + 21)	TMTRS TSp (p = y × 32 + 20)	TMTRS TSp (p = y × 32 + 19)	TMTRS TSp (p = y × 32 + 18)	TMTRS TSp (p = y × 32 + 17)	TMTRS TSp (p = y × 32 + 16)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTRS TSp (p = y × 32 + 15)	TMTRS TSp (p = y × 32 + 14)	TMTRS TSp (p = y × 32 + 13)	TMTRS TSp (p = y × 32 + 12)	TMTRS TSp (p = y × 32 + 11)	TMTRS TSp (p = y × 32 + 10)	TMTRS TSp (p = y × 32 + 9)	TMTRS TSp (p = y × 32 + 8)	TMTRS TSp (p = y × 32 + 7)	TMTRS TSp (p = y × 32 + 6)	TMTRS TSp (p = y × 32 + 5)	TMTRS TSp (p = y × 32 + 4)	TMTRS TSp (p = y × 32 + 3)	TMTRS TSp (p = y × 32 + 2)	TMTRS TSp (p = y × 32 + 1)	TMTRS TSp (p = y × 32 + 0)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.162 RSCFDnCFDTMTRSTSy Register Contents

Bit Position	Bit Name	Function
31 to 0	TMTRSTSp	Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 31 to y × 32 + 0) 0: No transmit request is present. 1: A transmit request is present.

TMTRSTSp Flags (p = see Table 20.6)

These flags indicate the status of the TMTR bit in the RSCFDnCFDTMCP register.

When the TMTR bit is set to 1 (transmission is requested), the corresponding TMTRSTSp flag is set to 1.

The corresponding TMTRSTSp flag is cleared to 0 when the TMTR bit is set to 0 (transmission is not requested) or in channel reset mode.

Table 20.163 shows the bit assignment.

Table 20.163 TMTRSTSp Bit Assignment

Bit Position	Channel	Transmit Buffer Number
0	0	0
1	0	1
.	.	.
.	.	.
15	0	15
16	1	0
.	.	.
.	.	.
30	1	14
31	1	15
32	2	0
33	2	1
.	.	.
.	.	.
47	2	15
48	3	0
.	.	.
.	.	.
62	3	14
63	3	15
64	4	0
65	4	1
.	.	.
.	.	.
78	4	14
79	4	15

20.4.12.2 RSCFDnCFDTMTARSTSy — Transmit Buffer Transmit Abort Request Status Register y

Note: For the value of index "y", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTARSTSp (p = y × 32 + 31)	TMTARSTSp (p = y × 32 + 30)	TMTARSTSp (p = y × 32 + 29)	TMTARSTSp (p = y × 32 + 28)	TMTARSTSp (p = y × 32 + 27)	TMTARSTSp (p = y × 32 + 26)	TMTARSTSp (p = y × 32 + 25)	TMTARSTSp (p = y × 32 + 24)	TMTARSTSp (p = y × 32 + 23)	TMTARSTSp (p = y × 32 + 22)	TMTARSTSp (p = y × 32 + 21)	TMTARSTSp (p = y × 32 + 20)	TMTARSTSp (p = y × 32 + 19)	TMTARSTSp (p = y × 32 + 18)	TMTARSTSp (p = y × 32 + 17)	TMTARSTSp (p = y × 32 + 16)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTARSTSp (p = y × 32 + 15)	TMTARSTSp (p = y × 32 + 14)	TMTARSTSp (p = y × 32 + 13)	TMTARSTSp (p = y × 32 + 12)	TMTARSTSp (p = y × 32 + 11)	TMTARSTSp (p = y × 32 + 10)	TMTARSTSp (p = y × 32 + 9)	TMTARSTSp (p = y × 32 + 8)	TMTARSTSp (p = y × 32 + 7)	TMTARSTSp (p = y × 32 + 6)	TMTARSTSp (p = y × 32 + 5)	TMTARSTSp (p = y × 32 + 4)	TMTARSTSp (p = y × 32 + 3)	TMTARSTSp (p = y × 32 + 2)	TMTARSTSp (p = y × 32 + 1)	TMTARSTSp (p = y × 32 + 0)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.164 RSCFDnCFDTMTARSTSy Register Contents

Bit Position	Bit Name	Function
31 to 0	TMTARSTSp	Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 31 to y × 32 + 0) 0: No transmit abort request is present. 1: A transmit abort request is present.

TMTARSTSp Flags (p = see Table 20.6)

These flags indicate the status of the TMTAR bit in the RSCFDnCFDTMCP register.

When the TMTAR bit is set to 1 (transmit abort is requested), the corresponding TMTARSTSp flag is set to 1.

The corresponding TMTARSTSp flag is cleared to 0 when the TMTAR bit is set to 0 (transmit abort is not requested) or in channel reset mode.

Table 20.165 shows the bit assignment.

Table 20.165 TMTARSTSp Bit Assignment

Bit Position	Channel	Transmit Buffer Number
0	0	0
1	0	1
.	.	.
.	.	.
15	0	15
16	1	0
.	.	.
.	.	.
30	1	14
31	1	15
32	2	0
33	2	1
.	.	.
.	.	.
47	2	15
48	3	0
.	.	.
.	.	.
62	3	14
63	3	15
64	4	0
65	4	1
.	.	.
.	.	.
78	4	14
79	4	15

20.4.12.3 RSCFDnCFDTMTCSTSy — Transmit Buffer Transmit Complete Status Register y

Note: For the value of index "y", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTCSTSp (p = y × 32 + 31)	TMTCSTSp (p = y × 32 + 30)	TMTCSTSp (p = y × 32 + 29)	TMTCSTSp (p = y × 32 + 28)	TMTCSTSp (p = y × 32 + 27)	TMTCSTSp (p = y × 32 + 26)	TMTCSTSp (p = y × 32 + 25)	TMTCSTSp (p = y × 32 + 24)	TMTCSTSp (p = y × 32 + 23)	TMTCSTSp (p = y × 32 + 22)	TMTCSTSp (p = y × 32 + 21)	TMTCSTSp (p = y × 32 + 20)	TMTCSTSp (p = y × 32 + 19)	TMTCSTSp (p = y × 32 + 18)	TMTCSTSp (p = y × 32 + 17)	TMTCSTSp (p = y × 32 + 16)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTCSTSp (p = y × 32 + 15)	TMTCSTSp (p = y × 32 + 14)	TMTCSTSp (p = y × 32 + 13)	TMTCSTSp (p = y × 32 + 12)	TMTCSTSp (p = y × 32 + 11)	TMTCSTSp (p = y × 32 + 10)	TMTCSTSp (p = y × 32 + 9)	TMTCSTSp (p = y × 32 + 8)	TMTCSTSp (p = y × 32 + 7)	TMTCSTSp (p = y × 32 + 6)	TMTCSTSp (p = y × 32 + 5)	TMTCSTSp (p = y × 32 + 4)	TMTCSTSp (p = y × 32 + 3)	TMTCSTSp (p = y × 32 + 2)	TMTCSTSp (p = y × 32 + 1)	TMTCSTSp (p = y × 32 + 0)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.166 RSCFDnCFDTMTCSTSy Register Contents

Bit Position	Bit Name	Function
31 to 0	TMTCSTSp	Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 31 to y × 32 + 0) 0: Transmission has not been completed. 1: Transmission has been completed.

TMTCSTSp Flags (p = see Table 20.6)

When the TMTRF[1:0] flag in the RSCFDnCFDTMSTSp register is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)), the corresponding TMTCSTSp flag is set to 1.

To clear the TMTCSTSp flag to 0, set the corresponding TMTRF[1:0] flag to 00_B. This flag is cleared to 0 in channel reset mode.

Table 20.167 shows the bit assignment.

Table 20.167 TMTCSSTp Bit Assignment

Bit Position	Channel	Transmit Buffer Number
0	0	0
1	0	1
.	.	.
.	.	.
15	0	15
16	1	0
.	.	.
.	.	.
30	1	14
31	1	15
32	2	0
33	2	1
.	.	.
.	.	.
47	2	15
48	3	0
.	.	.
.	.	.
62	3	14
63	3	15
64	4	0
65	4	1
.	.	.
.	.	.
78	4	14
79	4	15

20.4.12.4 RSCFDnCFDTMTASTSy — Transmit Buffer Transmit Abort Status Register y

Note: For the value of index "y", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTAS TSp (p = y × 32 + 31)	TMTAS TSp (p = y × 32 + 30)	TMTAS TSp (p = y × 32 + 29)	TMTAS TSp (p = y × 32 + 28)	TMTAS TSp (p = y × 32 + 27)	TMTAS TSp (p = y × 32 + 26)	TMTAS TSp (p = y × 32 + 25)	TMTAS TSp (p = y × 32 + 24)	TMTAS TSp (p = y × 32 + 23)	TMTAS TSp (p = y × 32 + 22)	TMTAS TSp (p = y × 32 + 21)	TMTAS TSp (p = y × 32 + 20)	TMTAS TSp (p = y × 32 + 19)	TMTAS TSp (p = y × 32 + 18)	TMTAS TSp (p = y × 32 + 17)	TMTAS TSp (p = y × 32 + 16)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTAS TSp (p = y × 32 + 15)	TMTAS TSp (p = y × 32 + 14)	TMTAS TSp (p = y × 32 + 13)	TMTAS TSp (p = y × 32 + 12)	TMTAS TSp (p = y × 32 + 11)	TMTAS TSp (p = y × 32 + 10)	TMTAS TSp (p = y × 32 + 9)	TMTAS TSp (p = y × 32 + 8)	TMTAS TSp (p = y × 32 + 7)	TMTAS TSp (p = y × 32 + 6)	TMTAS TSp (p = y × 32 + 5)	TMTAS TSp (p = y × 32 + 4)	TMTAS TSp (p = y × 32 + 3)	TMTAS TSp (p = y × 32 + 2)	TMTAS TSp (p = y × 32 + 1)	TMTAS TSp (p = y × 32 + 0)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.168 RSCFDnCFDTMTASTSy Register Contents

Bit Position	Bit Name	Function
31 to 0	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 31 to y × 32 + 0) 0: Transmission is not aborted 1: Transmission is aborted

TMTASTSp Flags (p = see Table 20.6)

When the TMTRF[1:0] flag in the RSCFDnCFDTMSTSp register is set to 01_B (transmit abort has been completed), the corresponding TMTASTSp flag is set to 1.

To clear the TMTASTSp flag to 0, set the corresponding TMTRF[1:0] flag to 00_B. This flag is cleared to 0 in channel reset mode.

Table 20.169 shows the bit assignment.

Table 20.169 TMTASTSp Bit Assignment

Bit Position	Channel	Transmit Buffer Number
0	0	0
1	0	1
.	.	.
.	.	.
15	0	15
16	1	0
.	.	.
.	.	.
30	1	14
31	1	15
32	2	0
33	2	1
.	.	.
.	.	.
47	2	15
48	3	0
.	.	.
.	.	.
62	3	14
63	3	15
64	4	0
65	4	1
.	.	.
.	.	.
78	4	14
79	4	15

20.4.13 Details of Transmit Queue-Related Registers

20.4.13.1 RSCFDnCFDTXQCCm — Transmit Queue Configuration and Control Register m

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TXQIM	TXQIE	TXQDC[3:0]			—	—	—	—	—	—	—	—	TXQE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 20.170 RSCFDnCFDTXQCCm Register Contents

Bit Position	Bit Name	Function
31 to 14	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
13	TXQIM	Transmit Queue Interrupt Source Select 0: When the transmit queue becomes empty upon completion of message transmission, a transmit queue interrupt request is generated. 1: A transmit queue interrupt request is generated each time a message has been transmitted.
12	TXQIE	Transmit Queue Interrupt Enable 0: Transmit queue interrupt is disabled. 1: Transmit queue interrupt is enabled.
11 to 8	TXQDC[3:0]	Transmit Queue Depth Configuration Setting these bits to g (g = 2 to 15) makes the (g + 1) transmit queue available. Setting these bits to 0 disables the transmit queue. Setting these bits to 1 is prohibited. For transmit buffer merge mode, set g to 2 to 9.
7 to 1	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	TXQE	Transmit Queue Enable 0: The transmit queue is not used. 1: The transmit queue is used.

TXQIM Bit

This bit is used to select a transmit queue interrupt source. Modify this bit in channel reset mode.

TXQIE Bit

When the TXQIE bit is set to 1 and the source selected by the TXQIM bit occurs, an interrupt request is generated.

Set the TXQE bit to 0 before modifying the TXQIE bit.

TXQDC[3:0] Bits

These bits are used to specify the number of transmit buffers to be allocated to the transmit queues. Transmit buffers are allocated to transmit queues in descending order of buffer number, that is, from $(m \times 16 + 15)$ to $(m \times 16 + 0)$. see

Table 20.98. For examples of how buffer allocation is done, see **Figure 20.9**.

When the TMME bit in the RSCFDnCFDCmFDCFG register is 1 (transmit buffer merge mode), transmit buffers $(m \times 16 + 5)$ to $(m \times 16 + 0)$ are merged and cannot be allocated to the transmit queue. Therefore, do not set TXQDC[3:0] bits to 10 to 15.

Modify these bits only in channel reset mode.

TXQE Bit

Setting this bit to 1 enables the transmit queue. Modify this bit in channel communication mode or channel halt mode.

This bit is cleared to 0 in channel reset mode.

Before setting the TXQE bit to 1, set the TXQDC[3:0] bits to 0010_B or more.

20.4.13.2 RSCFDnCFDTXQSTSm — Transmit Queue Status Register m

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQIF	TXQFL L	TXQEM P
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W* ¹	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 20.171 RSCFDnCFDTXQSTSm Register Contents

Bit Position	Bit Name	Function
31 to 13	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
12 to 8	—	Reserved When read, the undefined value is returned. When writing to these bits, write the value after reset.
7 to 3	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	TXQIF	Transmit Queue Interrupt Request Flag 0: No transmit queue interrupt request is present. 1: A transmit queue interrupt request is present.
1	TXQFLL	Transmit Queue Full Status Flag 0: The transmit queue is not full. 1: The transmit queue is full.
0	TXQEMP	Transmit Queue Empty Status Flag 0: The transmit queue contains messages. 1: The transmit queue contains no message (transmit queue empty).

TXQIF Flag

The TXQIF flag is set to 1 when the interrupt source specified by the TXQIM bit in the RSCFDnCFDTXQCCm register has occurred.

The TXQIF flag is cleared to 0 in channel reset mode or by writing 0 to this flag. This flag is not cleared to 0 by setting the TXQE bit in the RSCFDnCFDTXQCCm register to 0 (the transmit queue is not used).

TXQFLL Flag

The TXQFLL flag is set to 1 when the number of messages stored in the transmit queue matches the transmit queue depth set by the TXQDC[3:0] bits in the RSCFDnCFDTXQCCm register.

This flag is cleared to 0 in any of the following cases.

- The number of messages set for the transmit queue is smaller than the transmit queue depth set by the TXQDC[3:0] bits.

- In channel reset mode

TXQEMP Flag

The TXQEMP flag is cleared to 0 when even a single message stored in the transmit queue.

This flag is set to 1 in any of the following cases.

- The TXQE bit is set to 0 (the transmit queue is not used).
- The transmit queue becomes empty.
- In channel reset mode

20.4.13.3 RSCFDnCFDTXQPCTRM — Transmit Queue Pointer Control Register m

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 20.172 RSCFDnCFDTXQPCTRM Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved The write value should be the value after reset.
7 to 0	TXQPC[7:0]	Transmit Queue Pointer Control Writing FF _H to these bits move the write pointer of the transmit queue to the next queue buffer.

TXQPC[7:0] Bits

Writing FF_H to the TXQPC[7:0] bits move the write pointer to the next transmit queue buffer and generates a transmit request for the message. Write transmit messages to the RSCFDnCFDTMID_p, RSCFDnCFDTMPTR_p, RSCFDnCFDTMFDCTR_p, and RSCFDnCFDTMDFb_p registers (p = 15, 31, 47, 63, 79) before writing FF_H to the TXQPC[7:0] bits.

When writing FF_H to these bits, make sure that the TXQE bit in the RSCFDnCFDTXQCC_m register is set to 1 (the transmit queue is used) and the TXQFLL flag in the RSCFDnCFDTXQSTSM register is 0 (The transmit queue is not full).

20.4.14 Details of Transmit History-Related Registers

20.4.14.1 RSCFDnCFDTHLCCm — Transmit History List Configuration and Control Register m

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	THLDT E	THLIM	THLIE	—	—	—	—	—	—	—	THLE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 20.173 RSCFDnCFDTHLCCm Register Contents

Bit Position	Bit Name	Function
31 to 11	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
10	THLDTE	Transmit History Target Buffer Select 0: Entries from transmit/receive FIFO buffers and transmit queue 1: Entries from transmit buffers, transmit/receive FIFO buffers, and transmit queue
9	THLIM	Transmit History Interrupt Source Select 0: When 12 items of data have been stored in the transmit history buffer 1: Each time transmit history data is stored in the transmit history buffer
8	THLIE	Transmit History Interrupt Enable 0: Transmit history interrupt is disabled. 1: Transmit history interrupt is enabled.
7 to 1	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	THLE	Transmit History Buffer Enable 0: Transmit history buffer is not used. 1: Transmit history buffer is used.

THLDTE Bit

When this bit is set to 0, the transmit history data of messages transmitted from transmit/receive FIFO buffers and the transmit queue is stored in the transmit history buffer. When this bit is set to 1, the transmit history data of messages transmitted from transmit buffers, transmit/receive FIFO buffers, and the transmit queue is stored in the transmit history buffer.

Modify this bit only in channel reset mode.

THLIM Bit

This bit is used to select a transmit history interrupt source.

Modify this bit only in channel reset mode.

THLIE Bit

When the THLIE bit is set to 1 and the source selected by the THLIM bit has occurred, a transmit history interrupt request is generated. Modify the THLIE bit only when the THLE bit is set to 0.

THLE Bit

Setting this bit to 1 enables the transmit history buffer. When data transmission from the buffer selected by the THLDTE bit has been completed, the transmit history data of transmit messages is stored in the transmit history buffer.

Modify this bit in channel communication mode or channel halt mode.

This bit is cleared to 0 in channel reset mode.

20.4.14.2 RSCFDnCFDTHLSTSm — Transmit History List Status Register m

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THLMC[4:0]				—	—	—	—	THLIF	THLELT	THLFLL	THLEMP	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 20.174 RSCFDnCFDTHLSTSm Register Contents

Bit Position	Bit Name	Function
31 to 13	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
12 to 8	THLMC[4:0]	Transmit History Buffer Unread Data Counter These bits indicate the number of unread data stored in the transmit history buffer.
7 to 4	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	THLIF	Transmit History Interrupt Request Flag 0: No transmit history interrupt request is present. 1: A transmit history interrupt request is present.
2	THLELT	Transmit History Buffer Overflow Flag 0: Transmit history buffer overflow has not occurred. 1: Transmit history buffer overflow has occurred.
1	THLFLL	Transmit history Buffer Full Status Flag 0: Transmit history buffer is not full. 1: Transmit history buffer is full.
0	THLEMP	Transmit History Buffer Empty Status Flag 0: Transmit history buffer contains unread data. 1: Transmit history buffer contains no unread data (buffer empty).

THLMC[4:0] Bits

These bits indicate the number of unread data stored in the transmit history buffer. These bits are cleared to 0 in channel reset mode.

THLIF Flag

The THLIF flag is set to 1 when the interrupt source specified with the THLIM bit in the RSCFDnCFDTHLCCm register occurs.

This flag is cleared to 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to "0" to "0", and the bits not to be set to "0" to "1".

THLELT Flag

The THLELT flag is set to 1 when an attempt is made to store new transmit history data while the transmit history buffer is full. In this case, the new data is discarded. This flag becomes 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

THLFLL Flag

The THLFLL flag is set to 1 when 16 items of data have been stored in the transmit history buffer, and is cleared to 0 when the number of data stored in the transmit history buffer has decreased to less than 16. This bit is also cleared to 0 in channel reset mode or when the THLE bit in the RSCFDnCFDTHLCCm register is set to 0 (transmit history buffer is not used).

THLEMP Flag

The THLEMP flag is cleared to 0 when even a single item of transmit history data has been stored in the transmit history buffer.

This flag is set to 1 when all the data in the transmit history buffer has been read. This flag is also set to 1 in channel reset mode or when the THLE bit in the RSCFDnCFDTHLCCm register is set to 0 (transmit history buffer is not used).

NOTE

To clear THLIF or THLELT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

20.4.14.3 RSCFDnCFDTHLPCTRm — Transmit History List Pointer Control Register m

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	THLPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 20.175 RSCFDnCFDTHLPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When writing to these bits, write the value after reset.
7 to 0	THLPC[7:0]	Transmit History List Pointer Control Writing FF _H to these bits move the read pointer to the next unread data in the transmit history buffer.

THLPC[7:0] Bits

When the THLPC[7:0] bits are set to FF_H, the read pointer moves to the next data in the transmit history buffer. At this time, the THLMC[4:0] (transmit history buffer unread data counter) value in the RSCFDnCFDTHLSTSm register is decremented by 1. Write FF_H to the THLPC[7:0] bits after reading from the RSCFDnCFDTHLACCm register.

When writing FF_H to these bits, make sure that the THLE bit in the RSCFDnCFDTHLCCm register is set to 1 (transmit history buffer is used) and the THLEMP flag in the RSCFDnCFDTHLSTSm register is 0.

20.4.14.4 RSCFDnCFDTHLACCm — Transmit History List Access Register m

Note: For the value of index "m", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TID[7:0]							—	BN[3:0]			BT[2:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.176 RSCFDnCFDTHLACCm Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTS[15:0]	Timestamp Data The timestamp data of stored data can be read.
15 to 8	TID[7:0]	Label Data The label information of stored data can be read.
7	—	Reserved When read, the value after reset is returned.
6 to 3	BN[3:0]	Buffer Number Data The buffer number of transmit source (transmit buffer, transmit/receive FIFO or transmit queue) can be read.
2 to 0	BT[2:0]	Buffer Type Data b2 b1 b0 0 0 1: Transmit buffer 0 1 0: Transmit/receive FIFO buffer 1 0 0: Transmit queue

TMTS[15:0] Bits

Timestamp values in transmit history data stored in the transmit history buffer is displayed.

TID[7:0] Bits

These bits indicate the label information of transmit history data stored in the transmit history buffer.

BN[3:0] Bits

These bits indicate the transmit source buffer number in the transmit history data stored in the transmit history buffer.

BT[2:0] Bits

These bits indicate the type of the transmit source buffer in the transmit history data stored in the transmit history buffer.

20.4.15 Details of Test-Related Registers

20.4.15.1 RSCFDnCFDGTSTCFG — Global Test Configuration Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	RTMPS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	C4ICBCE	C3ICBCE	C2ICBCE	C1ICBCE	C0ICBCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 20.177 RSCFDnCFDGTSTCFG Register Contents

Bit Position	Bit Name	Function
31 to 23	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
22 to 16	RTMPS[6:0]	RAM Test Page Configuration Set a value within a range of page 0 (00 _H) to page 69 (45 _H).
15 to 5	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
4	C4ICBCE	CAN4 Inter-channel Communication Test Enable 0: CAN4 inter-channel communication test is disabled. 1: CAN4 inter-channel communication test is enabled.
3	C3ICBCE	CAN3 Inter-channel Communication Test Enable 0: CAN3 inter-channel communication test is disabled. 1: CAN3 inter-channel communication test is enabled.
2	C2ICBCE	CAN2 Inter-channel Communication Test Enable 0: CAN2 inter-channel communication test is disabled. 1: CAN2 inter-channel communication test is enabled.
1	C1ICBCE	CAN1 Inter-Channel Communication Test Enable 0: CAN1 inter-channel communication test is disabled. 1: CAN1 inter-channel communication test is enabled.
0	C0ICBCE	CAN0 Inter-Channel Communication Test Enable 0: CAN0 inter-channel communication test is disabled. 1: CAN0 inter-channel communication test is enabled.

Modify the RSCFDnCFDGTSTCFG register only in global test mode.

RTMPS[6:0] Bits

These bits are used to set the RAM test target page number for RAM test. Set a value in the range of 00_H to 45_H, inclusive. In CAN FD mode, do not access more than 96 bytes in the last page (RTMPS=45_H) during RAM test.

C4ICBCE Bit

Setting this bit to 1 enables the channel 4 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C3ICBCE Bit

Setting this bit to 1 enables the channel 3 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C2ICBCE Bit

Setting this bit to 1 enables the channel 2 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C1ICBCE Bit

Setting this bit to 1 enables the channel 1 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C0ICBCE Bit

Setting this bit to 1 enables the channel 0 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

20.4.15.2 RSCFDnCFDGTSTCTR — Global Test Control Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	ICBCTME
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W

Table 20.178 RSCFDnCFDGTSTCTR Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	RTME	RAM Test Enable 0: RAM test is disabled. 1: RAM test is enabled.
1	—	Reserved When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	ICBCTME	Inter-Channel Communication Test Enable 0: Inter-channel communication test is disabled. 1: Inter-channel communication test is enabled.

RTME Bit

Setting this bit to 1 enables the RAM test. Modify this bit only in global test mode.

This bit is cleared to 0 in global reset mode.

1. Set the GMDC[1:0] bits in the RSCFDnCFDGTSTCTR register to 10_B (Global test mode).
2. Set the RTME bit to 1.
3. Check that the RTME bit is set to 1.

For the setting procedure of RAM test, see **Figure 20.37**.

ICBCTME Bit

When this bit is set to 1, a communication test is enabled between the channels for which the CmICBCE bit (m = see **Table 20.6**) in the RSCFDnCFDGTSTCFG register has been set to 1. Modify the ICBCTME bit only in global test mode. This bit is cleared to 0 in global reset mode.

20.4.15.3 RSCFDnCFDGLCKK — Global Lock Key Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOCK[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1

Note 1. Writing to these bits is effective only when the RS-CANFD module is in global test mode.

Table 20.179 RSCFDnCFDGLCKK Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing these bits, write the value after reset.
15 to 0	LOCK[15:0]	Lock Key These bits are key bits to release protection of test mode.

The RSCFDnCFDGLCKK register releases protection of special test bits and is write only.

For the protection release data, see **Section 20.11.4.2, Procedure for Releasing the Protection.**

LOCK[15:0] Bits

Writing the protection release data to the LOCK[15:0] bits in succession enables writing 1 to the RTME bit in the RSCFDnCFDGTSTCTR register.

After the protection has been released, writing to the I/O register area (<RSCFDn_base> + 0000_H to <RSCFDn_base> + 05FF_H) of the CAN (except the RAM) enables the protection again.

Reading from the I/O register area of the CAN or reading from/writing to other areas does not enable the protection.

20.4.15.4 RSCFDnCFDRPGACC_r — RAM Test Page Access Register *r*

Note: For the value of index "r", see **Table 20.6**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDTA[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDTA[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.180 RSCFDnCFDRPGACC_r Register Contents

Bit Position	Bit Name	Function
31 to 0	RDTA[31:0]	RAM Data Test Access RAM data for CAN can be read and written.

Modify the RSCFDnCFDRPGACC_r register in global test mode with the RTME bit in the RSCFDnCFDGTSTCTR register set to 1 (RAM test is enabled).

The RSCFDnCFDRPGACC_r register can be read and written when the RTME bit is set to 1.

20.5 Interrupt Sources and DMA Trigger

20.5.1 Interrupt Sources

The RS-CANFD module has 17 interrupts that are grouped into global interrupts and channel interrupts.

- Global interrupts (2 interrupts per unit):
 - Receive FIFO interrupt
 - Global error interrupt
- Channel interrupts (3 interrupts per channel):
 - CAN_m transmit interrupt (m = see **Table 20.6**)
 - CAN_m transmit complete interrupt
 - CAN_m transmit abort interrupt
 - CAN_m transmit/receive FIFO transmit complete interrupt (in transmit mode or gateway mode)
 - CAN_m transmit history interrupt
 - CAN_m transmit queue interrupt
 - CAN_m transmit/receive FIFO receive complete interrupt (in receive mode or gateway mode)
 - CAN_m error interrupt

When an interrupt request is generated, the corresponding interrupt request flag is set to 1 (interrupt request present). In this case, when the interrupt enable bit is 1 (enabling interrupts), an interrupt request is output from the RS-CANFD module. (Generation of interrupts also depends on the interrupt control register settings of the interrupt controller.)

Setting the interrupt request flag to 0 (no interrupt request present) or setting the interrupt enable bit to 0 (disabling interrupts) clears the interrupt request. The interrupt request is kept being output until the interrupt request flag is cleared.

Table 20.181 lists the CAN interrupt sources. **Figure 20.2** shows the CAN global interrupt block diagram, and **Figure 20.3** shows the CAN channel interrupt block diagram.

Table 20.181 List of CAN Interrupt Sources (1/2)

		Interrupt Source	Corresponding Interrupt Request Flag	Corresponding Interrupt Enable Bit
Global interrupts	Receive FIFO	Receive FIFO 0	RFIF flag in the RSCFDn(CFD)RFSTS0 register	RFIE bit in the RSCFDn(CFD)RFCC0 register
		Receive FIFO 1	RFIF flag in the RSCFDn(CFD)RFSTS1 register	RFIE bit in the RSCFDn(CFD)RFCC1 register
		Receive FIFO 2	RFIF flag in the RSCFDn(CFD)RFSTS2 register	RFIE bit in the RSCFDn(CFD)RFCC2 register
		Receive FIFO 3	RFIF flag in the RSCFDn(CFD)RFSTS3 register	RFIE bit in the RSCFDn(CFD)RFCC3 register
		Receive FIFO 4	RFIF flag in the RSCFDn(CFD)RFSTS4 register	RFIE bit in the RSCFDn(CFD)RFCC4 register
		Receive FIFO 5	RFIF flag in the RSCFDn(CFD)RFSTS5 register	RFIE bit in the RSCFDn(CFD)RFCC5 register
		Receive FIFO 6	RFIF flag in the RSCFDn(CFD)RFSTS6 register	RFIE bit in the RSCFDn(CFD)RFCC6 register
		Receive FIFO 7	RFIF flag in the RSCFDn(CFD)RFSTS7 register	RFIE bit in the RSCFDn(CFD)RFCC7 register
Global interrupts	Global error	<ul style="list-style-type: none"> • DEF flag in the RSCFDn(CFD)GERFL register • MES flag in the RSCFDn(CFD)GERFL register • THLES flag in the RSCFDn(CFD)GERFL register • CMPOF flag in the RSCFDn(CFD)GERFL register 	<ul style="list-style-type: none"> • DEIE bit in the RSCFDn(CFD)GCTR register • MEIE bit in the RSCFDn(CFD)GCTR register • THLEIE bit in the RSCFDn(CFD)GCTR register • CMPOFIE bit in the RSCFDn(CFD)GCTR register 	
Channel interrupts (m = see Table 20.6)	CANm transmit	CANm transmit complete	TMTRF[1:0] flag in the RSCFDn(CFD)TMSTSp register	TMIE bit in the RSCFDn(CFD)TMIECy register
		CANm transmit abort	TMTRF[1:0] flag in the RSCFDn(CFD)TMSTSp register	TAIE bit in the RSCFDn(CFD)CmCTR register
		CANm transmit/receive FIFO transmit complete	CCTXIF flag in the RSCFDn(CFD)CFSTSk register	CCTXIE bit in the RSCFDn(CFD)CFCCk register
		CANm transmit queue	TXQIF flag in the RSCFDn(CFD)TXQSTSm register	TXQIE bit in the RSCFDn(CFD)TXQCCm register
		CANm transmit history	THLIF flag in the RSCFDn(CFD)THLSTSm register	THLIE bit in the RSCFDn(CFD)THLCCm register
		CANm transmit/receive FIFO receive complete	CFRXIF flag in the RSCFDn(CFD)CFSTSk register	CFRXIE bit in the RSCFDn(CFD)CFCCk register

Table 20.181 List of CAN Interrupt Sources (2/2)

Interrupt Source	Corresponding Interrupt Request Flag	Corresponding Interrupt Enable Bit
CANm error	• BEF flag in the RSCFDn(CFD)CmERFL register	• BEIE bit in the RSCFDn(CFD)CmCTR register
	• ALF flag in the RSCFDn(CFD)CmERFL register	• ALIE bit in the RSCFDn(CFD)CmCTR register
	• BLF flag in the RSCFDn(CFD)CmERFL register	• BLIE bit in the RSCFDn(CFD)CmCTR register
	• OVLV flag in the RSCFDn(CFD)CmERFL register	• OLIE bit in the RSCFDn(CFD)CmCTR register
	• BORF flag in the RSCFDn(CFD)CmERFL register	• BORIE bit in the RSCFDn(CFD)CmCTR register
	• BOEF flag in the RSCFDn(CFD)CmERFL register	• BOEIE bit in the RSCFDn(CFD)CmCTR register
	• EPF flag in the RSCFDn(CFD)CmERFL register	• EPIE bit in the RSCFDn(CFD)CmCTR register
	• EWF flag in the RSCFDn(CFD)CmERFL register	• EWIE bit in the RSCFDn(CFD)CmCTR register
	• SOCO flag in the RSCFDnCFDCmFDSTS register	• SOCOIE bit in the RSCFDnCFDCmCTR register
	• EOCO flag in the RSCFDnCFDCmFDSTS register	• EOCOIE bit in the RSCFDnCFDCmCTR register
	• TDCVF flag in the RSCFDnCFDCmFDSTS register	• TDCVFIE bit in the RSCFDnCFDCmCTR register

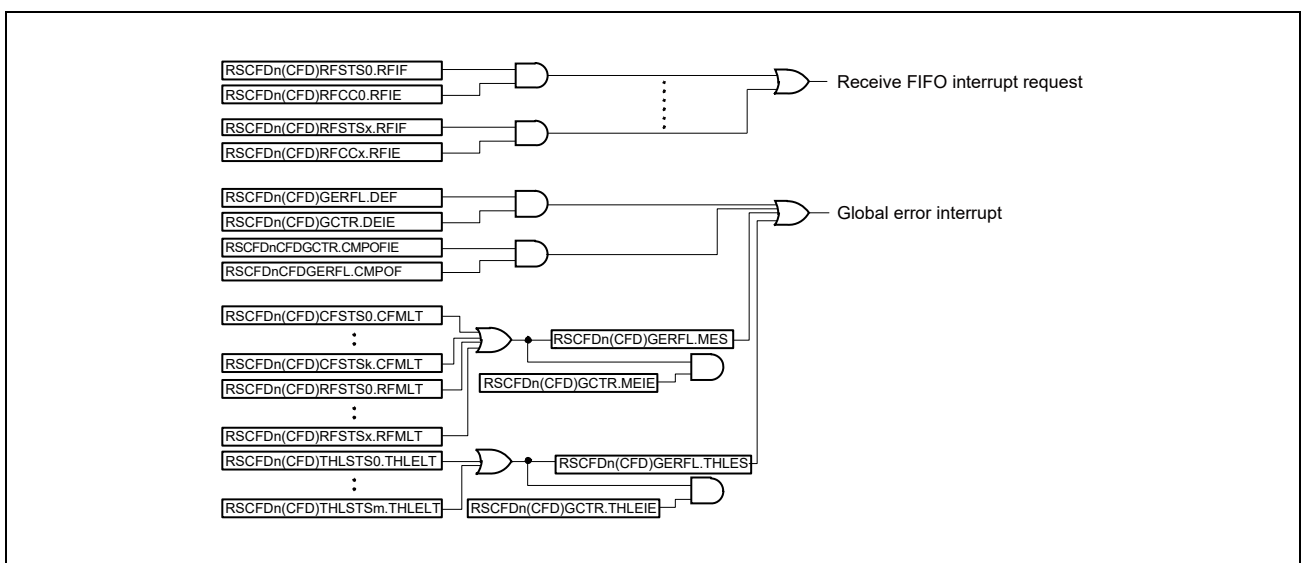


Figure 20.2 CAN Global Interrupt Block Diagram

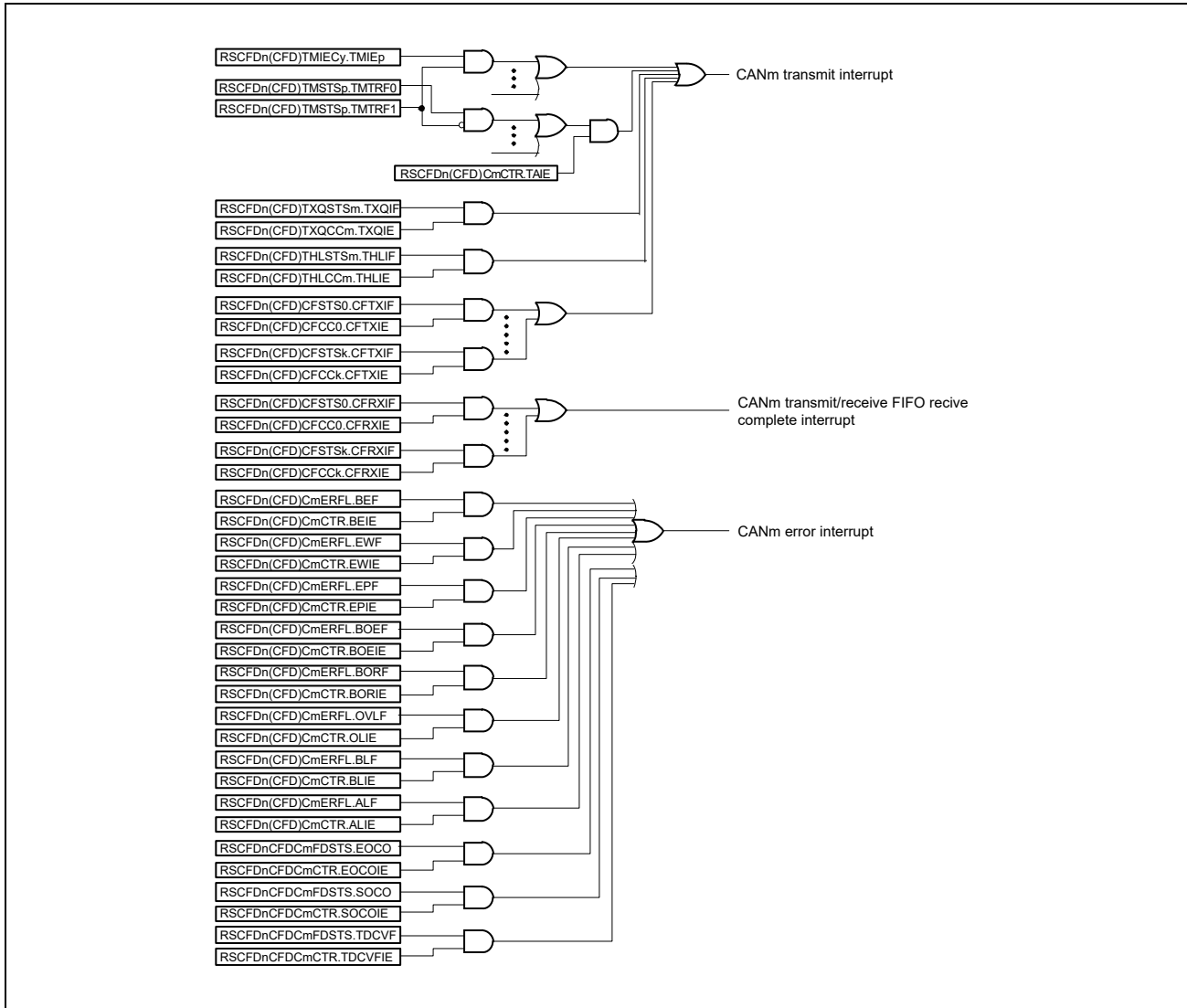


Figure 20.3 CAN Channel Interrupt Block Diagram

20.5.2 DMA Trigger (Only in CAN FD Mode)

In CAN FD mode, receive FIFO buffers can be related to DMA channels. The following 13 FIFO buffers can be related.

- All receive FIFO buffers x ($x =$ see **Table 20.6**)
- The first transmit/receive FIFO buffer k ($k = 3 \times m$, $m =$ see **Table 20.6**) allocated to channel m

When the DMA enable bit (RFDMAEx or CFDMAEm bit in the RSCFDnCFDCDTCT register) is set to 1 and an unread message is remaining in the related FIFO, a DMA transfer request trigger is generated.

20.6 CAN Modes

The RS-CANFD module has four global modes to control entire RS-CANFD module status and four channel modes to control individual channel status. Details of global modes are described in **Section 20.6.1, Global Modes**, and details of channel modes are described in **Section 20.6.2, Channel Modes**.

- Global stop mode: Stops clocks of the entire module to achieve low power consumption.
- Global reset mode: Performs initial settings for the entire module.
- Global test mode: Performs test settings and performs RAM test.
- Global operating mode: Makes the entire module operable.
- Channel stop mode: Stops the channel clock.
- Channel reset mode: Performs initial settings for the channel.
- Channel halt mode: Stops CAN communication and performs channel test.
- Channel communication mode: Performs CAN communication.

20.6.1 Global Modes

Figure 20.4 shows the transitions of global modes.

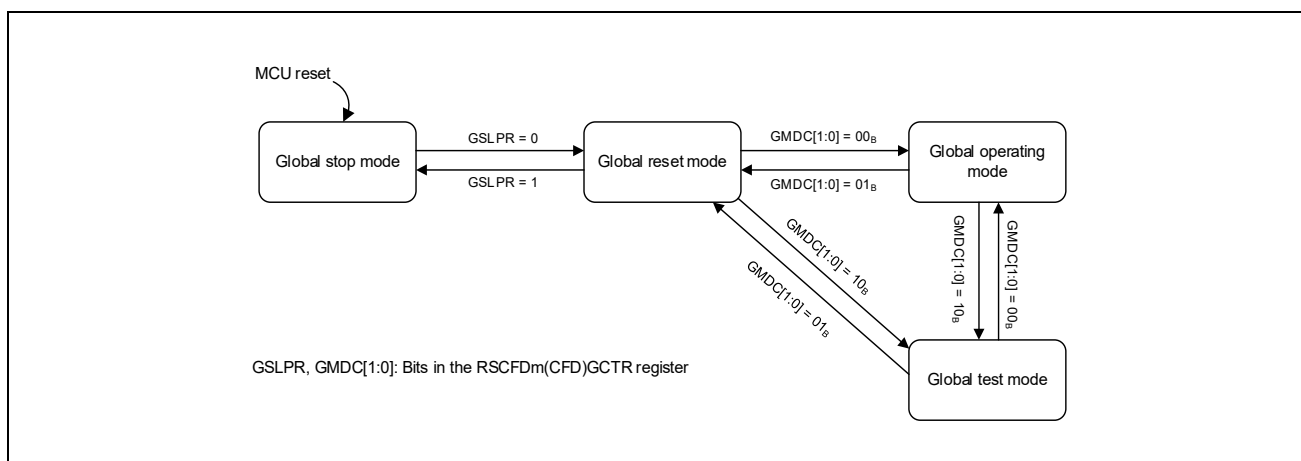


Figure 20.4 Transitions of Global Modes

Channel modes may transition in accordance with transitions of global modes. **Table 20.182** shows the transitions of channel modes depending on the global mode setting (GMDC[1:0] bits and GSLPR bit).

Table 20.182 Transitions of Channel Modes Depending on Global Mode Setting (GMDC[1:0] and GSLPR Bits)

Channel Mode before Setting	Channel Mode after Setting			
	GMDC[1:0] = 00 _B GSLPR = 0 (Global Operating)	GMDC[1:0] = 10 _B GSLPR = 0 (Global Test)	GMDC[1:0] = 01 _B GSLPR = 0 (Global Reset)	GMDC[1:0] = 01 _B GSLPR = 1 (Global Stop)
Channel communication	Channel communication	Channel halt	Channel reset	Transition prohibited
Channel halt	Channel halt	Channel halt	Channel reset	Transition prohibited
Channel reset	Channel reset	Channel reset	Channel reset	Channel stop
Channel stop	Channel stop	Channel stop	Channel stop	Channel stop

Note: GMDC[1:0], GSLPR: Bits in the RSCFDn(CFD)GCTR register.

Table 20.183 shows the transition time between each global mode.

Table 20.183 Transition Time of Global Mode

Mode before Transition	Mode after Transition	Maximum Transition Time
Global stop	Global reset	3 clocks of pclk
Global reset	Global stop	3 clocks of pclk
Global reset	Global test	10 clocks of pclk
Global reset	Global operation	10 clocks of pclk
Global test	Global reset	Two CAN bit times*1, *2
Global test	Global operation	3 clocks of pclk
Global operation	Global reset	Two CAN bit times*1, *2
Global operation	Global test	2 times of CAN frame*1

Note 1. CAN frame time of the channel whose communication speed is lowest among the channels in use.

Note 2. In CAN FD mode, this time value is the CAN bit time of the nominal bit rate.

20.6.1.1 Global Stop Mode

In global stop mode, the CAN clocks do not run and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained. In global stop mode, only the clock for CPU writing to the GSLPR bit is running.

After the MCU is reset, the RS-CANFD module transitions to global stop mode. Setting the GSLPR bit in the RSCFDn(CFD)GCTR register to 1 (in global stop mode) in global reset mode sets the CSLPR bit in each RSCFDn(CFD)CmCTR register to 1 (channel stop mode). If all channels are forcibly caused to transition to channel stop mode, the RS-CANFD module transitions to global stop mode. The GSLPR bit should not be modified in global operating mode or global test mode.

20.6.1.2 Global Reset Mode

In global reset mode, RS-CANFD module settings are performed. When the RS-CANFD module transitions to global reset mode, some registers are initialized. **Table 20.186** and **Table 20.187** list the registers to be initialized.

Setting the GMDC[1:0] bits in the RSCFDn(CFD)GCTR register to 01_B sets the CHMDC[1:0] bits in each of the RSCFDn(CFD)CmCTR register (m = see **Table 20.6**) to 01_B (channel reset mode). If all channels are forcibly caused to transition to channel reset mode, the RS-CANFD module transitions to global reset mode. Channels that are already in channel reset mode or channel stop mode do not transition (because the CHMDC[1:0] bits have already been set to 01_B).

20.6.1.3 Global Test Mode

In global test mode, settings for test-related registers are performed. When the RS-CANFD module transitions to global test mode, all CAN communications are disabled.

Setting the GMDC[1:0] bits in the RSCFDn(CFD)GCTR register to 10_B sets the CHMDC[1:0] bits in each RSCFDn(CFD)CmCTR register to 10_B (channel halt mode). If all channels are forcibly caused to transition to channel halt mode, the RS-CANFD module transitions to global test mode. Channels that are in channel stop mode, channel reset mode, or channel halt mode do not transition.

20.6.1.4 Global Operating Mode

In global operating mode, entire RS-CANFD module operates.

When the GMDC[1:0] bits in the RSCFDn(CFD)GCTR register are set to 00_B, the RS-CANFD module transitions to global operating mode.

20.6.2 Channel Modes

Figure 20.5 shows a channel mode state transition chart. Table 20.184 shows transition time between channel modes.

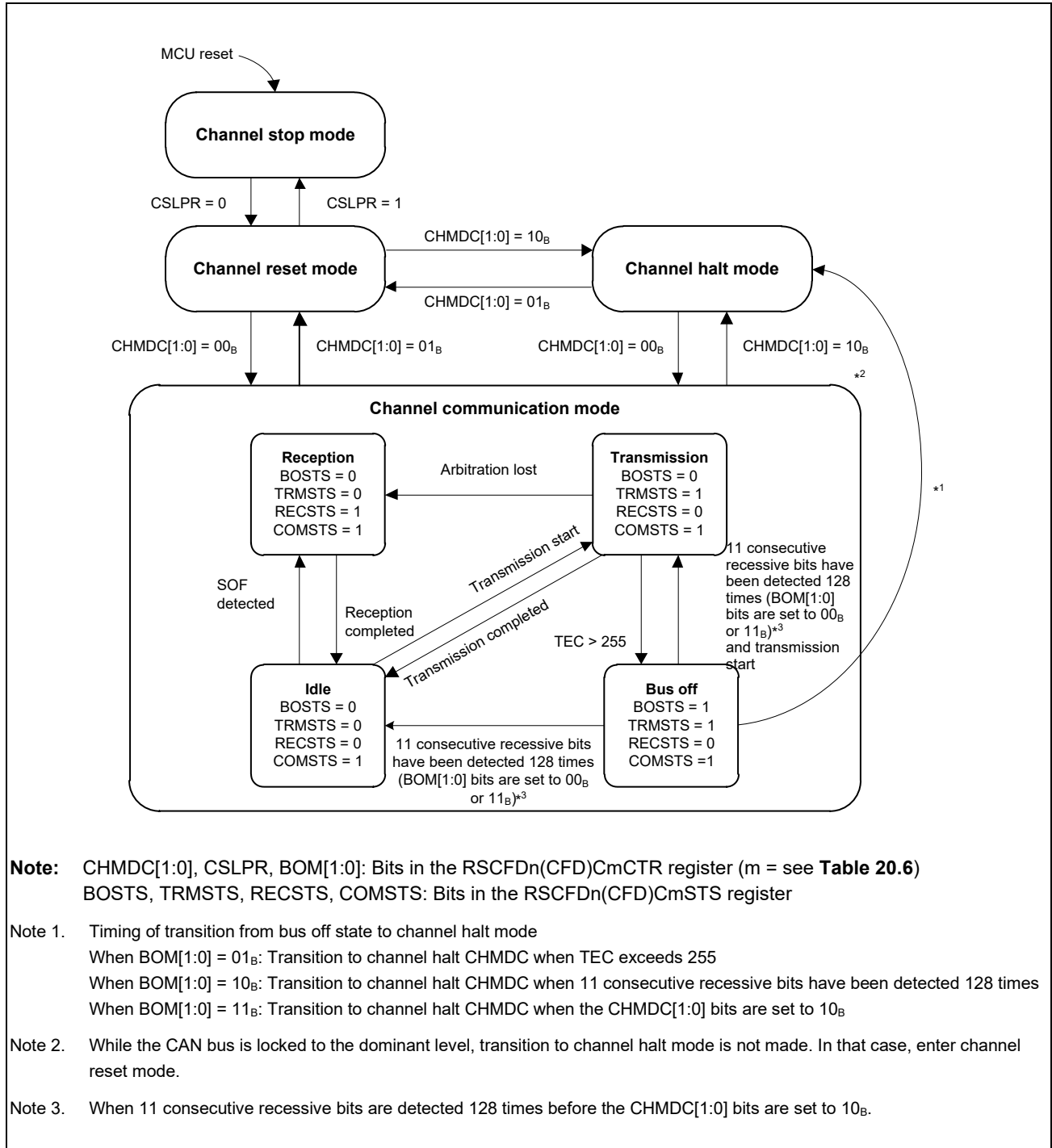


Figure 20.5 Channel Mode State Transition Chart

Table 20.184 Transition Time of Channel Mode (m = see Table 20.6)

Mode before Transition	Mode after Transition	Maximum Transition Time
Channel stop	Channel reset	3 clocks of pclk
Channel reset	Channel stop	3 clocks of pclk
Channel reset	Channel halt	3 CANm bit time* ¹
Channel reset	Channel communication	4 CANm bit time* ¹
Channel halt	Channel reset	2 CANm bit times* ¹
Channel halt	Channel communication	4 CANm bit times* ¹
Channel communication	Channel reset	2 CANm bit times* ¹
Channel communication	Channel halt	2 times of CANm frame

Note 1. In CAN FD mode, this time value is the CANm bit time of the nominal bit rate.

20.6.2.1 Channel Stop Mode

In channel stop mode, clocks are not supplied to channels and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited (except write to CSLPR bit). Register values are retained.

Each channel enters channel stop mode after the MCU is reset. The channel transitions to channel stop mode when the CSLPR bit in the RSCFDn(CFD)CmCTR register (m = see Table 20.6) is set to 1 (channel stop mode) in channel reset mode. The CSLPR bit should not be modified in channel communication mode and channel halt mode.

20.6.2.2 Channel Reset Mode

In channel reset mode, channel settings are performed. When a channel transitions to channel reset mode, some channel-related registers are initialized. Table 20.186 lists the registers to be initialized.

When the CHMDC[1:0] bits in the RSCFDn(CFD)CmCTR register are set to 01_B (channel reset mode) during CAN communication, communication is terminated before it is completed and the channel transitions to channel reset mode. Table 20.185 shows the operation when the CHMDC[1:0] bits are set to 01_B (channel reset mode) during CAN communication.

20.6.2.3 Channel Halt Mode

In channel halt mode, settings for test-related registers of channels are performed. When a channel transitions to channel halt mode, CAN communication of the channel stops.

Table 20.185 shows operation when the CHMDC[1:0] bits are set to 10_B (channel halt mode) during CAN communication.

Table 20.185 Operation When a Channel Transitions to Channel Reset Mode/Channel Halt Mode

Mode	During Reception	During Transmission	Bus Off State
Channel reset (CHMDC[1:0] = "01 _B ")	Transitions to channel reset mode before reception is completed.*1	Transitions to channel reset mode before transmission is completed.*1	Transitions to channel reset mode before bus off recovery is completed.
Channel halt *3 (CHMDC[1:0] = "10 _B ")	Transitions to channel halt mode after reception is completed.*2	Transitions to channel halt mode after transmission is completed.	[When BOM[1:0] = 00 _B] Transitions to channel halt mode (CHMDC[1:0] = 10 _B) only after bus off recovery. [When BOM[1:0] = 01 _B] Transitions to channel halt mode automatically when the condition for transition to bus off state is met. [When BOM[1:0] = 10 _B] Transitions to channel halt mode automatically after bus off recovery is completed. [When BOM[1:0] = 11 _B] Transitions to channel halt mode immediately after the CHMDC[1:0] bits are set to 10 _B before bus off recovery is completed.

Note 1. To allow transition to channel reset mode after communication is completed, set the CHMDC[1:0] bits to 10_B and confirm that communication has been completed and transition to channel halt mode has been made, and then set the CHMDC[1:0] bits to 01_B.

Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode. The CAN bus status can be confirmed with the BLF flag of the RSCFDn(CFD)CmERFL register that becomes 1 when dominant lock is detected.

Note 3. In classical CAN mode, when the transition from channel reset mode to channel halt mode is to be made, set the RSCANnCmCFG register in channel reset mode and then transition to channel halt mode. In CAN FD mode, set the RSCFDnCFDCmNCFG register and the RSCFDnCFDCmDCFG register, and then make a transition.

20.6.2.4 Channel Communication Mode

In channel communication mode, CAN communication is performed. Each channel is in the following communication states during CAN communication.

- Idle : Neither reception nor transmission is in progress.
- Reception : Receiving a message sent from another node.
- Transmission: Transmitting a message.
- Bus off : Isolated from CAN communication.

When the CHMDC[1:0] bits in the RSCFDn(CFD)CmCTR register are set to 00_B, the channel transitions to channel communication mode. After that, when 11 consecutive recessive bits have been detected, the COMSTS flag in the RSCFDn(CFD)CmSTS register (m = see **Table 20.6**) is set to 1 (communication is ready) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

20.6.2.5 Bus Off State

A channel transitions to the bus off state according to the transmit/receive error counter increment/decrement rules of the CAN specifications. How to return from the bus off state is set by the BOM[1:0] bits in the RSCFDn(CFD)CmCTR register.

- When BOM[1:0] = 00_B:
Bus off recovery is compliant with the CAN specification. After 11 consecutive recessive bits have been detected 128 times, a channel returns from the bus off state to the CAN communication ready state (error active state). At that time, the TEC[7:0] and REC[7:0] bits in the RSCFDn(CFD)CmSTS register are initialized to 00_H, the BORF flag in the RSCFDn(CFD)CmERFL register is set to 1 (bus off recovery is detected), and a bus off recovery interrupt request is generated. When the CHMDC[1:0] bits in the RSCFDn(CFD)CmCTR register are set to 10_B (channel halt mode) in the bus off state, the channel transitions to channel halt mode after completion of bus off recovery (11 consecutive recessive bits have been detected 128 times).
- When BOM[1:0] = 01_B:
When a channel transitions to the bus off state, the CHMDC[1:0] bits are set to 10_B and the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H but the BORF flag is not set to 1 and a bus off recovery interrupt request is not generated.
- When BOM[1:0] = 10_B:
When a channel has transitioned to the bus off state, the CHMDC[1:0] bits are set to 10_B. After completion of bus off recovery (11 consecutive recessive bits have been detected 128 times), the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H and the BORF flag is set to 1, and bus off recovery interrupt request is generated.
- When BOM[1:0] = 11_B:
When the CHMDC[1:0] bits are set to 10_B in the bus off state, the channel transitions to channel halt mode before bus off recovery is completed. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H but the BORF flag is not set to 1. And bus off recovery interrupt request is not generated.
However, the BORF flag becomes 1, and bus off recovery interrupt request is generated if a RS-CANFD module transitions to error active state (by detecting 11 consecutive recessive bits 128 times) before CHMDC[1:0] bits are set to 10_B.

If the channel transitions to channel halt mode by the RS-CANFD module simultaneously when the program writes a value to the CHMDC[1:0] bits, writing by the program takes precedence. An automatic transition to channel halt mode when the BOM[1:0] bits are set to 01_B or 10_B is made only when the CHMDC[1:0] bits are 00_B (channel communication mode).

Furthermore, setting the RTBO bit in the RSCFDn(CFD)CmCTR register to 1 allows forcible return from the bus off state. As soon as the RTBO bit is set to 1, the state changes to the error active state. After 11 consecutive recessive bits have been detected, the RS-CANFD module becomes ready for communication. In this case, the BORF flag is not set to 1 and the TEC[7:0] and REC[7:0] bits are initialized to 00_H. Write 1 to the RTBO bit when the BOM[1:0] value is 00_B. Writing 1 to the RTBO bit in a state other than the bus off state is ignored, and the RTBO bit is immediately set to 0.

20.6.3 Initializing Registers by Transition to CAN Mode

Table 20.186 lists bits and flags to be initialized by a transition to channel reset mode. These bits and flags are also initialized by a transition to global reset mode. Furthermore, **Table 20.187** lists bits and flags to be initialized only by a transition to global reset mode.

Table 20.186 Registers Initialized in Global Reset Mode or Channel Reset Mode

Register	Bit / Flag
RSCFDn(CFD)CmCTR register	(ROM), CRCT, CTMS[1:0], CTME, CHMDC[1:0]
RSCFDn(CFD)CmSTS register	CHLTSTS, EPSTS, BOSTS, TRMSTS, RECSTS, COMSTS, (ESIF), REC[7:0], TEC[7:0]
RSCFDn(CFD)CmERFL register	CRCREG[14:0], ADERR, B0ERR, B1ERR, CERR, AERR, FERR, SERR, ALF, BLF, OVLf, BORf, BOEF, EPF, EWF, BEF
RSCFDnCFDCmFDCTR register	EOCCLR, SOCCLR
RSCFDnCFDCmFDSTS register	SOC[7:0], EOC[7:0], SOCO, EOCO, TDCVF, TDCR[6:0]
RSCFDnCFDCmFDCRC register	CRCREG[20:0], SCNT[3:0]
RSCFDn(CFD)CFCCk register	When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFE
RSCFDn(CFD)CFSTSk register	When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFMC[7:0], CFFLL, CFEMP, CFMLT, CFRXIF, CFTXIF
RSCFDn(CFD)CFTISTS register	CFkTXIF
RSCFDn(CFD)TMCP register	TMOM, TMTAR, TMTR
RSCFDn(CFD)TMSTSp register	TMTARM, TMTRM, TMTRF[1:0], TMTSTS
RSCFDn(CFD)TMTRSTSy register	TMTRSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCFDn(CFD)TMTARSTSy register	TMTARSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCFDn(CFD)TMTCASTSy register	TMTCASTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCFDn(CFD)TMTASTSy register	TMTASTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCFDn(CFD)TXQCCm register	TXQE
RSCFDn(CFD)TXQSTSm register	TXQIF, TXQFLL, TXQEMP
RSCFDn(CFD)THLCCm register	THLE
RSCFDn(CFD)THLSTSm register	THLMC[4:0], THLIF, THLELT, THLFLL, THLEMP
RSCFDn(CFD)GTINTSTS0 register	TSIFm, TAIFm, TQIFm, CFTIFm, THIFm (m = 0 to 3)
RSCFDn(CFD)GTINTSTS1 register	TSIFm, TAIFm, TQIFm, CFTIFm, THIFm (m = 4)

Note: Bits and flags in parentheses exist only in registers in CAN FD mode.

Table 20.187 Registers Initialized Only in Global Reset Mode

Register	Bit / Flag
RSCFDn(CFD)GSTS register	GHLTSTS
RSCFDn(CFD)GERFL register	EEF0, EEF1, EEF2, EEF3, EEF4, (CMPOF), THLES, MES, DEF
RSCFDn(CFD)GTSC register	TS[15:0]
RSCFDn(CFD)RMNDy register	RMNSq
RSCFDn(CFD)RFCCx register	RFE
RSCFDn(CFD)RFSTsx register	RFMC[7:0], RFIF, RFMLT, RFFLL, RFEMP
RSCFDn(CFD)CFCCk register	When transmit/receive FIFO buffer is in receive mode: CFE
RSCFDn(CFD)CFSTSk register	When transmit/receive FIFO buffer is in receive mode: CFMC[7:0], CFFLL, CFEMP, CFTXIF, CFRXIF, CFMLT
RSCFDn(CFD)FESTS register	CFkEMP, RFxEMP
RSCFDn(CFD)FFSTS register	CFkFLL, RFxFLL
RSCFDn(CFD)FMSTS register	CFkMLT, RFxMLT
RSCFDn(CFD)RFISTS register	RFxIF
RSCFDn(CFD)CFRISTS register	CFkRXIF
RSCFDnCFDCDTCT register	CFDMAEm, RFDMAEx
RSCFDnCFDCDTSTS register	CFDMASTSm, RFDMASTsX
RSCFDn(CFD)GTSTCFG register	RTMPS[6:0], C0ICBCE, C1ICBCE, C2ICBCE, C3ICBCE, C4ICBCE
RSCFDn(CFD)GTSTCTR register	RTME, ICBCTME

Note: Bits and flags in parentheses exist only in registers in CAN FD mode.

20.7 Reception Functions

There are two reception types.

- Reception by receive buffers:

0 to 80 receive buffers can be shared by all channels. Since messages stored in receive buffers are overwritten at each reception, the latest received data can always be read.
- Reception by receive FIFO buffers and transmit/receive FIFO buffers (receive mode):

Eight receive FIFO buffers can be shared by all channels and three dedicated transmit/receive FIFO buffers are provided for each channel. Messages of the number specified by the buffer depth can be stored in FIFO buffers and can be read sequentially from the oldest.

20.7.1 Data Processing Using the Receive Rule Table

Data processing using the receive rule table allows selected messages to be stored in the specified buffer. Data processing includes acceptance filter processing, DLC filter processing, routing processing, label addition processing, and mirror function processing.

Up to 128 receive rules can be registered per channel and up to “64 × number of channels” receive rules can be registered in the entire module.

(Up to 320 receive rules can be registered in this module that has five channels.)

Set receive rules for each channel. No receive rule can be shared with other channels. If receive rules are not set, no message can be received. **Figure 20.6** illustrates how receive rules are registered, giving an example using 2 channels of RS-CANFD.

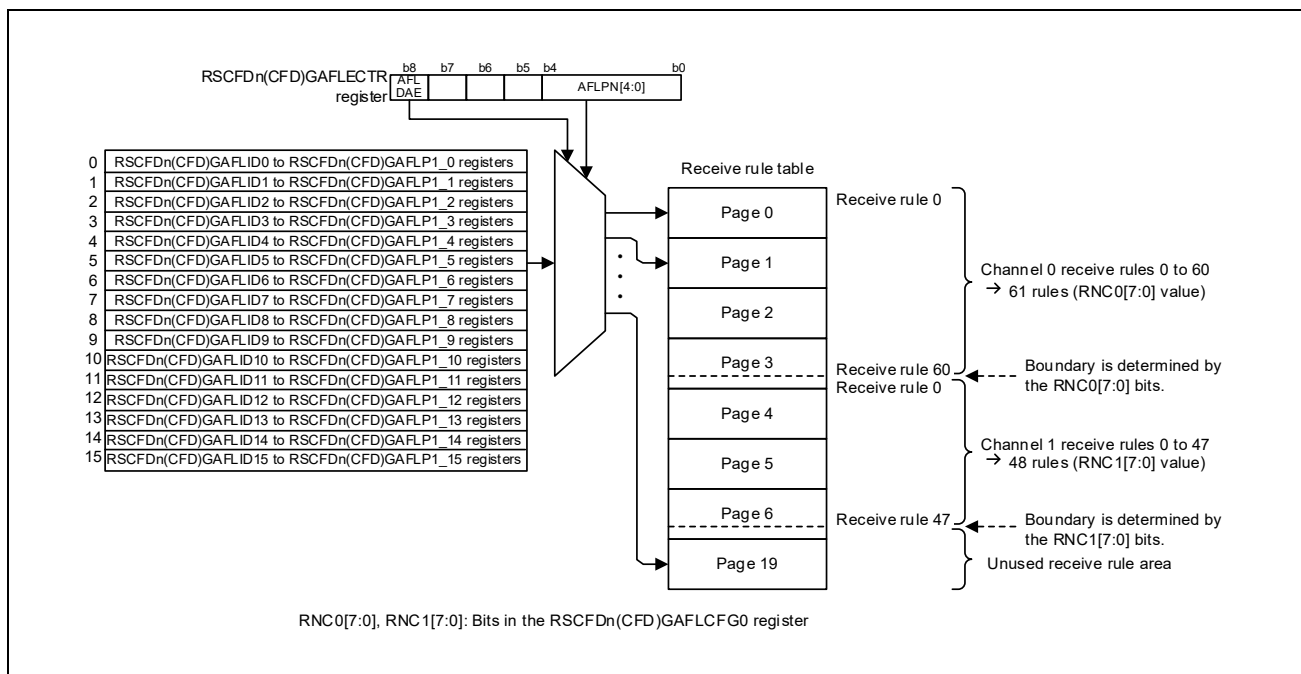


Figure 20.6 Registration of Receive Rules (in the case of setting channel 0 and 1)

CAUTION

Set receive rules of each channel in succession.
 It is not possible to set channel 0 receive rules and channel 1 receive rules alternately.

Each receive rule consists of 16 bytes of the RSCFDn(CFD)GAFLIDj, RSCFDn(CFD)GAFLMj, RSCFDn(CFD)GAFLP0j, and RSCFDn(CFD)GAFLP1j registers (j = see **Table 20.6**). The RSCFDn(CFD)GAFLIDj register is used to set GAFLID, GAFLIDE bit, GAFLRTR bit, and the mirror function. The RSCFDn(CFD)GAFLMj register is used to set mask. The RSCFDn(CFD)GAFLP0j register is used to set label information to be added, DLC value, and destination receive buffer. The RSCFDn(CFD)GAFLP1j register is used to set a destination FIFO buffer. Up to 16 receive rules can be set per page.

20.7.1.1 Acceptance Filter Processing

In the acceptance filter processing, the ID data, IDE bit, and RTR bit in a received message are compared with the ID data, IDE bit, and RTR bit set in the receive rule of the corresponding channel. When all these bits match, the message passes through the acceptance filter processing. Bits set to 0 (bits are not compared) in the RSCFDn(CFD)GAFLMj register are not compared and are regarded as matched.

Check begins with the lowest-numbered receive rule. When all the bits to be compared in a received message match the bits set in the receive rule or when all the receive rules are compared without any match, filter processing stops. If there is no matching receive rule, the received message is not stored in the receive buffer or FIFO buffer.

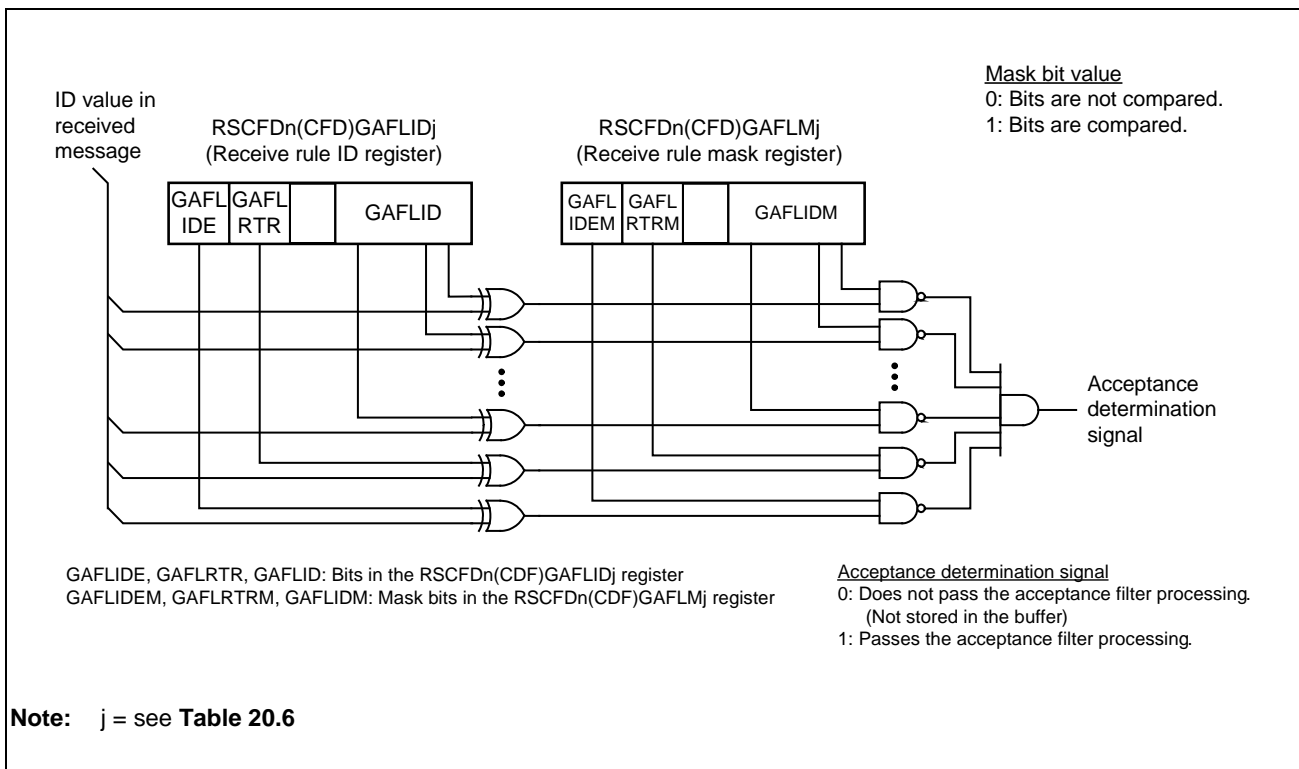


Figure 20.7 Acceptance Filter Function

20.7.1.2 DLC Filter Processing

When the DCE bit in the RSCFDn(CFD)GCFG register is set to 1 (DLC check is enabled), DLC filter processing is performed for messages that pass through the acceptance filter processing. When the DLC value in a message is equal to or larger than the DLC value set in the receive rule, the message passes through the DLC filter processing.

When a message has passed through the DLC filter processing with the DRE bit in the RSCFDn(CFD)GCFG register set to 0 (DLC replacement is disabled), the DLC value of the received message is stored in the buffer. In this case, all the data bytes in the received message are stored in the buffer.

When a message has passed through the DLC filter processing with the DRE bit in the RSCFDn(CFD)GCFG register set to 1 (DLC replacement is enabled), the DLC value of the receive rule is stored in the buffer instead of the DLC value of the received message. In this case, a value of 00_H is written to data bytes that are larger than the DLC value of the receive rule.

When the DLC value of the received message is smaller than that of the receive rule, the message does not pass through the DLC filter processing. In this case, the message is not stored in the receive buffer or the FIFO buffer and the DEF flag in the RSCFDn(CFD)GERFL register is set to 1 (a DLC error is present).

20.7.1.3 Routing Processing

Messages that passed through the acceptance filter processing and the DLC filter processing are stored in receive buffers, receive FIFO buffers, or transmit/receive FIFO buffers set to receive mode or gateway mode. Message storage destination is set by the GAFLRMV and GAFLRMDP[6:0] bits in the RSCFDn(CFD)GAFLP0j register (j = see **Table 20.6**) and by the RSCFDn(CFD)GAFLP1j register. Messages that passed through the acceptance filter processing and the DLC filter processing can be stored in up to 8 buffers.

In CAN FD mode, if the payload length of the received message exceeds the payload storage size of the storage buffer, the CMPOF flag in the RSCFDnCFDGERFL register is set to 1 (payload overflow) and the processing is handled according to the CMPOC bit in the RSCFDnCFDGCFG register. When the CMPOC bit is 0, the received message which exceeds the payload storage size is not stored in the buffer. When the CMPOC bit is 1, the received message is stored in the buffer with payloads exceeding the storage size being discarded, and depending on the DRE bit in the RSCFDnCFDGCFG register the received DLC value or the DLC value of the receive rule is stored in the buffer.

20.7.1.4 Label Addition Processing

It is possible to add 12-bit label information to messages that passed through the filter processing and store them in buffers. This label information is set in the GAFLPTR[11:0] bits of the RSCFDn(CFD)GAFLP0j register.

20.7.1.5 Mirror Function Processing

The mirror function allows reception of messages transmitted from the own CAN node. The mirror function is enabled by setting the MME bit in the RSCFDn(CFD)GCFG register to 1 (mirror function is enabled).

When the mirror function is in use, receive rules for which the GAFLLB bit in the RSCFDn(CFD)GAFLIDj register is set to 0 are applied to the data processing for messages received from other CAN nodes. When messages transmitted from the own CAN node are received, receive rules for which the GAFLLB bit is set to 1 are used for data processing.

20.7.1.6 Timestamp

The timestamp counter is a 16-bit free-running counter used for recording the message reception time and transmission time. The timestamp counter value is fetched at the timing set with the TSCCFG[1:0] bits in the RSCFDn(CFD)GFDCFG register and is then stored in a receive buffer or a FIFO buffer together with the message ID and data during data reception. The clock source of the timestamp counter is selected by the TSBTCS[2:0] and TSSS bits in the RSCFDn(CFD)GCFG register. In classical CAN mode, the clock source is selectable from $pclk/2$ or the CAN m bit time clock ($m =$ see **Table 20.6**). In CAN FD mode, the clock source is selectable from $pclk/2$ or nominal CAN m bit time clock. However, do not select the nominal CAN m bit time clock of channels that handle CAN FD frames. The clock obtained by dividing the selected clock source by the TSP[3:0] value of the RSCFDn(CFD)GCFG register is used as the count source of the timestamp counter.

When the CAN m bit time clock or nominal CAN m bit time clock is used as a clock source, the timestamp counter stops when the corresponding channel transitions to channel reset mode or channel halt mode. When $pclk/2$ is used as a clock source, the timestamp function is not affected by channel mode.

The timestamp counter value is reset to 0000_H by setting the TSRST bit in the RSCFDn(CFD)GCTR register to 1.

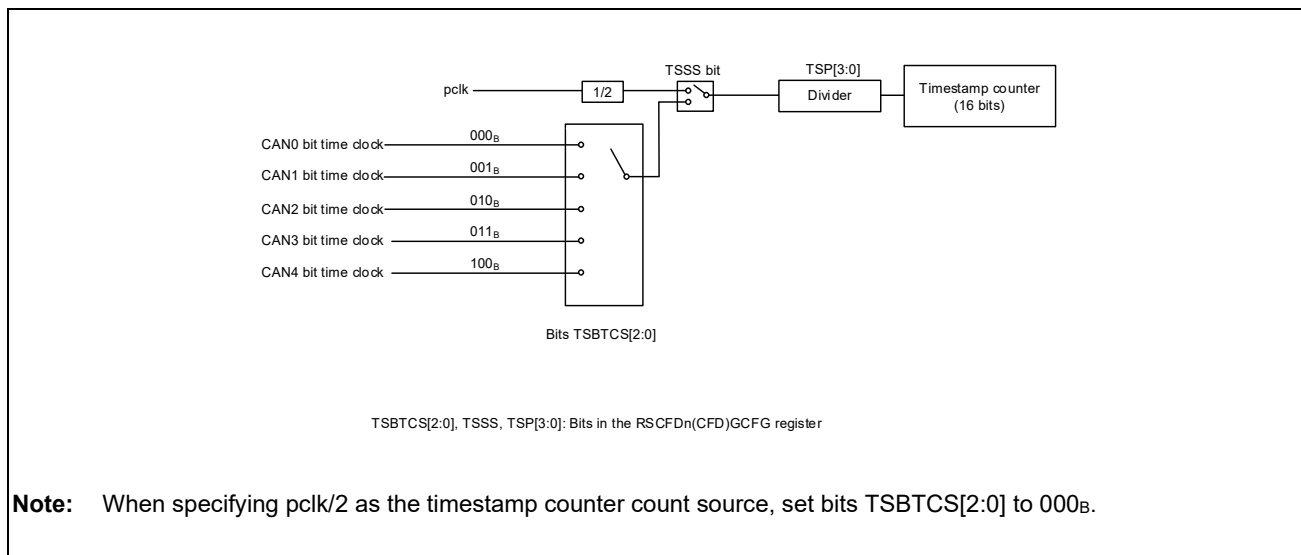


Figure 20.8 Timestamp Function Block Diagram

20.8 Transmission Functions

There are three types of transmission. In classical CAN mode, transmittable payload length is 8 bytes in every transmission type. In CAN FD mode, transmittable payload length varies with transmission types.

- Transmission using transmit buffers:

Each channel has 16 buffers. Transmittable payload length in CAN FD mode is 20 bytes. However, when transmit buffer merge mode is used, four buffers out of 16 buffers are allocated as a payload-only storage area and two buffers are able to transmit payloads with a length of more than 20 bytes.

- Transmission using transmit/receive FIFO buffers (transmit mode):

Each channel has three FIFO buffers. Up to 128 messages can be saved in a single FIFO buffer. Transmittable payload length in CAN FD mode is 64 bytes. Each FIFO buffer is used with a link to a transmit buffer. Only the message to be transmitted next in a FIFO buffer becomes a target of transmit priority determination. Messages are transmitted sequentially on a first-in, first-out basis.

- Transmission using transmit queues:

Up to 16 transmit buffers per channel can be allocated to transmit queues. Transmittable payload length in CAN FD mode is 20 bytes. Transmit buffer number $(16 \times m) + 15$ is used as a common access window. Transmit buffers are allocated to transmit queues in descending order of buffer numbers. All messages in transmit queues, which are targets of priority determination, are transmitted in the order of ID numbers.

Figure 20.9 shows the allocation of transmit queues and transmit/receive FIFO buffer link.

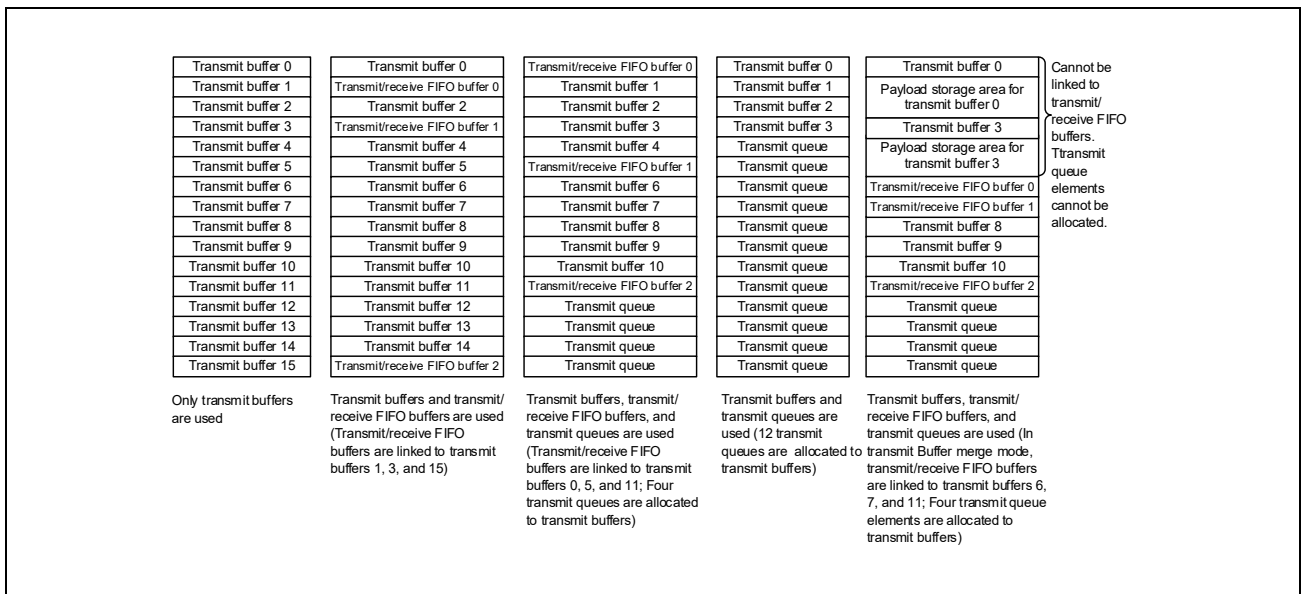


Figure 20.9 Allocation of Transmit Queues and Transmit/Receive FIFO Buffer Links

20.8.1 Transmit Priority Determination

If transmit requests are issued from multiple buffers or queues in the same channel, transmit priority is determined.

The priority is determined by using one of the following methods.

- ID priority (TPRI bit = 0)
- Transmit buffer number priority (TPRI bit = 1)

The setting of the TPRI bit in the RSCFDn(CFD)GCFG register is effective in all CAN channels.

When the TPRI bit is set to 0, messages are transmitted according to the priority of stored message IDs. Priority of IDs conforms to the CAN bus arbitration specification defined in the CAN specification. Targets of priority determination are IDs of messages placed in transmit buffers, transmit/receive FIFO buffers (set to transmit mode or gateway mode), and transmit queues. When one or more transmit queues are used, select the ID priority method. When transmit/receive FIFO buffers are used, the oldest message in a FIFO buffer becomes a target of priority determination. When a message is being transmitted from a transmit/receive FIFO buffer, the next message in the same FIFO buffer becomes a target of priority determination. When transmit queues are used, all messages in transmit queues are targets of priority determination. If the same ID is set for two or more buffers, the buffer with a lower number takes precedence.

When the TPRI bit is set to 1, the message in the transmit buffer whose number is the lowest among buffers having transmit requests are transmitted first. When transmit/receive FIFO buffers are linked to transmit buffers, transmit priority is determined according to the linked transmit buffer numbers.

When messages are retransmitted due to an arbitration-lost or an error, transmit priority determination is made again regardless of the TPRI bit setting. When a 2-bit ECC error is detected in the priority determination processing, no message is transmitted (only when the EEFE bit in the RSCANnGCFG register is 1 in classical CAN mode).

20.8.2 Transmission Using Transmit Buffers

Setting the transmit request bit (TMTR bit in the RSCFDn(CFD)TMCp register) of a transmit buffer to 1 (transmission is requested) allows transmission of data frames or remote frames.

Transmit result is shown by the TMTRF[1:0] flag in the corresponding RSCFDn(CFD)TMSTSp register (p = see **Table 20.6**). When transmit completes successfully, the TMTRF[1:0] flag is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)).

20.8.2.1 Transmit Abort Function

Setting the TMTAR bit in the RSCFDn(CFD)TMCp register to 1 (transmit abort is requested) cancels the transmit request from a transmit buffer for which the TMTRM bit in the RSCFDn(CFD)TMSTSp register is set to 1 (a transmit request is present). When transmit abort is completed, the TMTRF[1:0] flag of the RSCFDn(CFD)TMSTSp register is set to 01_B (transmit abort has been completed) and the transmit request is canceled (clearing the TMTRM bit to 0).

A message that is being transmitted or a message to be transmitted next according to the transmit priority determination cannot be aborted. However, if an arbitration lost or an error has occurred while a message for which the TMTAR bit is set to 1 is being transmitted, retransmission is not performed.

20.8.2.2 One-Shot Transmission Function (Retransmission Disabling Function)

When the TMOM bit of the RSCFDn(CFD)TMCp register is set to 1 (one-shot transmission is enabled), transmission is performed only once. Even if an arbitration lost or an error occurs, retransmission is not performed.

One-shot transmission result is shown by the TMTRF[1:0] flag in the corresponding RSCFDn(CFD)TMSTSp register. When one-shot transmission completes successfully, the TMTRF[1:0] flag is set to 10_B or 11_B. When an arbitration lost or an error has occurred, the TMTRF[1:0] flag is set to 01_B (transmit abort has been completed).

20.8.2.3 Transmit Buffer Merge Mode (Only in CAN FD Mode)

Transmit buffers can transmit messages with a payload length of 20 bytes, but can transmit messages with a payload length of up to 64 bytes by merging three transmit buffers in transmit buffer merge mode.

Setting the TMME bit to 1 in the RSCFDnCFDCmFDCFG register enables transmit buffer merge mode. In this mode, six buffers per channel become a merge area and two sets of transmit buffers ($16 \times m + 0$ to $16 \times m + 2$ and transmit buffers $16 \times m + 3$ to $16 \times m + 5$ are merged. A transmission request is made by the first transmit buffer, and subsequent two buffers are used as a payload storage area. Do not set the transmission request bit (TMTR bit in the RSCFDnCFDTMCp register) and the transmission abort request bit (TMTAR bit in the RSCFDnCFDTMCp register) to 1 for transmit buffers except for the first buffer.

While transmit buffer merge mode is enabled, do not link the transmit/receive FIFO buffer to six merged buffers or allocate it to the transmit queue.

20.8.3 Transmission Using FIFO Buffers

A single transmit/receive FIFO buffer can save as many messages as specified by the CFDC[2:0] bits of the RSCFDn(CFD)CFCCk register (k = see **Table 20.6**). Messages are transmitted sequentially on a first-in, first out basis.

Each Transmit/receive FIFO buffers are linked to transmit buffers selected by the CFTML[3:0] bits in the RSCFDn(CFD)CFCCk register. When the CFE bit of the RSCFDn(CFD)CFCCk register is set to 1 (transmit/receive FIFO buffers are used), transmit/receive FIFO buffers become targets of transmit priority determination. Priority determination is applied only to the message to be transmitted next in a FIFO buffer.

When the CFE bit is set to 0 (no transmit/receive FIFO buffer is used), the CFEMP flag is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) at the timing below.

- The transmit/receive FIFO buffer becomes empty immediately when the message in it is not being transmitted or is not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, arbitration-lost or the transition to channel halt mode in the case that a message in it is being transmitted or to be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages cannot be stored in FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

20.8.3.1 Interval Transmission Function

To transmit messages continuously from the same transmit/receive FIFO buffer that is set to transmit mode or gateway mode, message transmission interval time can be set.

Immediately after the first message has been transmitted successfully from the FIFO buffer with the CFE bit set to 1 in the RSCFDn(CFD)CFCCk register, the interval timer starts counting (after EOF7 of the CAN protocol). After that, when the interval time has passed, the next message is transmitted. The interval timer stops in channel reset mode or by clearing the CFE bit to 0.

The interval time is set by the CFITT[7:0] bits in the RSCFDn(CFD)CFCCk register. When the interval timer is not used, set the CFITT[7:0] bits to 00_H.

Select an interval timer count source using the CFITR and CFITSS bits in the RSCFDn(CFD)CFCCk register. When the CFITR and CFITSS bits are set to 00_B, the count source is obtained by dividing $pclk/2$ by the value of the ITRCP[15:0] bits. When the CFITR and CFITSS bits are set to 10_B, the count source is obtained by dividing $pclk/2$ by (the value of the ITRCP[15:0] bits $\times 10$). When the CFITR and CFITSS bits are set to x1_B, the CANm bit time clock becomes a count source in classical CAN mode and the nominal CANm bit time clock becomes a count source in CAN FD mode. (Use this count source only for the channel does not handle the CAN FD frames in CAN FD mode).

The interval time is calculated by the following formulae where M is the ITRCP[15:0] value and N is the CFITT[7:0] value.

- When CFITR and CFITSS bits are 00_B

$$\frac{1}{pclk \text{ frequency}} \times 2 \times M \times N$$

- When CFITR and CFITSS bits are 10_B

$$\frac{1}{pclk \text{ frequency}} \times 2 \times M \times 10 \times N$$

- When CFITR and CFITSS bits are x1_B

Classical CAN mode: $\frac{1}{\text{CANm bit time clock frequency}} \times N$

CAN FD mode: $\frac{1}{\text{Nominal CANm bit time clock frequency}} \times N$

Figure 20.10 shows the interval timer block diagram.

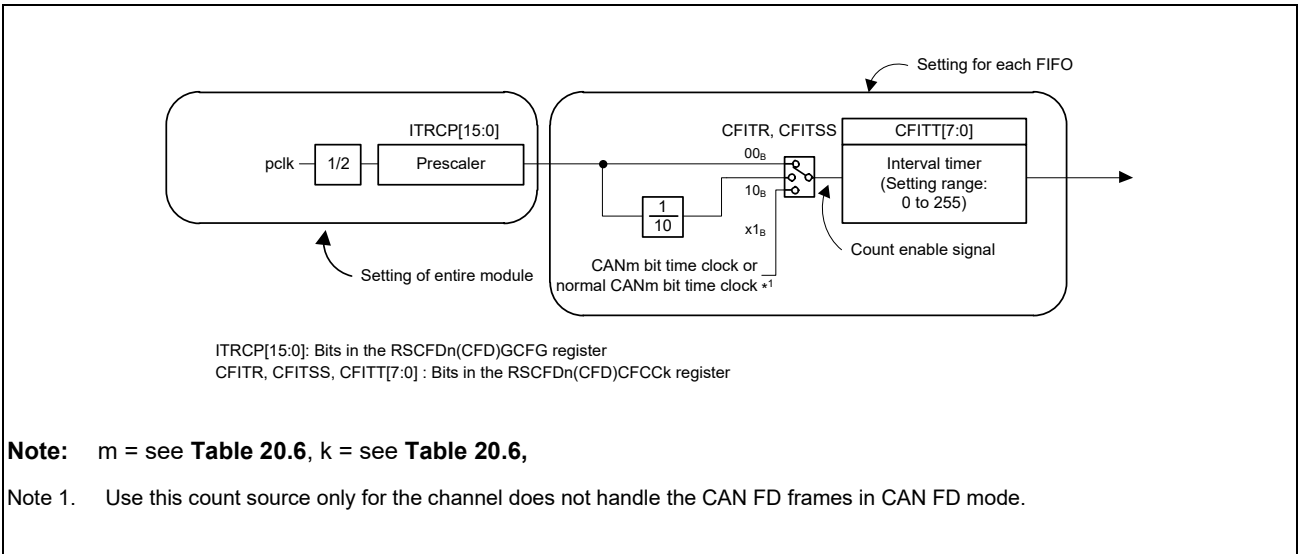


Figure 20.10 Interval Timer Block Diagram

Figure 20.11 shows the interval timer timing chart.

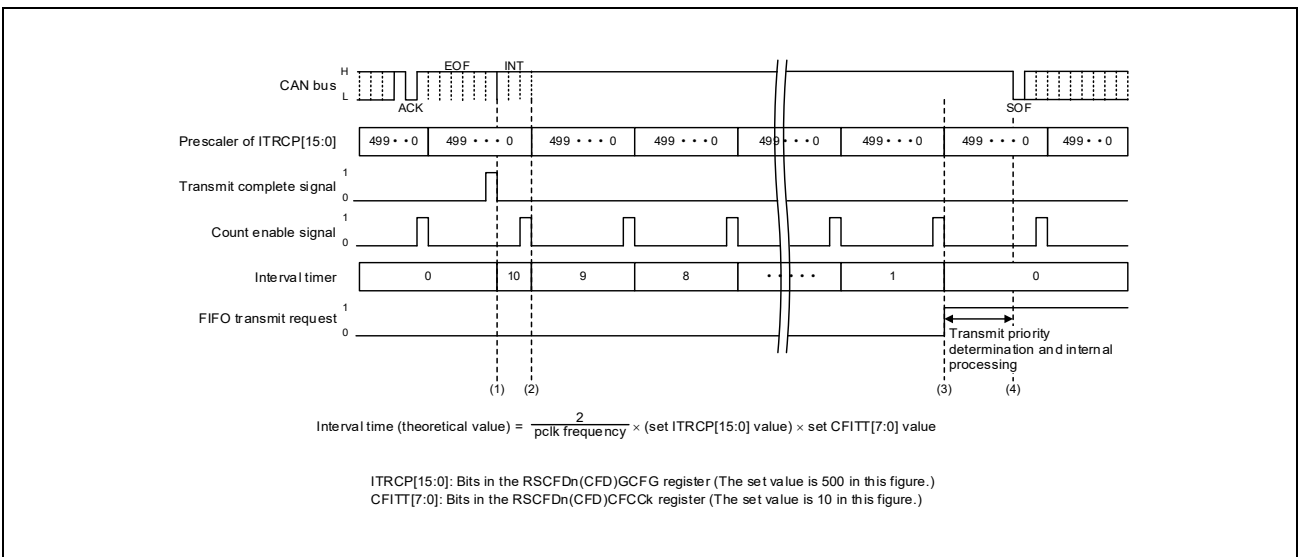


Figure 20.11 Interval Timer Timing Chart

1. The interval timer starts counting upon completion of transmission. Since the prescaler is not initialized at the time of transmission completion, the first interval time has a margin of error of up to one interval timer count.
2. The interval timer is decremented by 1 upon the next count enable signal.
3. When the interval timer has decreased to 0, the transmit/receive FIFO buffer issues a transmit request.
4. When the transmit/receive FIFO buffer is determined for the next transmission by the priority determination, it starts transmitting data. Transmission starts usually with a delay of three CANm bit time clock cycles or less from the issue of transmit request.
 Transmission start may delay up to 1164 pclk clock cycles if several internal processing including receive filter processing, message routing, and transmit priority determination is performed at all channels.

20.8.4 Transmission Using Transmit Queues

Three to sixteen buffers (up to 10 buffers in transmit buffer merge mode) are allocated to a transmit queue for each channel, and the transmit buffer $(16 \times m) + 15$ is used as a common access window.

All messages in a transmit queue are targets of transmit priority determination and are transmitted in the ID priority order regardless of the storing order. If two messages having the same ID are stored in a transmit queue, these messages are not always be transmitted in the storing order in the transmit queue.

Setting the TXQE bit in the RSCFDn(CFD)TXQCCm register to 0 disables transmit queues. When the TXQE bit is set to 0, the TXQEMP flag in the RSCFDn(CFD)TXQSTSm register is set to 1 (the transmit queue contains no message (transmit queue empty)) at the timing below.

- The transmit queue becomes empty immediately when the message in it is not being transmitted or is not to be transmitted next.
- The transmit queue becomes empty after transmission completion, CAN bus error detection, or arbitration lost when the message in it is being transmitted or to be transmitted next.

When the TXQE bit is cleared to 0, all messages in transmit queues are lost and messages cannot be stored in transmit queues. Confirm that the TXQEMP flag is set to 1 before setting the TXQE bit to 1 again.

20.8.5 Transmit Data Padding (Only in CAN FD Mode)

When the payload length indicated by the set DLC value in a transmit message exceeds the payload storage area size of a buffer to be used for transmission, excessive payloads are padded by CC_H .

This processing is performed in the following cases when the transmit buffer merge mode is disabled (TMME bit in the RSCFDnCFDCmFDCFG register is 0).

- Transmit/receive FIFO set to transmission or gateway mode:

When the payload length of the transmit DLC exceeds the transmit/receive FIFO payload storage area size set by the CFPLS[2:0] bits in the RSCFDnCFDCFCCK register

- Transmit buffer (including transmit queue):

When the payload length of the transmit DLC exceeds 20 bytes

When the transmit buffer merge mode is enabled, no transmit data is padded in any transmission using a transmit buffer, transmit/receive FIFO buffer, or transmit queue. At this time, do not set a payload length more than the payload storage size of the buffer for transmitting as the DLC value in the transmit message.

20.8.6 Transmit History Function

Information of transmitted messages can be stored in the transmit history buffer. Each channel has a single transmit history buffer that can contain 16 sets of transmit history data.

A message transmit source buffer type can be selected by the THLDTE bit in the RSCFDn(CFD)THLCCm register. Whether to store transmit history data for each message can be set by the THLEN bit in the RSCFDn(CFD)CFIDk register (k = see **Table 20.6**).

In classical CAN mode, the TMTSCE bit in the RSCANnGCFG register can be used to set whether to include a timestamp value in the transmit history data. In CAN FD mode, a timestamp value is always included.

The following information on a transmitted message will be stored in the transmit history buffer after the successful completion of transmission.

Storage of the transmit history data after the successful completion of transmission may take up to 152 cycles of pclk in classical CAN mode or 420 pclk cycles in CAN FD mode.

- Buffer type

001_B: Transmit buffer

010_B: Transmit/receive FIFO buffer

100_B: Transmit queue

- Buffer number

Number assigned to source transmit buffer, transmit queue, or transmit/receive FIFO buffer. This number depends on buffer types. See **Table 20.188**.

- Label data

Label information of transmitted message

- Timestamp

Value of the transmit message (When the TMTSCE bit is 1 in classical CAN mode)

Table 20.188 Transmit History Data Buffer Numbers

Buffer type	001 _B	010 _B	100 _B
Buffer No.			
0000 _B	Transmit buffer 16 × m + 0	Numbers of transmit buffers linked to transmit/receive FIFO buffers by the CFTML[3:0] bits in the RSCFDn(CFD)CFIDk register (k = see Table 20.6)	Numbers of transmit buffers allocated to the transmit queue that performed transmission
0001 _B	Transmit buffer 16 × m + 1		
0010 _B	Transmit buffer 16 × m + 2		
0011 _B	Transmit buffer 16 × m + 3		
0100 _B	Transmit buffer 16 × m + 4		
0101 _B	Transmit buffer 16 × m + 5		
0110 _B	Transmit buffer 16 × m + 6		
0111 _B	Transmit buffer 16 × m + 7		
1000 _B	Transmit buffer 16 × m + 8		
1001 _B	Transmit buffer 16 × m + 9		
1010 _B	Transmit buffer 16 × m + 10		
1011 _B	Transmit buffer 16 × m + 11		
1100 _B	Transmit buffer 16 × m + 12		
1101 _B	Transmit buffer 16 × m + 13		
1110 _B	Transmit buffer 16 × m + 14		
1111 _B	Transmit buffer 16 × m + 15		

Label data is used to identify each message. A unique label data can be added to each message transmitted from a transmit buffer, transmit queue, or transmit/receive FIFO buffer.

The timestamp value is fetched from the timestamp counter at the SOF (start of frame) timing of the message. For details about the timestamp counter, see **Section 20.7.1.6, Timestamp**.

Transmit history data can be read from the RSCFD_n(CFD)THLACC_m register. If it is attempted to store new transmit history data while the buffer is full, the buffer overflows and the new data is discarded.

20.9 Gateway Function

When a transmit/receive FIFO buffer is set to gateway mode, received messages can be transmitted from an arbitrary channel without CPU intervention.

When a transmit/receive FIFO buffer for which the CFM[1:0] bits of the RSCFDn(CFD)CFCCk register are set to 10_B (gateway mode) is selected by the RSCFDn(CFD)GAFLP1j register, messages that passed through the filter processing of the receive rule are stored in the specified transmit/receive FIFO buffer and are automatically transmitted from the buffer.

Messages stored in a transmit/receive FIFO buffer are transmitted sequentially on a first-in, first-out basis. Only the message to be transmitted next becomes a target of transmit priority determination.

Transmit/receive FIFO buffers in the gateway mode are disabled by setting the CFE bit in the RSCFDn(CFD)CFCCk register to 0. When the CFE bit is set to 0, the CFEMP flag becomes 1 at the timing below.

- The transmit/receive FIFO buffer becomes empty immediately when the message in it is not being transmitted or is not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, or arbitration lost when the message in it is being transmitted or to be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages can no longer be stored in transmit/receive FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

20.9.1 CAN-CAN FD Gateway (Only in CAN FD Mode)

When the gateway function is to be used in CAN FD mode, frames for transmission can be replaced with classical CAN frames or CAN FD frames.

The CAN-CAN FD gateway is enabled when the GWEN bit in the RSCFDnCFDCmFDCFG register is set to 1. The GWFDF and GWBRS bits in the RSCFDnCFDCmFDCFG register can be used to select the FDF and BRS bits of the transmission frame. If the DLC value of the received CAN frame is 1001_B or above and the setting of the GWFDF bit is 1 (CAN FD frame), the DLC value is replaced with 1000_B.

While the CAN-CAN FD gateway is enabled, do not route the following frames.

- CAN FD frames with a payload length of more than 8 bytes
- Remote frames

Also, while the CAN-CAN FD gateway is enabled, transmit only the following frames from the corresponding channel by setting GWFDF.

- When GWFDF = 0, transmit only classical CAN frames.
- When GWFDF = 1, transmit only CAN FD frames.

20.10 Test Function

The test function is classified into communication tests and global tests.

- Communication tests: Performed for each channel.
 - Standard test mode
 - Listen-only mode
 - Self-test mode 0 (external loopback mode)
 - Self-test mode 1 (internal loopback mode)
 - Restricted operation mode (only in CAN FD mode)

- Global tests: Performed in entire module
 - RAM test (read/write test)
 - Inter-channel communication test (CRC error test enabled)

20.10.1 Standard Test Mode

Standard test mode allows CRC test. The CRC value calculated by the RS-CANFD module based on the transmit message or receive message is stored in the register. This CRC value is stored in the CRCREG[14:0] bits in the RSCFDn(CFD)CmERFL register when the message is a classical CAN frame (CRC length = 15 bits) or in the CRCREG[20:0] bits in the RSCFDnCFDCmFDCRC register when the message is a CAN FD frame (CRC length = 17 or 21 bits). Use the inter-channel communication test function for CRC error tests. For details, see **Section 20.10.6.1, CRC Error Test**.

20.10.2 Listen-Only Mode

Listen-only mode allows reception of data frames and remote frames. Only recessive bits are transmitted on the CAN bus, and the ACK bit, overload flag, and active error flag are not transmitted.

Listen-only mode is available for detecting the communication speed.

Do not make a transmit request from any buffer or queue in listen-only mode.

Figure 20.12 shows the connection when listen-only mode is selected.

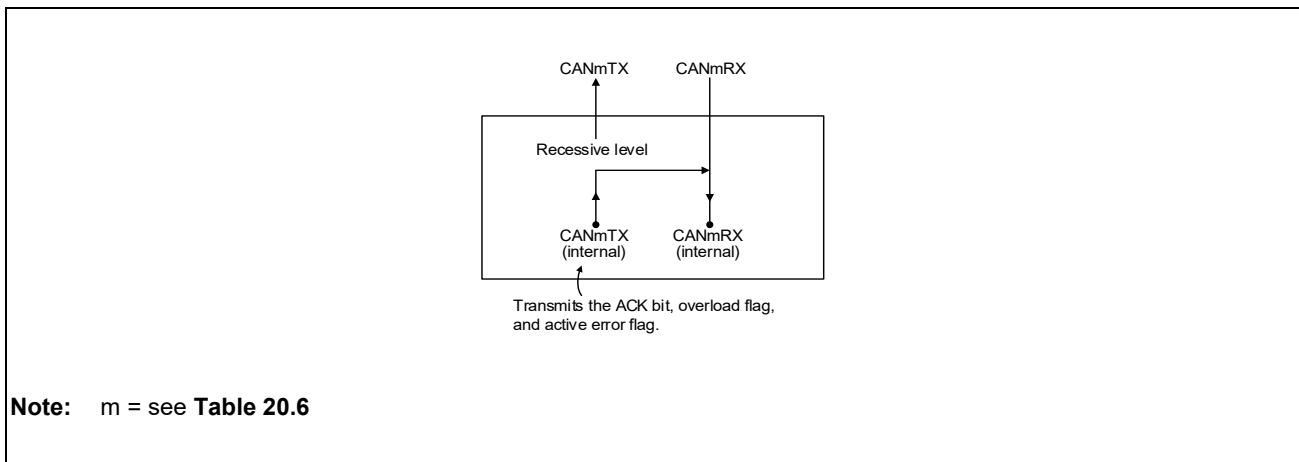


Figure 20.12 Connection When Listen-Only Mode is Selected

20.10.3 Self-Test Mode (Loopback Mode)

In self-test mode, messages transmitted from a channel are compared with the receive rule of the same channel. Messages are stored in a buffer if they have passed through the filter processing. Messages transmitted from other CAN nodes are compared only with the receive rule for which the GAFLLB bit in the RSCFDn(CFD)GAFLIDj register (j = see **Table 20.6**) is set to 0 (when a message transmitted from another CAN node is received).

If the mirror function and self-test mode are both enabled, the self-test mode setting takes precedence.

20.10.3.1 Self-Test Mode 0 (External Loopback Mode)

Self-test mode 0 is used to perform a loopback test within a channel including the CAN transceiver.

In self-test mode 0, transmitted messages are handled as messages received through the CAN transceiver and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

Figure 20.13 shows the connection when self-test mode 0 is selected.

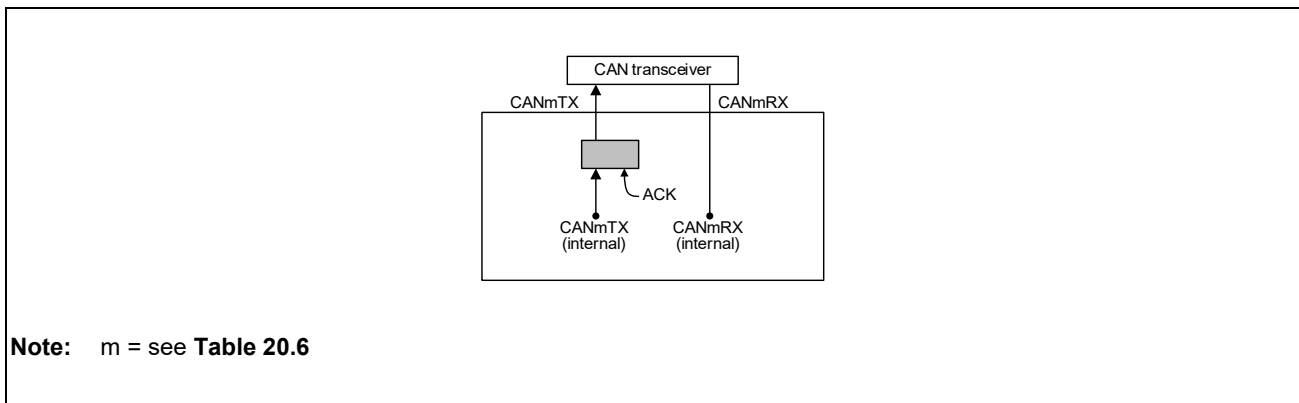


Figure 20.13 Connection When Self-Test Mode 0 is Selected

20.10.3.2 Self-Test Mode 1 (Internal Loopback Mode)

In self-test mode 1, transmitted messages are handled as received messages and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

In self-test mode 1, internal feedback from the internal CANmTX pin ($m =$ see **Table 20.6**) to the internal CANmRX pin is performed. The external CANmRX pin input is isolated. The external CANmTX pin outputs only recessive bits.

Figure 20.14 shows the connection when self-test mode 1 is selected.

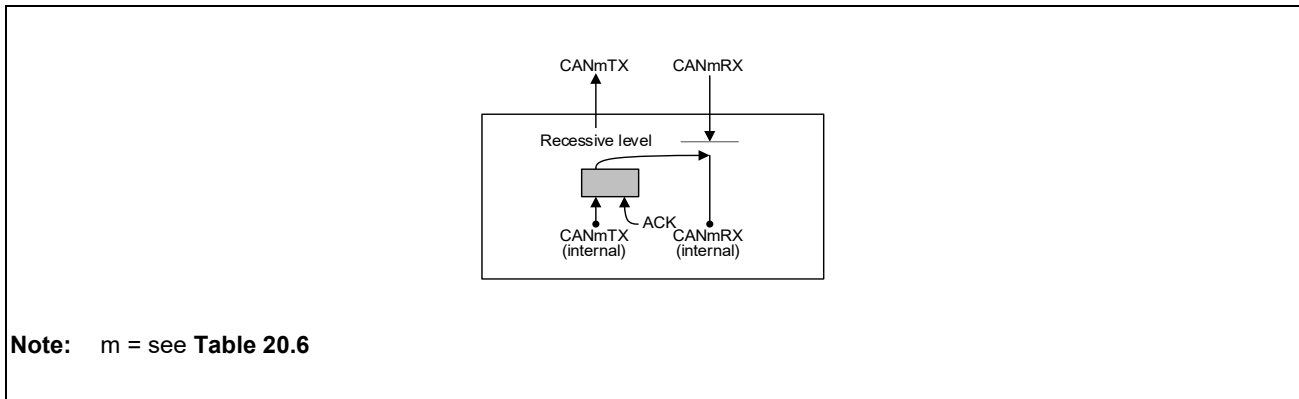


Figure 20.14 Connection When Self-Test Mode 1 is Selected

20.10.4 Restricted Operation Mode (Only in CAN FD Mode)

In restricted operation mode, an ACK bit is generated when a valid data frame and a remote frame have been received, but these frames are not transmitted even if an error frame or an overload frame transmit condition is detected. When a condition is detected, operation is suspended until the bus idle state comes for resynchronization with the CAN communication. The receive error counter (REC) and the transmit error counter (TEC) do not change due to an error.

A desired transmission request can be made for transmission without restrictions.

20.10.5 RAM Test Mode

The RAM test function allows accesses to all CAN RAM addresses. RAM initialization which is performed after resetting the MCU does not initialize all CAN RAM areas.

When the RAM test function is used, the RAM is divided into pages of 256 bytes each. A RAM test page is selected by the RTMPS[6:0] bits in the RSCFDn(CFD)GTSTCFG register. Data in the set page can be read from and written to the RSCFDn(CFD)RPGACCr register (r = see **Table 20.6**).

The available total RAM size is shown below.

In classical CAN mode:

12640 bytes (3160_H)

In CAN FD mode:

17760 bytes (4560_H)

In CAN FD mode, do not access more than 96 bytes in the last page (RTMPS=45_H) during RAM test.

20.10.6 Inter-Channel Communication Test

The inter-channel communication test function allows communication test by internally and mutually connecting CAN channels. During this test, channels are isolated from the external CAN bus.

Before starting data transmission/reception in channel communication mode, make transmission/reception settings for each channel. Channels not included in the test must be placed in channel halt mode.

Figure 20.15 shows the connection for inter-channel communication test.

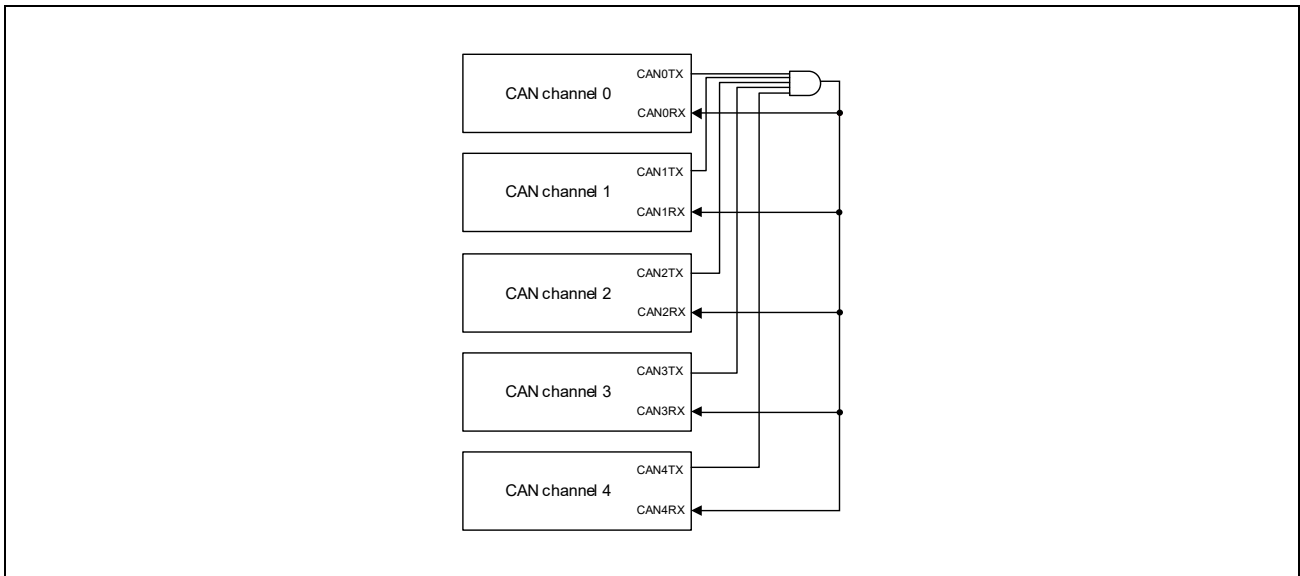


Figure 20.15 Connection for Inter-Channel Communication Test

20.10.6.1 CRC Error Test

A CRC error test is enabled during an inter-channel communication test. The following shows an example of channel 0 CRC error test procedure during a communication test between channel 0 and channel 1.

Preconditions

- Inter-channel communication test is enabled.
- Channel 0 and channel 1 are in standard test mode.

Procedure

1. Make a setting to send a message from the transmit buffer p of channel 1.
2. Set the CRCT bit in the RSCFDn(CFD)C0CTR register to 1 (to enable inversion of the first bit in the received ID field).
3. Set the TMTR bit in the RSCFDn(CFD)TMCp register to 1 (to issue a transmission request to the transmit buffer p of channel 1).
4. Wait for occurrence of a CAN0 error interrupt due to a channel bus error.
5. Read the CRCREG[14:0] bits in the RSCFDn(CFD)CmERFL register or the CRCREG[20:0] bits in the RSCFDnCFDCmFDCRC register of channel 0 and channel 1, and confirm that the CRC values are different on the transmission and the reception side.
6. Confirm that the CERR bit in RSCFDm(CFD)C0ERFL is 1 (CRC error detected).

The CRC error test function generates an incorrect CRC value by inverting the first bit in the received ID field. Therefore, note that not a CRC error but a stuff error (continuous 6-bit data of the same level) is detected when a message in which ID's upper 5-bit value is 10000_B or ID's upper 6-bit value is 011111_B is received.

The CRC generation circuit of the RS-CANFD module is contained in the protocol controller of each channel. Another CRC calculation test is not necessary during transmission because the same circuit is used for both transmission and reception.

20.11 RS-CANFD Setting Procedure

20.11.1 Initial Settings

The RS-CANFD module initializes the CAN RAM after the MCU is reset. The RAM initialization time is 6322 μs cycles.

The GRAMINIT flag in the RSCFDn(CFD)GSTS register is set to 1 (CAN RAM initialization is ongoing) during the RAM initialization and is cleared to 0 (CAN RAM initialization is finished) when the initialization is completed.

Configure CAN settings after the GRAMINIT flag is cleared to 0. **Figure 20.16** shows the CAN setting procedure after the MCU is reset.

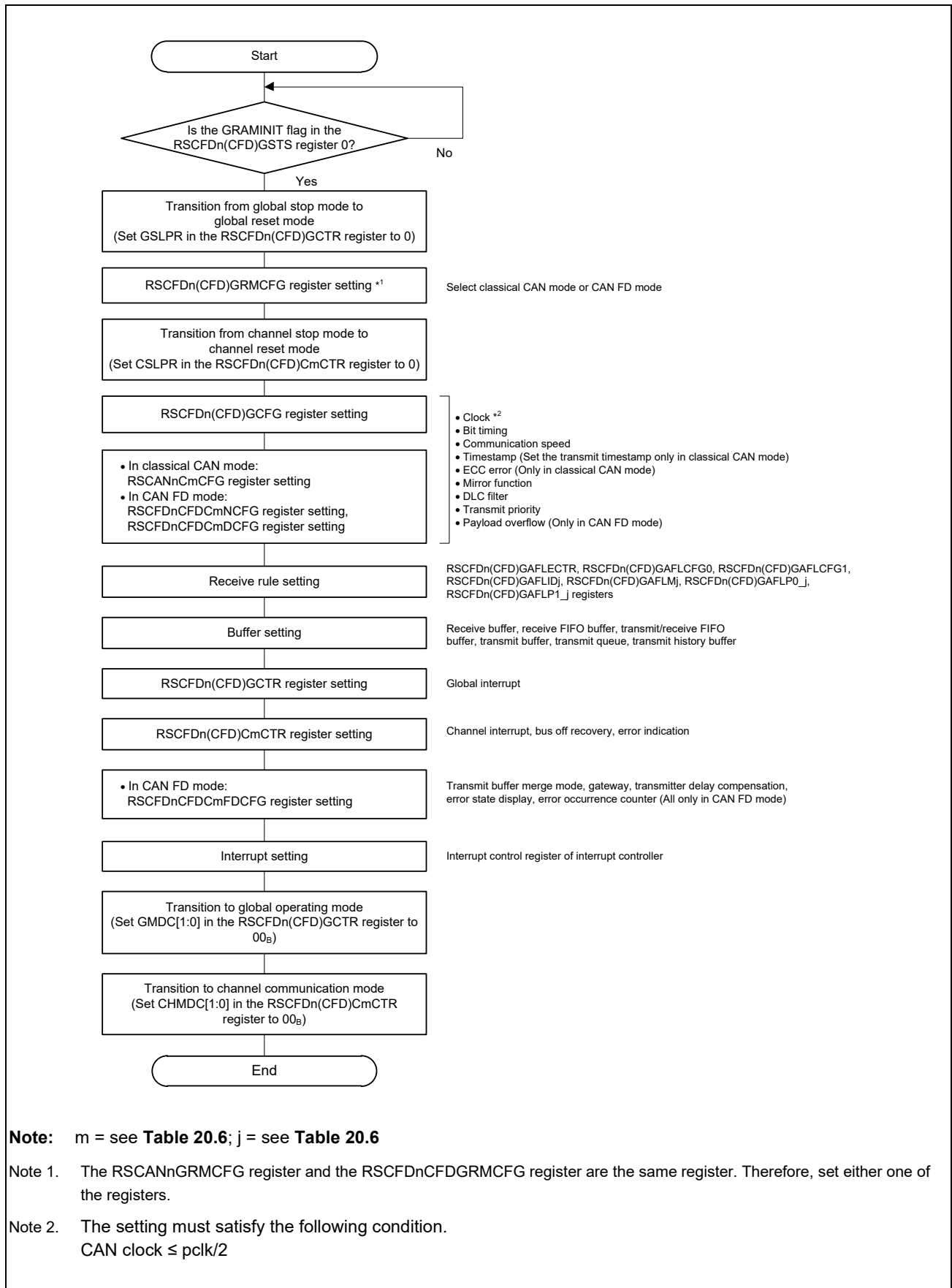


Figure 20.16 CAN Setting Procedure after the MCU is Reset

20.11.1.1 Clock Setting

Set the CAN clock (fCAN) which is the clock source of the RS-CANFD module. Select clk_c or clk_{xincan} by the DCS bit in the RSCFDn(CFD)GCFG register.

20.11.1.2 Bit Timing Setting

In the CAN protocol, one bit of a communication frame consists of three segments SS, TSEG1, and TSEG2, of which two segments TSEG1 and TSEG2 can be set by the corresponding registers for each channel. In classical CAN mode, set these two segments in the RSCANn_{Cm}CFG register. Two bit rates (nominal bit rate and data bit rate) are provided for CAN FD mode. Set the nominal bit rate in the RSCFDnCFDCmNCFG register and set the data bit rate in the RSCFDnCFDCmDCFG register.

Sample point timing can be determined by setting these two segments. This timing can be adjusted in units of 1 Time Quantum (hereafter referred to as T_q). A single T_q is the cycle of clock obtained by dividing the clock selected by the DCS bit in the RSCFDnGCFG register. Set a division ratio by the BRP[9:0] bits in the RSCANn_{Cm}CFG register in classical CAN mode (CAN_mT_q clock), and by the NBRP[9:0] bits in the RSCFDnCFDCmNCFG register and the DBRP[7:0] bits in the RSCFDnCFDCmDCFG register in CAN FD mode (CAN_mT_q(N) clock and CAN_mT_q(D) clock). Be sure to specify the same values for both NBRP[9:0] and DBRP[7:0].

To specify different values for the nominal bit rate and the data bit rate, change the values of the RSCFDnCFDCmNCFG.NTSEG1 and NTSEG2 bits and RSCFDnCFDCmDCFG.DTSEG1 and DTSEG2 bits, respectively.

When the TDCE bit is set to 1 (Transmitter delay compensation is enabled) in the RSCFDnCFDCmFDCFG register, set the equal value of 1 or less to the bits NBRP[9:0] and DBRP[7:0].

Figure 20.17 shows the bit timing chart. **Table 20.189** shows an example of bit timing setting.

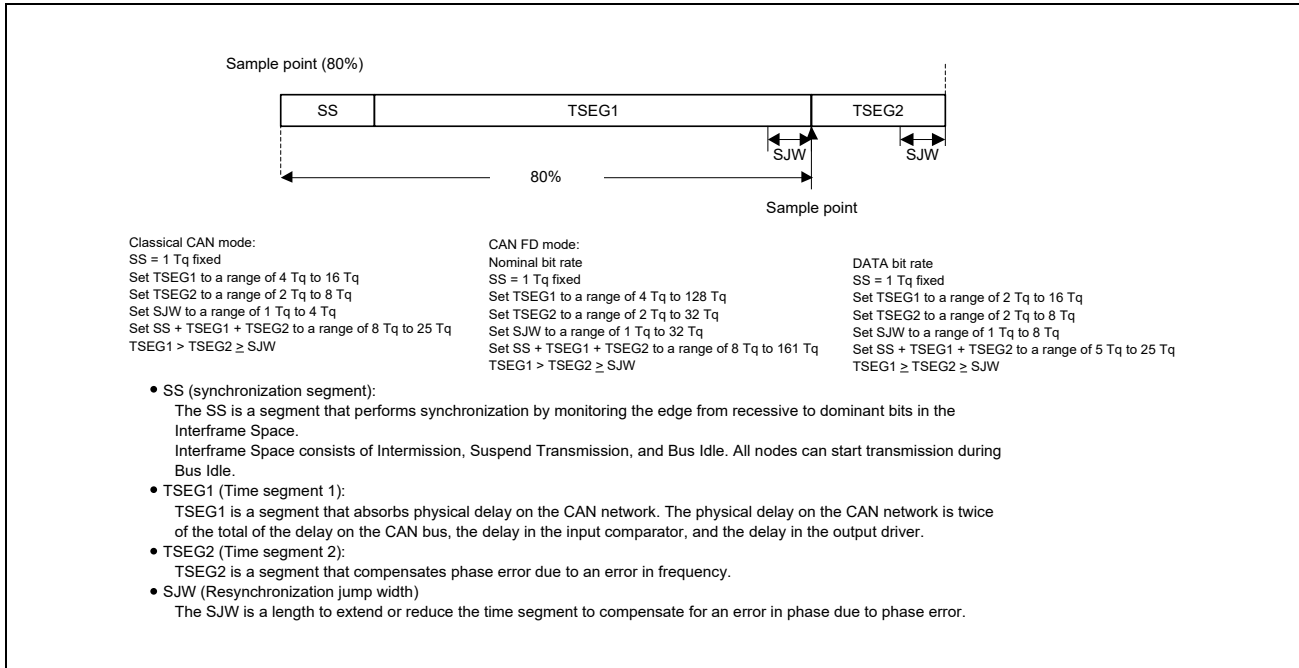


Figure 20.17 Bit Timing Chart

Table 20.189 Example of Bit Timing Settings

1 Bit	Set Value (Tq)				Sampling Point (%) Note: See Figure 20.17 .
	SS	TSEG1	TSEG2	SJW	
5 Tq *1	1	2	2	1	60.00
8 Tq	1	4	3	1	62.50
	1	5	2	1	75.00
10 Tq	1	6	3	1	70.00
	1	7	2	1	80.00
16 Tq	1	10	5	1	68.75
	1	11	4	1	75.00
20 Tq	1	12	7	1	65.00
	1	13	6	1	70.00
50 Tq *1	1	39	10	4	80.00

Note 1. Only in CAN FD mode

20.11.1.3 Communication Speed Setting

Set the CAN communication speed for each channel using the fCAN, baud rate prescaler division value, and Tq count per bit time. For CAN FD mode, set two types of transmission rate (arbitration phase and data phase) for each channel.

Figure 20.18 shows the CAN clock control block diagram, and Table 20.190 and Table 20.191 show an example of the communication speed setting.

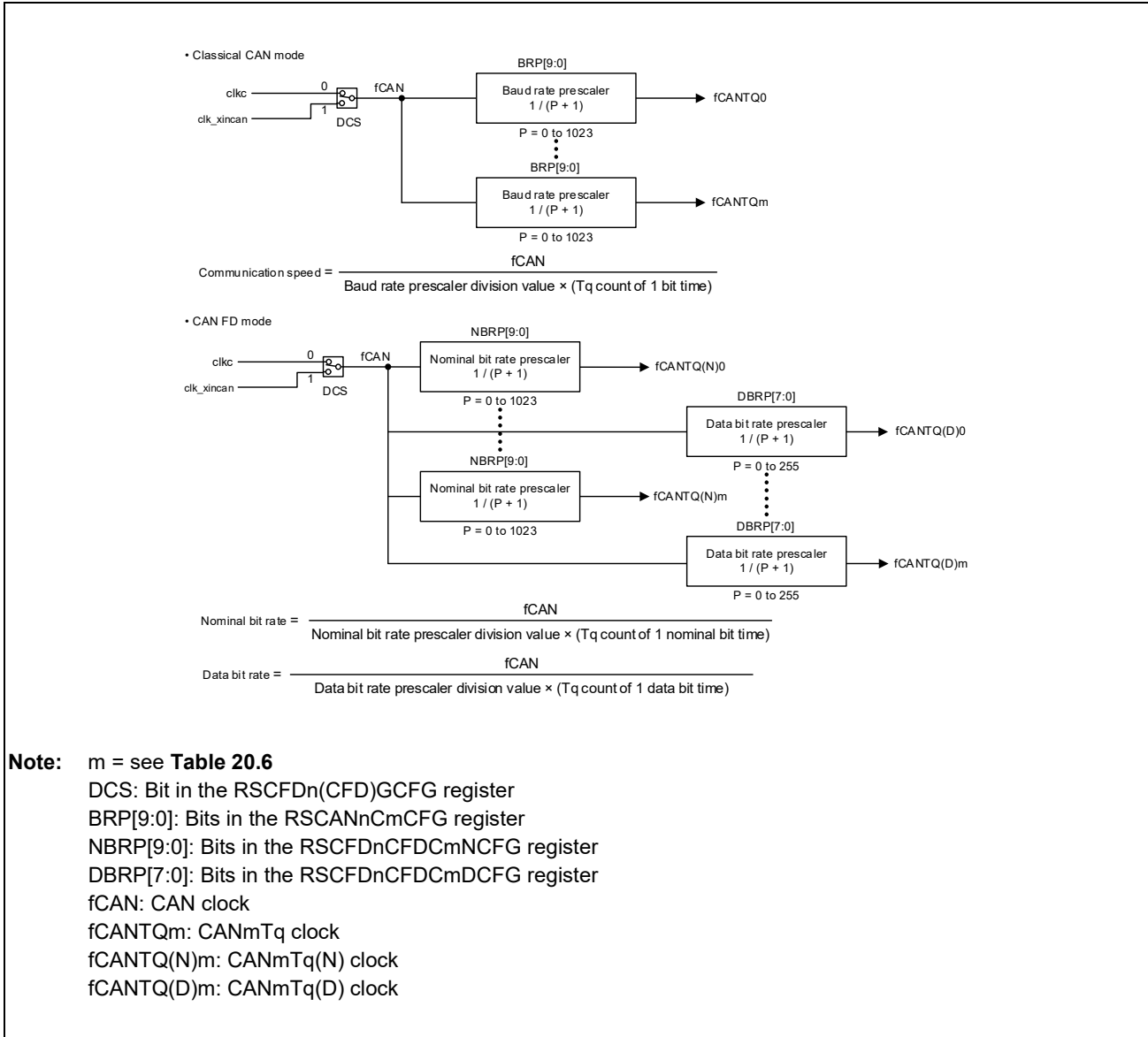


Figure 20.18 CAN Clock Control Block Diagram

Table 20.190 Example of Communication Speed Setting (in Classical CAN Mode)

Communication Speed	fCAN					
	40 MHz	32 MHz	24 MHz	20 MHz	16 MHz	8 MHz
1 Mbps	8 Tq (5) 20 Tq (2)	8 Tq (4) 16 Tq (2)	8 Tq (3) 12 Tq (2) 24 Tq (1)	10 Tq (2) 20 Tq (1)	8 Tq (2) 16 Tq (1)	8 Tq (1)
500 Kbps	8 Tq (10) 20 Tq (4)	8 Tq (8) 16 Tq (4)	8 Tq (6) 12 Tq (4) 24 Tq (2)	10 Tq (4) 20 Tq (2)	8 Tq (4) 16 Tq (2)	8 Tq (2) 16 Tq (1)
250 Kbps	8 Tq (20) 20 Tq (8)	8 Tq (16) 16 Tq (8)	8 Tq (12) 12 Tq (8) 24 Tq (4)	10 Tq (8) 20 Tq (4)	8 Tq (8) 16 Tq (4)	8 Tq (4) 16 Tq (2)
125 Kbps	8 Tq (40) 20 Tq (16)	8 Tq (32) 16 Tq (16)	8 Tq (24) 12 Tq (16) 24 Tq (8)	10 Tq (16) 20 Tq (8)	8 Tq (16) 16 Tq (8)	8 Tq (8) 16 Tq (4)

Table 20.191 Example of Communication Speed Setting (Nominal Bit Rate and Data Bit Rate in CAN FD Mode)

Communication Speed	fCAN	
	40 MHz	20 MHz
Nominal bit rate 1 Mbps Data bit rate 5 Mbps	Nominal 40 Tq (1) Data 8 Tq (1)	None
Nominal bit rate 500 kbps Data bit rate 2 Mbps	Nominal 80 Tq (1) Data 20 Tq (1)	Nominal bit rate 40 Tq (1) Data bit rate 10 Tq (1)

Note: Values in () are baud rate prescaler division values.

20.11.1.4 Receive Rule Setting

Receive rules can be set using receive rule-related registers.

Up to 16 receive rules can be registered per page. Specify pages 0 to 19 (for 5-channel unit) by the AFLPN[4:0] bits in the RSCFDn(CFD)GAFLECTR register. Set receive rule table write enable/disable by the AFLDAE bit.

Figure 20.19 shows the receive rule setting procedure.

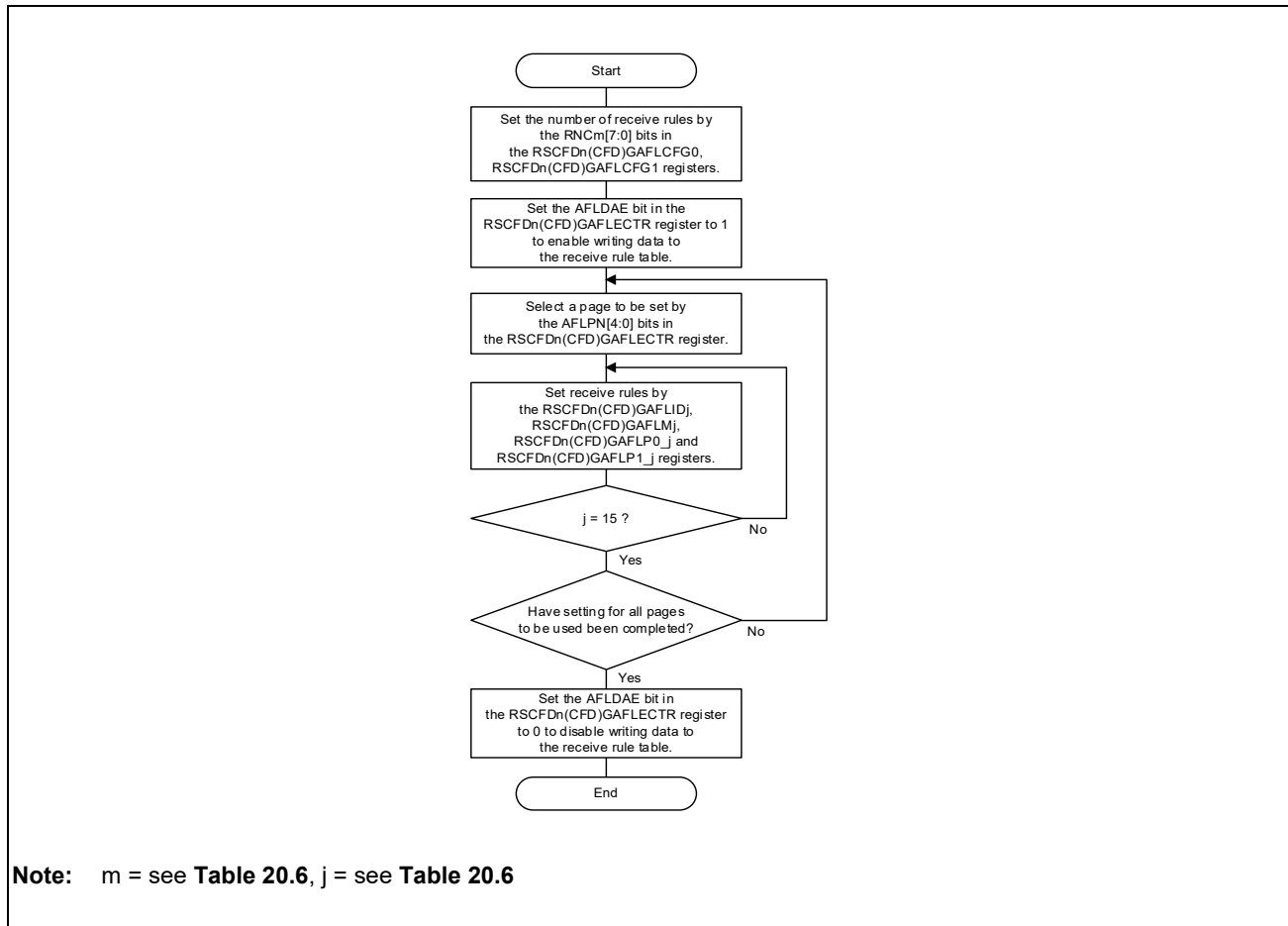


Figure 20.19 Receive Rule Setting Procedure

20.11.1.5 Buffer Setting

Set the number of buffers to be used (number of messages to be stored) and interrupt sources. Also set the payload storage size for CAN FD mode. For transmit/receive FIFO buffers that are set to transmit mode, set transmit buffers to be linked.

In classical CAN mode, up to 5120 bytes of the RAM can be used in receive buffers and FIFO buffers. Up to 320 buffers are available, and 16 bytes are used per buffer.

Configure the buffers so that the following conditions are met.

Number of receive buffers + total number of depth of receive FIFO buffers \times
+ total number of depth of transmit/receive FIFO buffers $k \leq 320$ buffers

In CAN FD mode, up to 8960 bytes of the RAM can be used in receive buffers and FIFO buffers.

Configure the buffers so that the following conditions are met.

Number of receive buffers $\times (12 + \text{payload storage size}) + \text{total of (number of depth} \times (12 + \text{payload storage size}))$ of receive FIFO buffers $\times + \text{total of (number of depth} \times (12 + \text{payload storage size}))$ of transmit/receive FIFO buffers $k \leq 8960$ bytes.

Figure 20.20 shows the buffer configuration. **Figure 20.21** shows the buffer setting procedure.

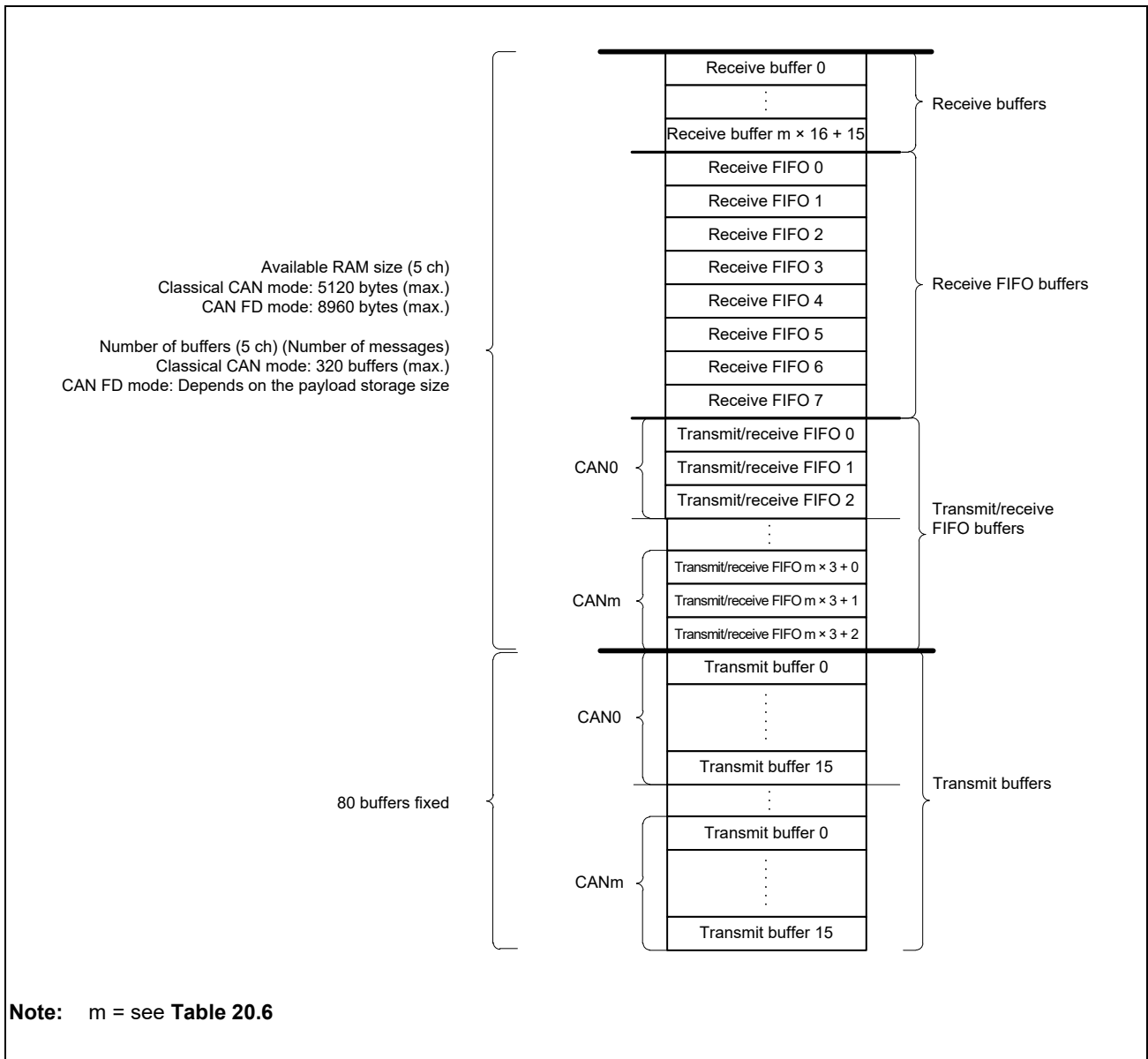


Figure 20.20 Buffer Configuration

CAUTION

Receive buffers, receive FIFO buffers, transmit/receive FIFO buffers, and transmit buffers are located in succession.

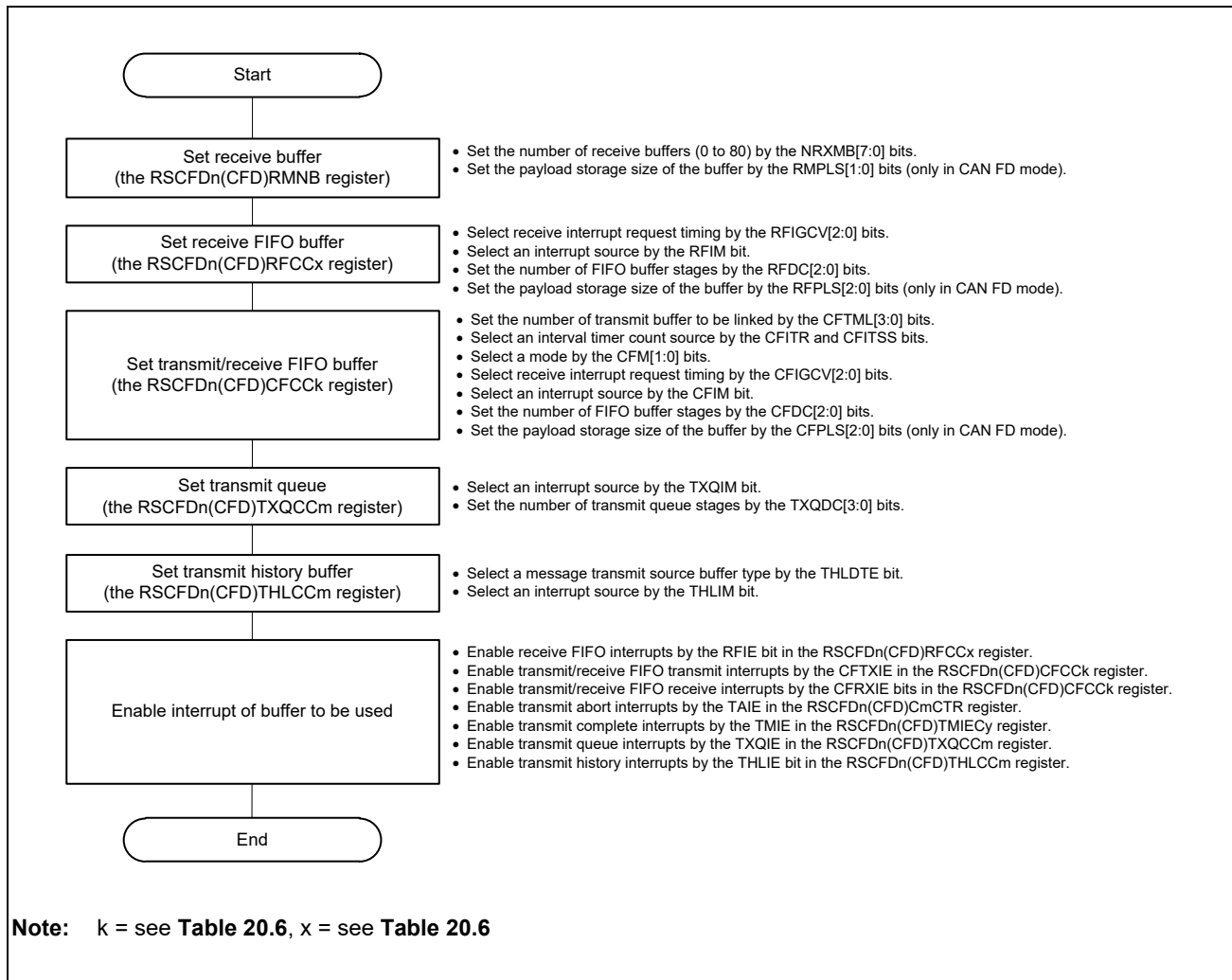


Figure 20.21 Buffer Setting Procedure

20.11.1.6 Transmitter Delay Compensation (Only in CAN FD Mode)

A high baud rate is used in the data phase in CAN FD mode. Transmitter delay compensation is provided as a function to accept propagation delay in this case.

To use this function, set the TDCE bit in the RSCFDnCFDCmFDCFG register to 1. Also set the secondary sample point (SSP) timing used in the data phase by the TDCOC bit and TDCO[6:0] bits in the RSCFDnCFDCmFDCFG register.

When the TDCOC bit is 0, the SSP timing equals the total value of the delay measured by the RS-CANFD module and the TDCO[6:0] value. (This value is rounded off to the nearest integer of T_q .)

Usually, the TDCO[6:0] value must be equal to SS + TSEG1, the sample point timing.

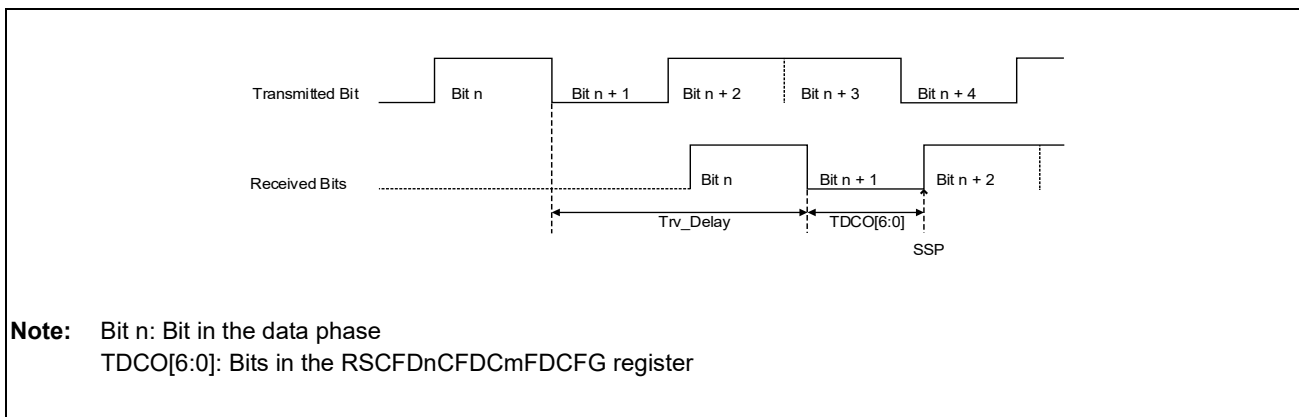


Figure 20.22 SSP timing

When the TDCOC bit is 1, the SSP timing is determined only by the TDCO[6:0] value. (When the DBRP[7:0] value in the RSCFDnCFDCmDCFG register is larger than 0, the TDCO[6:0] value is also rounded off to the nearest integer of T_q .)

The RS-CANFD module compensates a delay up to $(3 \text{ CANm bit time} - 2 f_{\text{CAN}})$. (CANm bit time is the value of data bit rate.)

When the TDCE bit is set to 1 (Transmitter delay compensation is enabled) in the RSCFDnCFDCmFDCFG register, set the equal value of 1 or less to the bits NBRP[9:0] and DBRP[7:0].

20.11.2 Reception Procedure

20.11.2.1 Receive Buffer Reading Procedure

When the processing to store received messages in a receive buffer starts, the RMNSq flag in the RSCFDn(CFD)RMNDy register (y = see **Table 20.6**, q = see **Table 20.6**) is set to 1 (receive buffer q contains a new message). Messages can be read from registers RSCFDn(CFD)RMIDq, RSCFDn(CFD)RMPTRq, RSCFDnCFDRMFDSTSq (only in CAN FD mode), and RSCFDn(CFD)RMDFb_q (b = 0 or 1 in classical CAN mode, b = 0 to 4 in CAN FD mode). If the next message has been received before the current message is read from the receive buffer, the message is overwritten.

Figure 20.23 shows the receive buffer reading procedure. This procedure ensures the consistency of messages read from registers RSCFDn(CFD)RMIDq, RSCFDn(CFD)RMPTRq, RSCFDnCFDRMFDSTSq, and RSCFDn(CFD)RMDFb_q.

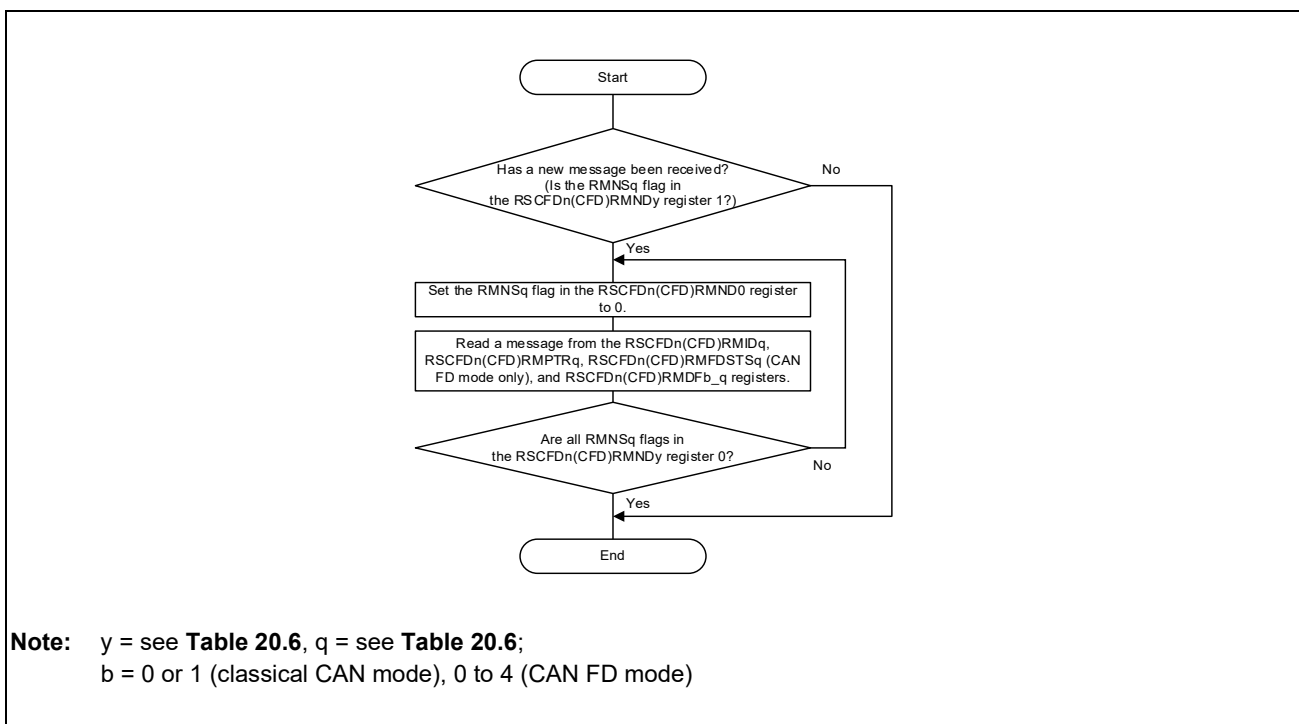


Figure 20.23 Receive Buffer Reading Procedure

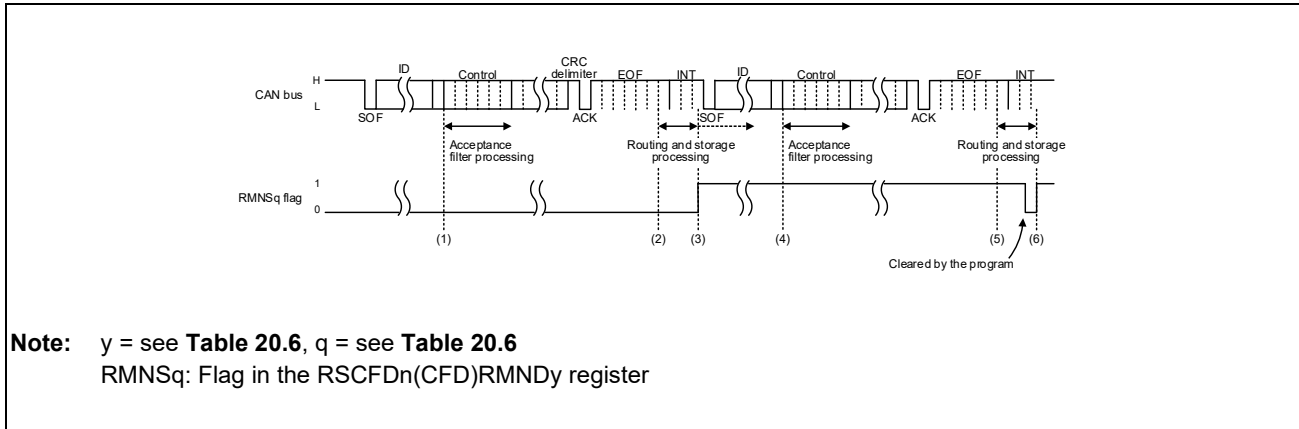


Figure 20.24 Receive Buffer Reception Timing Chart

- (1) When the ID field of a message has been received, the acceptance filter processing starts.
- (2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCFDn(CFD)GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.
- (3) When the message has passed through the DLC filter processing, the processing to store the message in the specified receive buffer starts.
When the message storage processing starts, the RMNSq flag in the corresponding RSCFDn(CFD)RMNDy register is set to 1 (receive buffer n contains a new message). If any other channels are performing filter processing or transmit priority determination processing, the routing processing and the storage processing may be delayed.
- (4) When the ID field of the next message has been received, the acceptance filter processing starts.
- (5) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCFDn(CFD)GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.
- (6) When the corresponding RMNSq flag is cleared to 0 (receive buffer n contains no new message), this flag is set to 1 again when the message storage processing starts. Even if the RMNSq flag remains 1, a new message is overwritten to the receive buffer. The RMNSq flag cannot be cleared to 0 during storage of messages.

20.11.2.2 FIFO Buffer Reading Procedure

When received messages have been stored in one or more receive FIFO buffers or transmit/receive FIFO buffers that are set to receive mode or gateway mode, the corresponding message counter (RFMC[7:0] bits in the RSCFDn(CFD)RFSTSx register (x = see **Table 20.6**) or CFMC[7:0] bits in the RSCFDn(CFD)CFSTS_k register (k = see **Table 20.6**) is incremented by 1. At this time, if the RFIE bit (receive FIFO interrupt is enabled) in the RSCFDn(CFD)RFCCx register or the CFRXIE bit (transmit/receive FIFO receive interrupt is enabled) in the RSCFDn(CFD)CFCC_k register is set to 1, an interrupt request is generated. Received messages can be read from the RSCFDn(CFD)RFID_x, RSCFDn(CFD)RFPTR_x, RSCFDn(CFD)RFFDSTS_x (only in CAN FD mode), and RSCFDn(CFD)RFDf_x (d = 0 or 1 in classical CAN mode, d = 0 to 15 in CAN FD mode) registers for receive FIFO buffers, or from the RSCFDn(CFD)CFID_k, RSCFDn(CFD)CFPTR_k, RSCFDn(CFD)CFFDCSTSk (only in CAN FD mode), and RSCFDn(CFD)CFDFd_k registers for transmit/receive FIFO buffers. Messages in FIFO buffers can be read sequentially on a first-in, first-out basis.

When the message counter value matches the FIFO buffer depth (a value set by the RFDC[2:0] bits in the RSCFDn(CFD)RFCCx register or the CFDC[2:0] bits in the RSCFDn(CFD)CFCC_k register), the RFFLL or CFFLL flag is set to 1 (the receive FIFO buffer is full).

When all messages have been read out of the FIFO buffer, the RFEMP flag in the RSCFDn(CFD)RFSTSx register or the CFEMP flag in the RSCFDn(CFD)CFSTS_k register is set to 1 (the receive FIFO buffer contains no unread message (buffer empty)).

If the RFE bit or the CFE bit is cleared to 0 (no receive FIFO buffer is used) with the interrupt request flag (RFIF flag in the RSCFDn(CFD)RFSTSx register or CFRXIF flag in the RSCFDn(CFD)CFSTS_k register) set to 1 (a receive FIFO interrupt request is present), the interrupt request flag is not automatically cleared to 0. The program must clear the interrupt request flag to 0.

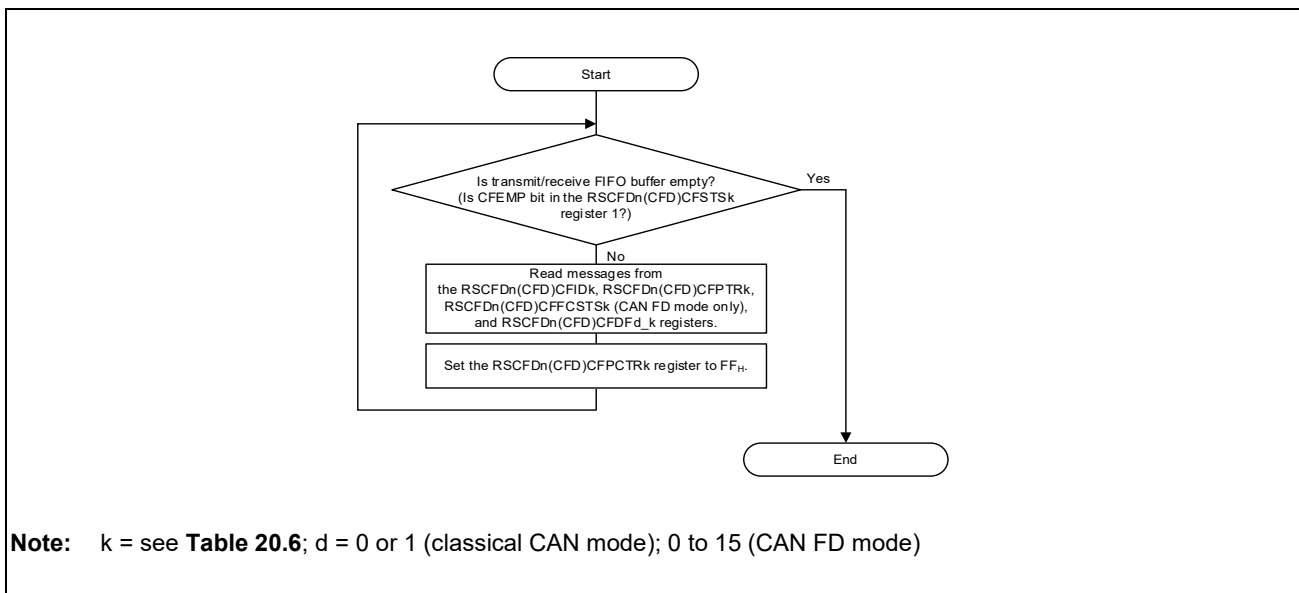


Figure 20.25 Transmit/Receive FIFO Buffer Reading Procedure

When reading a message in CAN FD mode, do not read the RSCFDn(CFD)RFDf_x or RSCFDn(CFD)CFDFd_k register corresponding to the area exceeding the payload storage size specified by the RFPLS[2:0] bits in the RSCFDn(CFD)RFCCx register or the CFPLS[2:0] bits in the RSCFDn(CFD)CFCC_k register.

Table 20.192 Payload Storage Area of Receive FIFO Buffer

Setting of RFPLS[2:0] Bits	Payload Storage Size	Corresponding Data Field Registers
000 _B	8 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF1_x
001 _B	12 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF2_x
010 _B	16 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF3_x
011 _B	20 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF4_x
100 _B	24 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF5_x
101 _B	32 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF7_x
110 _B	48 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF11_x
111 _B	64 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF15_x

Table 20.193 Payload Storage Area of Transmit/Receive FIFO Buffer

Setting of CFPLS[2:0] Bits	Payload Storage Size	Corresponding Data Field Registers
000 _B	8 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF1_k
001 _B	12 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF2_k
010 _B	16 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF3_k
011 _B	20 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF4_k
100 _B	24 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF5_k
101 _B	32 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF7_k
110 _B	48 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF11_k
111 _B	64 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF15_k

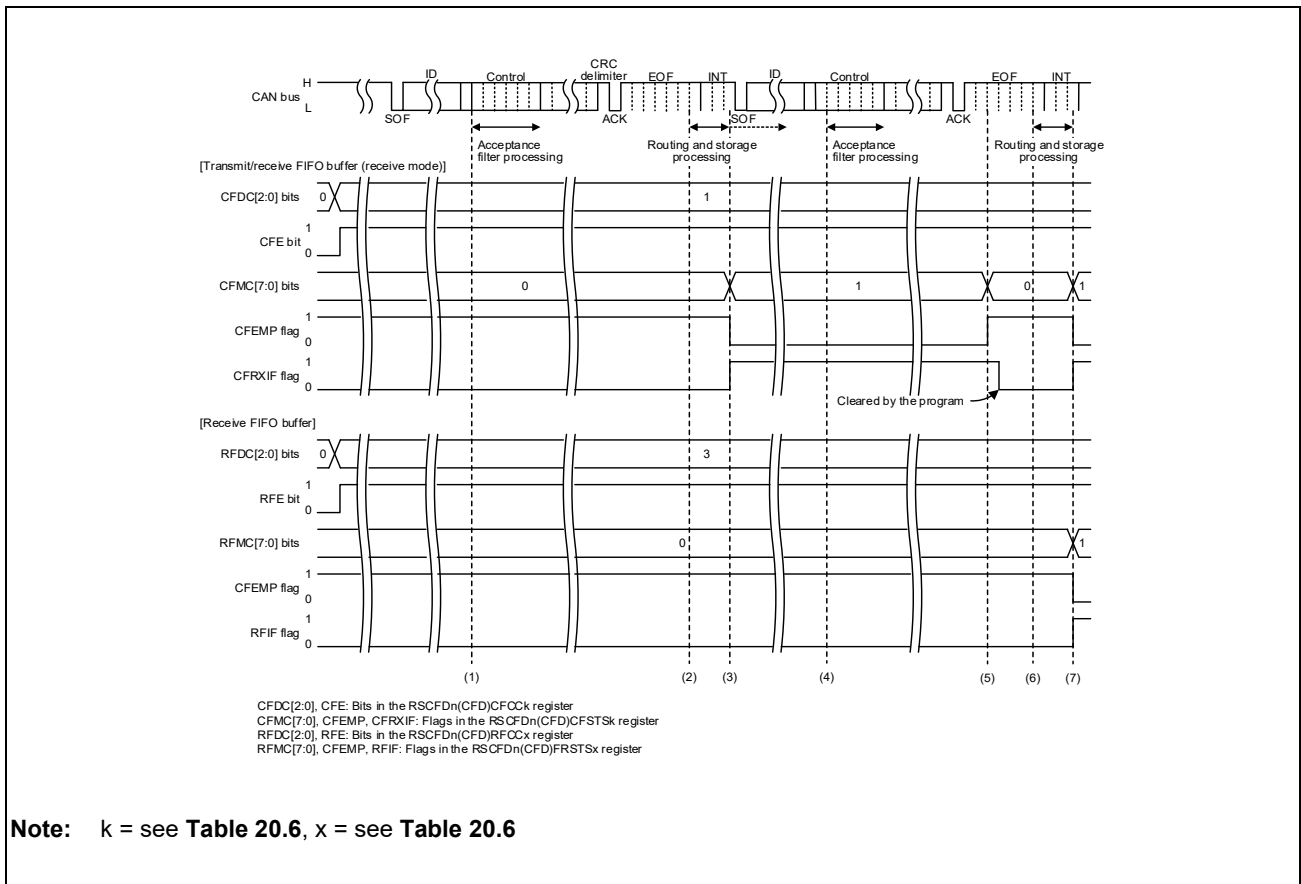


Figure 20.26 FIFO Buffer Reception Timing Chart

- (1) When the ID field of a message has been received, the acceptance filter processing starts.
- (2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCFDn(CFD)GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.
- (3) When the message has passed through the DLC filter processing and the CFE bit in the RSCFDn(CFD)CFCCk register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCFDn(CFD)CFCCk register is 001_B or more, the message is stored in the transmit/receive FIFO buffer that is set to receive mode. The CFMC[7:0] value in the RSCFDn(CFD)CFSTSk register is incremented by 1 and becomes 01_H. When the CFIM bit in the RSCFDn(CFD)CFCCk register is set to 1 (a FIFO receive interrupt request is generated each time a message has been received), the CFRXIF flag in the RSCFDn(CFD)CFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). The CFRXIF flag can be cleared to 0 by the program.
- (4) When the ID field of the next message has been received, the acceptance filter processing starts.
- (5) Read received messages from the RSCFDn(CFD)CFIDk, RSCFDn(CFD)CFPTRk, RSCFDn(CFD)CFDF_k, and RSCFDn(CFD)CFDFk registers and write FF_H to the RSCFDn(CFD)CFPCTRk register. This causes the CFMC[7:0] value in the RSCFDn(CFD)CFSTSk register is decremented by 1 and become 00_H, and the CFEMP flag in the RSCFDn(CFD)CFSTSk register becomes 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).
- (6) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCFDn(CFD)GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

- (7) The message having passed through the DLC filter process is stored in the transmit/receive FIFO buffer which is set to receive mode, when the CFE bit is set to 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] bits are set to 001_B or more. The CFMC[7:0] value is incremented by 1 to 01_H. When the CFIM bit is set to 1 (an interrupt occurs each time a message has been received), the CFRXIF flag is set to 1 (a transmit/receive FIFO receive interrupt request is present). The message is stored in the receive FIFO buffer when the RFE bit in the RSCFDn(CFD)RFCCx register is set to 1 (receive FIFO buffers are used.) and RFDC[2:0] bits in the RSCFDn(CFD)RFCCx register are set to 001_B or more. The RFMC[7:0] value in the RSCFDn(CFD)RFSTsx register is incremented by 1 to 01_H. When the RFIM bit in the RSCFDn(CFD)RFCCx register is set to 1 (an interrupt request occurs each time a message has been received), the RFIF flag in the RSCFDn(CFD)RFSTsx register is set to 1 (a receive FIFO interrupt request is present).

20.11.2.3 FIFO Buffer Reading Procedure by DMA Transfer

In CAN FD mode, the following FIFO buffers can be read by DMA transfer.

- All receive FIFO buffers x ($x =$ see **Table 20.6**)
- The first transmit/receive FIFO buffer k allocated to channel m ($k = 3 \times m$, $m =$ see **Table 20.6**)

The DMA enable bit (RFDMAEx or CFDMAEm bit in the RSCFDnCFDCDTCT register) can be set at any time. However, before setting the DMA enable bit to 1 (to enable DMA transfer requests), set the receive interrupt enable bit (RFIE bit in the RSCFDnCFDRFCCx register or CFRXIE bit in the RSCFDnCFDCFCCK register) of related FIFO buffers to 0 (to disable interrupts). When DMA transfer requests are enabled, do not write a value to the FIFO control register (RSCFDnCFDRFCCx register or RSCFDnCFDCFCCK register).

When an unread message is remaining in a DMA transfer-enabled FIFO buffer, a DMA transfer request trigger is generated. Specify the FIFO access register address(*1) for the transfer source address, and adjust the transfer size so that data can be read to the end of the payload storage area with a single trigger. The end of the payload storage area depends on the payload storage size specified by the RFPLS[2:0] bits in the RSCFDnCFDRFCCx register or the CFPLS[2:0] bits in the RSCFDnCFDCFCCK register.

After the end of the payload stored in the FIFO buffer has been read, the RFMC[7:0] value in the RSCFDnCFDRFSTSx register or the CFMC[7:0] value in the RSCFDnCFDCFSTS k register is automatically decremented by 1. After that, if an unread message is remaining in the FIFO buffer, a trigger is generated again.

When the RFDMAEx or CFDMAEm bit is set to 0 (to disable DMA transfer requests) during DMA transfer, wait until the DMA transfer status (RFDMASTSx or CFDMASTS m bit in the RSCFDnCFDCDTSTS register) is cleared to 0 (DMA transfer disabled), and then start the next processing (enabling DMA transfer again etc.). When disabling DMA transfer, examine how to process a message remaining in the FIFO buffer and a newly arriving message. When the FIFO buffer is enabled, it continues to receive messages.

Note 1.

- Receive FIFO buffer
RSCFDnCFDRFID x , RSCFDnCFDRFPTR x , RSCFDnCFDRFFDSTS x , RSCFDnCFDRFDFd $_x$
- Transmit / Receive FIFO buffer
RSCFDnCFDCFID k , RSCFDnCFDCFPTR k , RSCFDnCFDCFFDSTS k , RSCFDnCFDCFDFd $_k$

20.11.3 Transmission Procedure

20.11.3.1 Procedure for Transmission from Transmit Buffers

Figure 20.27 shows the procedure for transmission from transmit buffers.

Figure 20.28 shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmission has been successfully completed. **Figure 20.29** shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmit abort has been completed.

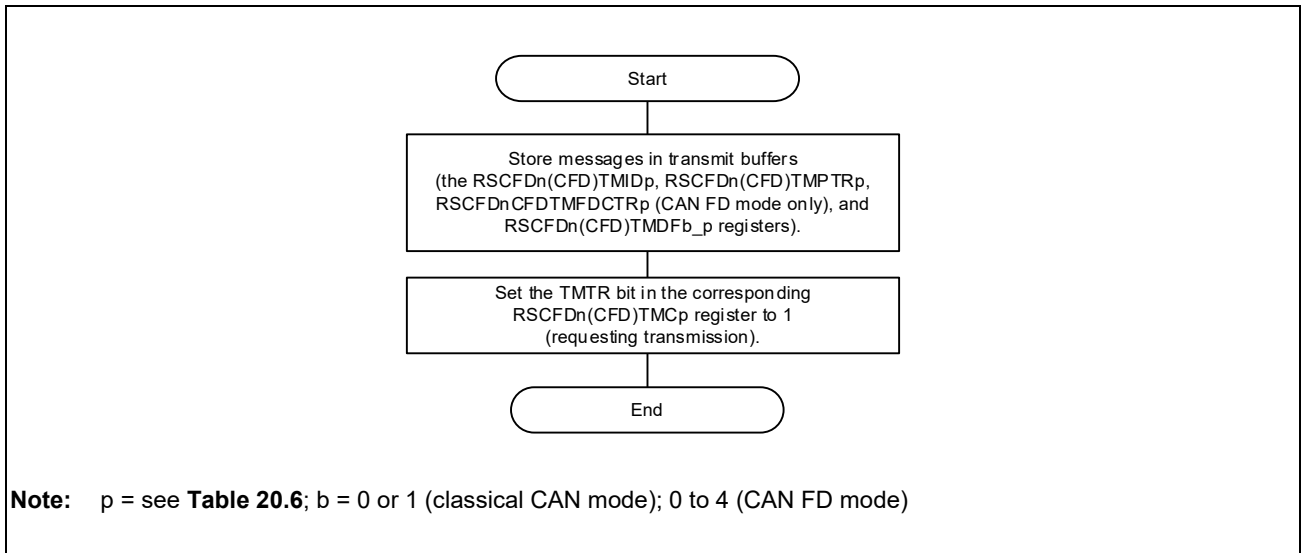


Figure 20.27 Procedure for Transmission from Transmit Buffers

In CAN FD mode and transmit buffer merge mode, messages with a payload size of more than 20 bytes can be transmitted from transmit buffers $(16 \times m) + 0$ and transmit buffers $(16 \times m) + 3$. At this time, transmit buffers $(16 \times m) + 1$ to $(16 \times m) + 2$ and transmit buffers $(16 \times m) + 4$ to $(16 \times m) + 5$ are allocated as a payload storage area. Registers RSCFDnCFDTMIDp, RSCFDnCFDTMPTRp, and RSCFDnCFDTMFDCTRp corresponding to these buffers can be used as data field registers that can store 4-byte data bytes (payload) like the RSCFDnCFDTMDFb_p register. **Table 20.194** shows message storage registers when transmitting a message with a payload size of more than 20 bytes from transmit buffer 0.

Table 20.194 Message Storage Registers in Transmit Buffer Merge Mode (Example of Transmit Buffer 0)

Transmit Buffer	Offset from Base Address	Symbol	Register Function in Transmit Buffer Merge Mode
Transmit buffer 0	4000 _H	RSCFDnCFDTMID0	Transmit buffer 0 ID data, transmit history data store enable bit, RTR bit, and IDE bit
	4004 _H	RSCFDnCFDTMPTR0	Transmit buffer 0 label data and DLC data
	4008 _H	RSCFDnCFDTMFDCTR0	Transmit buffer 0 ESI bit, BRS bit, and FDF bit
	400C _H to 401C _H	RSCFDnCFDTMDF0_0 to RSCFDnCFDTMDF4_0	Transmit buffer 0 data bytes 0, 1, 2, and 3 to transmit buffer 0 data bytes 16, 17, 18, and 19
Transmit buffer 1	4020 _H	RSCFDnCFDTMID1	Transmit buffer 0 data bytes 20, 21, 22, and 23
	4024 _H	RSCFDnCFDTMPTR1	Transmit buffer 0 data bytes 24, 25, 26, and 27
	4028 _H	RSCFDnCFDTMFDCTR1	Transmit buffer 0 data bytes 28, 29, 30, and 31
	402C _H to 403C _H	RSCFDnCFDTMDF0_1 to RSCFDnCFDTMDF4_1	Transmit buffer 0 data bytes 32, 33, 34, and 35 to transmit buffer 0 data bytes 48, 49, 50, and 51
Transmit buffer 2	4040 _H	RSCFDnCFDTMID2	Transmit buffer 0 data bytes 52, 53, 54, and 55
	4044 _H	RSCFDnCFDTMPTR2	Transmit buffer 0 data bytes 56, 57, 58, and 59
	4048 _H	RSCFDnCFDTMFDCTR2	Transmit buffer 0 data bytes 60, 61, 62, and 63
	404C _H to 405C _H	RSCFDnCFDTMDF0_2 to RSCFDnCFDTMDF4_2	Not used

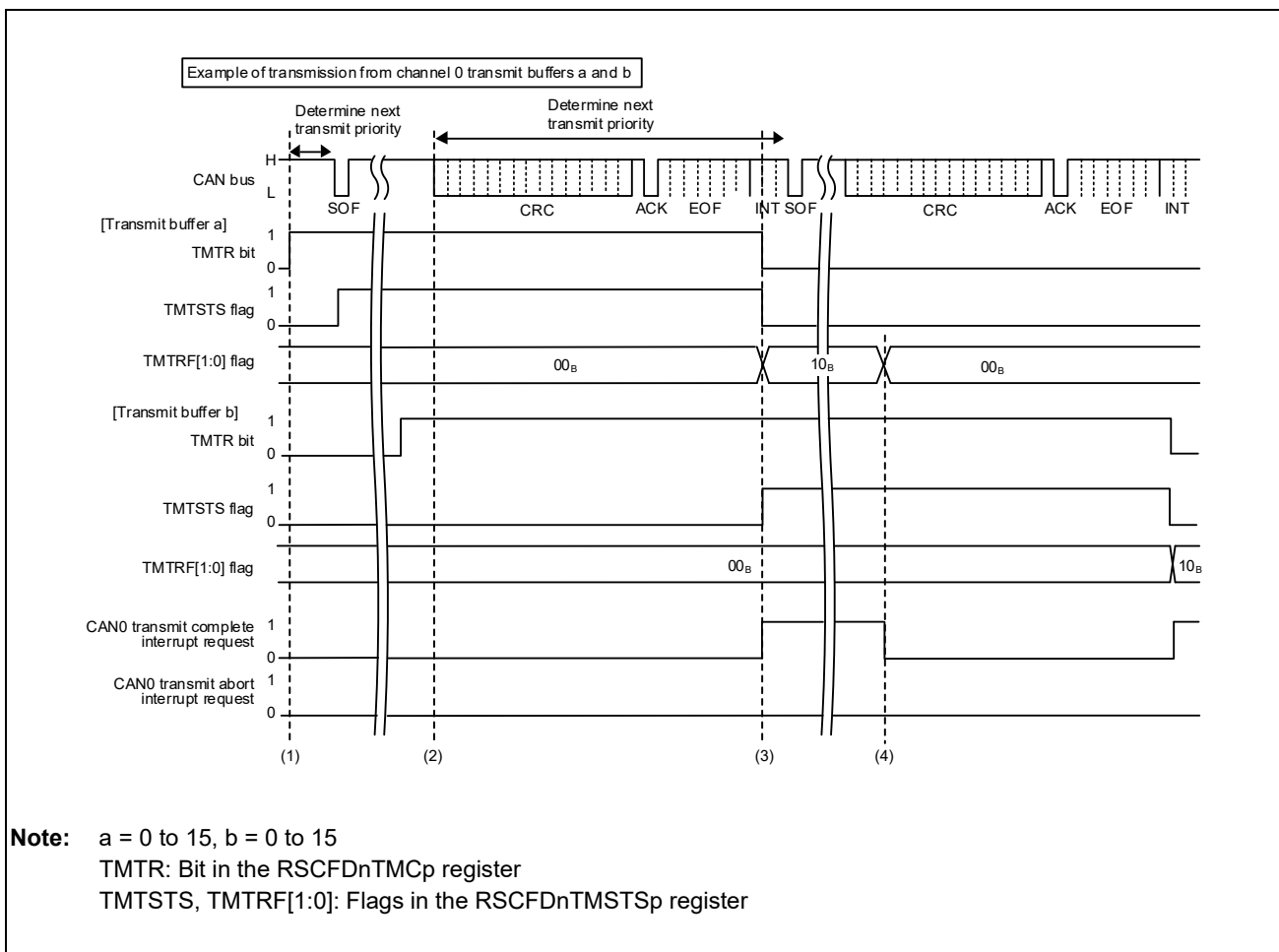


Figure 20.28 Transmit Buffer Transmission Timing Chart (Transmission Completed Successfully)

- (1) When the TMTR bit in the RSCFDn(CFD)TMCa register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer "a" is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding RSCFDn(CFD)TMSTSa register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.
- (2) When a transmit request from a buffer is present, the priority determination starts at the first bit of CRC field for the next transmission. Some delay may occur in the determination time if any other channels are performing the transmit priority determination processing. However, it does not affect the transmission interval because the determination will be completed before bit 3 of intermission.
- (3) When transmit completes successfully, the TMTRF[1:0] flag in the RSCFDn(CFD)TMSTSa register is set to 10_B (transmission has been completed (without transmit abort request)) and the TMTSTS flag and the TMTR bit in the RSCFDn(CFD)TMCa register are cleared to 0. When the TMIEa bit in the RSCFDn(CFD)TMIEC0 register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B (transmission is in progress or no transmit request is present).
- (4) Before starting the next transmission, set the TMTRF[1:0] flag to 00_B. Write the next message to the transmit buffer, and then set the TMTR bit to 1 (transmission is requested). The TMTR bit can be set to 1 only when the TMTRF[1:0] flag value is 00_B.

If an arbitration-lost occurs after transmission started, the TMTSTS flag is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC delimiter to search for the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration lost, the priority determination processing is reexecuted during transmission of an error frame.

When a 2-bit ECC error is detected during the priority determination processing, no data is transmitted (in classical CAN mode, only when the EEFE bit in the RSCANnGCFG register is 1).

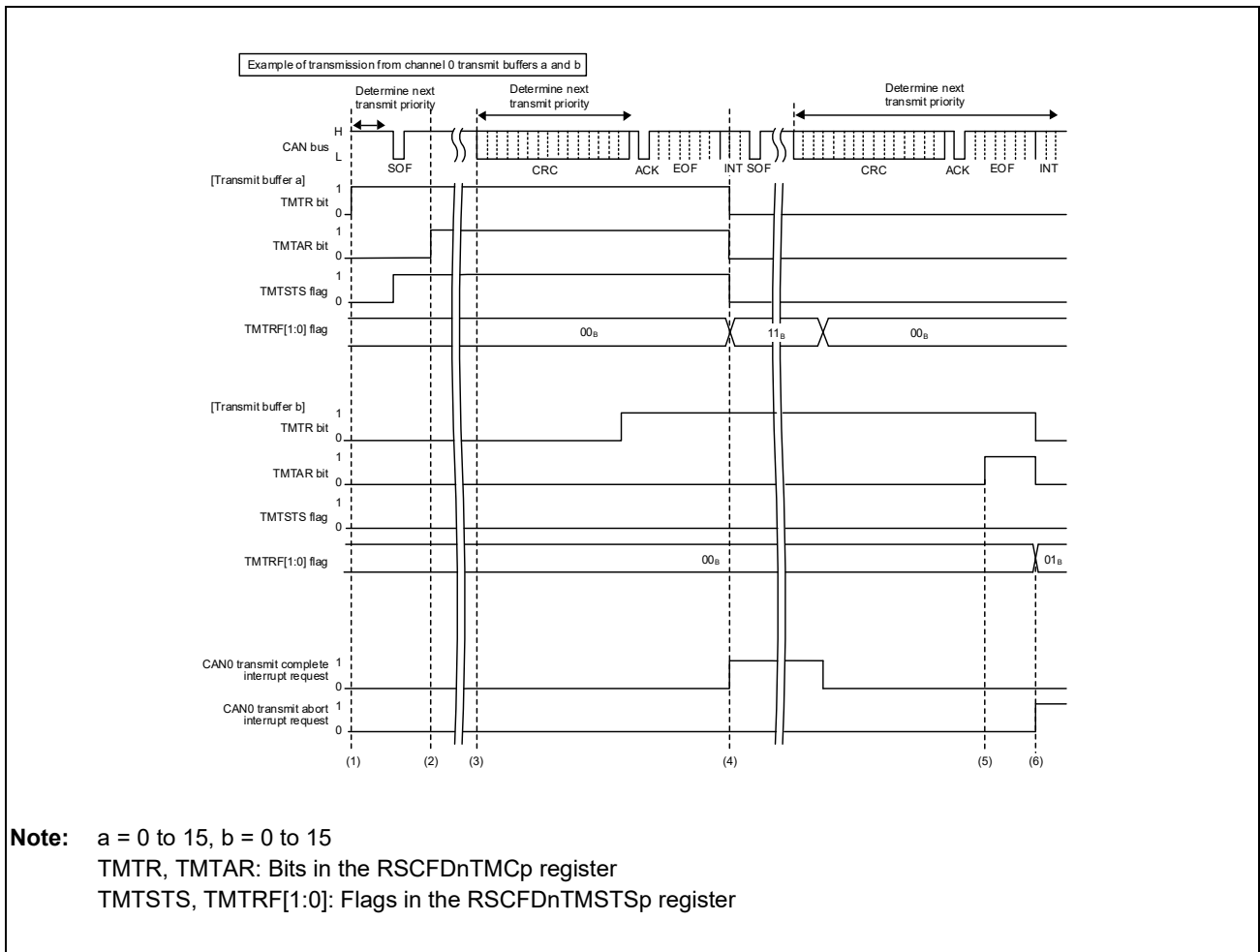


Figure 20.29 Transmit Buffer Transmission Timing Chart (Transmit Abort Completed)

- (1) When the TMTR bit in the RSCFDn(CFD)TMCa register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer "a" is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding RSCFDn(CFD)TMSTSa register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.
- (2) When the transmit buffer is determined for the next transmission or is in progress of transmission, message transmission is not aborted unless an error or arbitration lost occurs even if the TMTAR bit is set to 1 (transmit abort is requested).
- (3) The priority determination starts with the CRC delimiter for the next transmission. In this timing chart, buffer "b" is not selected as the next transmit buffer. Some delay may occur in the determination time if any other channels are performing the transmit priority determination processing. However, it does not affect the transmission interval because the determination will be completed before bit 3 of intermission.
- (4) When transmit completes successfully, the TMTRF[1:0] flag in the RSCFDn(CFD)TMSTSa register is set to 11_B (transmission has been completed (with transmit abort request)) and the TMTSTS flag and the TMTR bit in the RSCFDn(CFD)TMCa register are cleared to 0. When the TMIEa bit in the RSCFDn(CFD)TMIEC0 register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B (transmission is in progress or no transmit request is present).

- (5) When another CAN node is transmitting data on the CAN bus (TMTSTS flag = 0), the TMTR bit cannot be cleared to 0 if the TMTAR bit is set to 1 while the corresponding channel is determining transmit priority.
- (6) After the internal processing time has passed, the transmission is terminated and the TMTRF[1:0] flag is set to 01_B. When the transmit buffer is neither transmitting data nor selected as the next transmit buffer and when priority determination is not being made, an abort request is immediately accepted and the TMTRF[1:0] flag is set to 01_B. At this time, the TMTR and TMTAR bits are cleared to 0. When transmit abort is completed with the TAIE bit in the RSCFDn(CFD)CmCTR register set to 1 (transmit abort interrupt is enabled), an interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B.

If an arbitration lost occurs after the CAN channel started transmission, the TMTSTS bit is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC delimiter to search for the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration lost, the priority determination processing is reexecuted during transmission of an error frame.

When a 2-bit ECC error is detected during the priority determination processing, no data is transmitted (in classical CAN mode, only when the EEFE bit in the RSCANnGCFG register is 1).

20.11.3.2 Procedure for Transmission from Transmit/Receive FIFO Buffers

Figure 20.30 shows the procedure for transmission from transmit/receive FIFO buffers.

Figure 20.31 shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmission has been successfully completed. **Figure 20.32** shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmit abort has been completed.

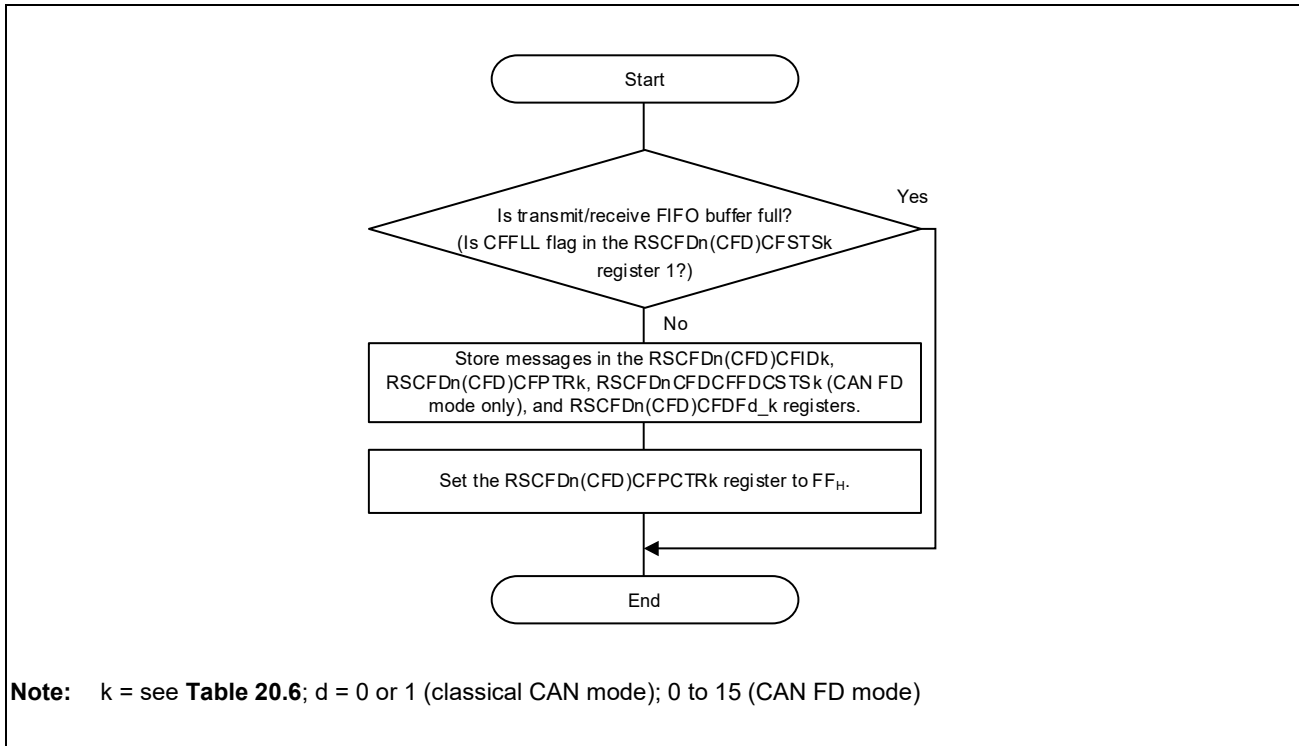


Figure 20.30 Procedure for Transmission from Transmit/Receive FIFO Buffers

When storing a message, do not write a value to the RSCFDnCFDCFDf_k register corresponding to the area exceeding the payload storage size specified by the CFPLS[2:0] bits in the RSCFDnCFDCFCCK register.

Table 20.195 Payload Storage Area of Transmit/Receive FIFO Buffer

Setting of CFPLS[2:0] Bits	Payload Storage Size	Corresponding Data Field Registers
000 _B	8 bytes	RSCFDnCFDCFDf _{0_k} to RSCFDnCFDCFDf _{1_k}
001 _B	12 bytes	RSCFDnCFDCFDf _{0_k} to RSCFDnCFDCFDf _{2_k}
010 _B	16 bytes	RSCFDnCFDCFDf _{0_k} to RSCFDnCFDCFDf _{3_k}
011 _B	20 bytes	RSCFDnCFDCFDf _{0_k} to RSCFDnCFDCFDf _{4_k}
100 _B	24 bytes	RSCFDnCFDCFDf _{0_k} to RSCFDnCFDCFDf _{5_k}
101 _B	32 bytes	RSCFDnCFDCFDf _{0_k} to RSCFDnCFDCFDf _{7_k}
110 _B	48 bytes	RSCFDnCFDCFDf _{0_k} to RSCFDnCFDCFDf _{11_k}
111 _B	64 bytes	RSCFDnCFDCFDf _{0_k} to RSCFDnCFDCFDf _{15_k}

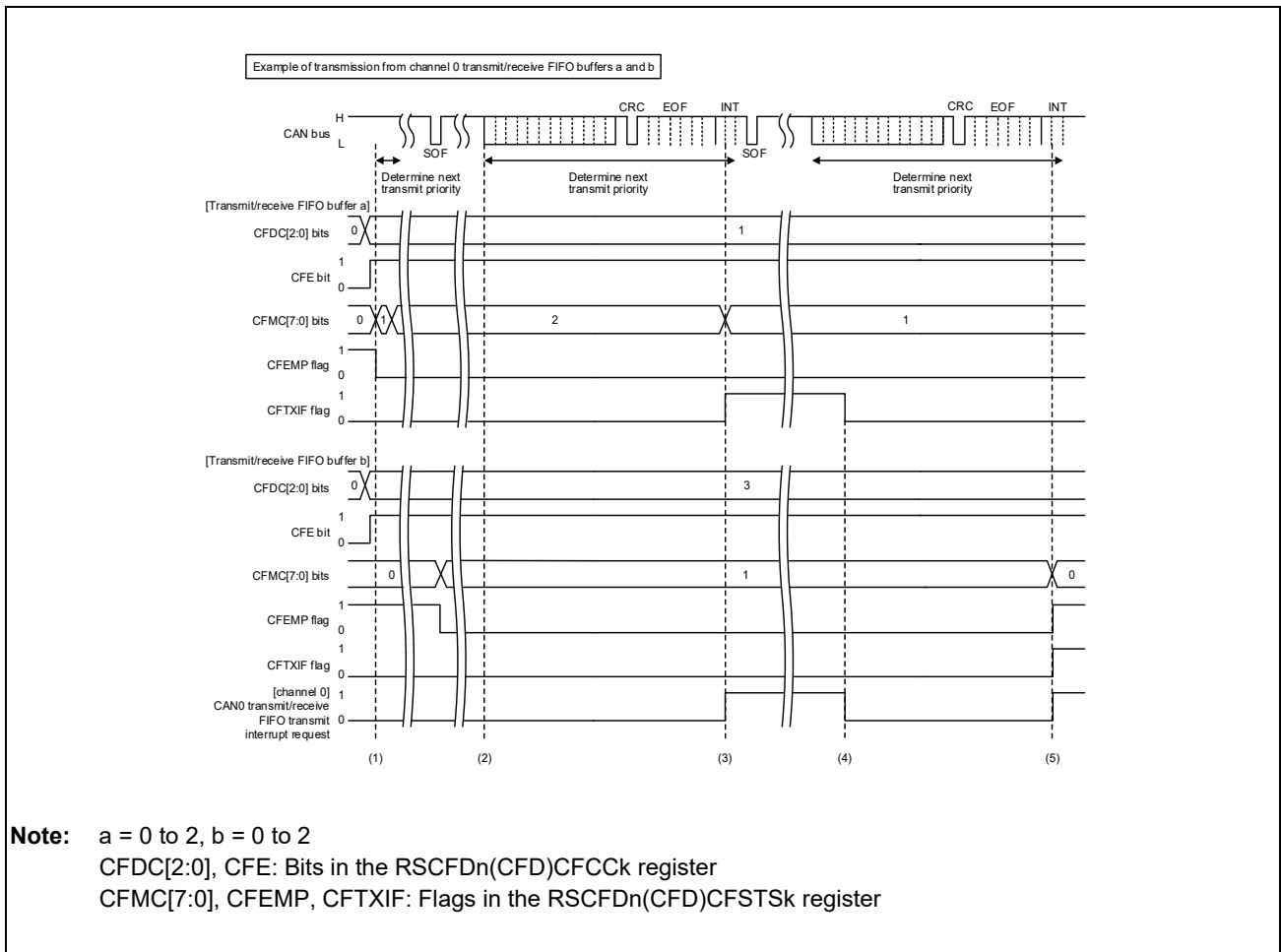


Figure 20.31 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmission Completed Successfully)

- (1) While the CAN bus is idle, the priority determination processing starts to determine the highest-priority transmit message when the CFE bit in the RSCFDn(CFD)CFCCa register is 1 (transmit/receive FIFO buffers are used), the CFDC[2:0] value in the RSCFDn(CFD)CFCCa register is 001_B (4 messages) or more, and the CFMC[7:0] value in the RSCFDn(CFD)CFSTSa register is 01_H or more. When the highest-priority transmit message has been determined, transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer "a" of channel 0.
- (2) When a transmit request from a buffer is present, the priority determination starts at the first bit of the CRC field for the next transmission. Some delay may occur in the determination time if any other channels are performing the transmit priority determination processing. However, it does not affect the transmission interval because the determination will be completed before bit 3 of intermission.
- (3) When transmit completes successfully, the CFMC[7:0] value in the RSCFDn(CFD)CFSTSa register is decremented by 1. Setting the CFIM bit in the RSCFDn(CFD)CFCCa register to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CCTXIF flag in the RSCFDn(CFD)CFSTSa register to 1 (a transmit/receive FIFO transmit interrupt request is present).
- (4) The CCTXIF flag can be cleared by the program.
- (5) Message transmission from transmit/receive FIFO buffer "b" of channel 0 completes and the CFMC[7:0] value in the RSCFDn(CFD)CFSTSa register is decremented by 1. The CFMC[7:0] bits are cleared to 00_H and therefore the

CFEMP flag in the RSCFDn(CFD)CFSTSb register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

Transmission is continued until the CFEMP flag is set to 1. It is possible to continuously store transmit messages in FIFO buffers until the CFFLL flag in the RSCFDn(CFD)CFSTSa register and RSCFDn(CFD)CFSTSb register are set to 1 (the transmit/receive FIFO buffer is full).

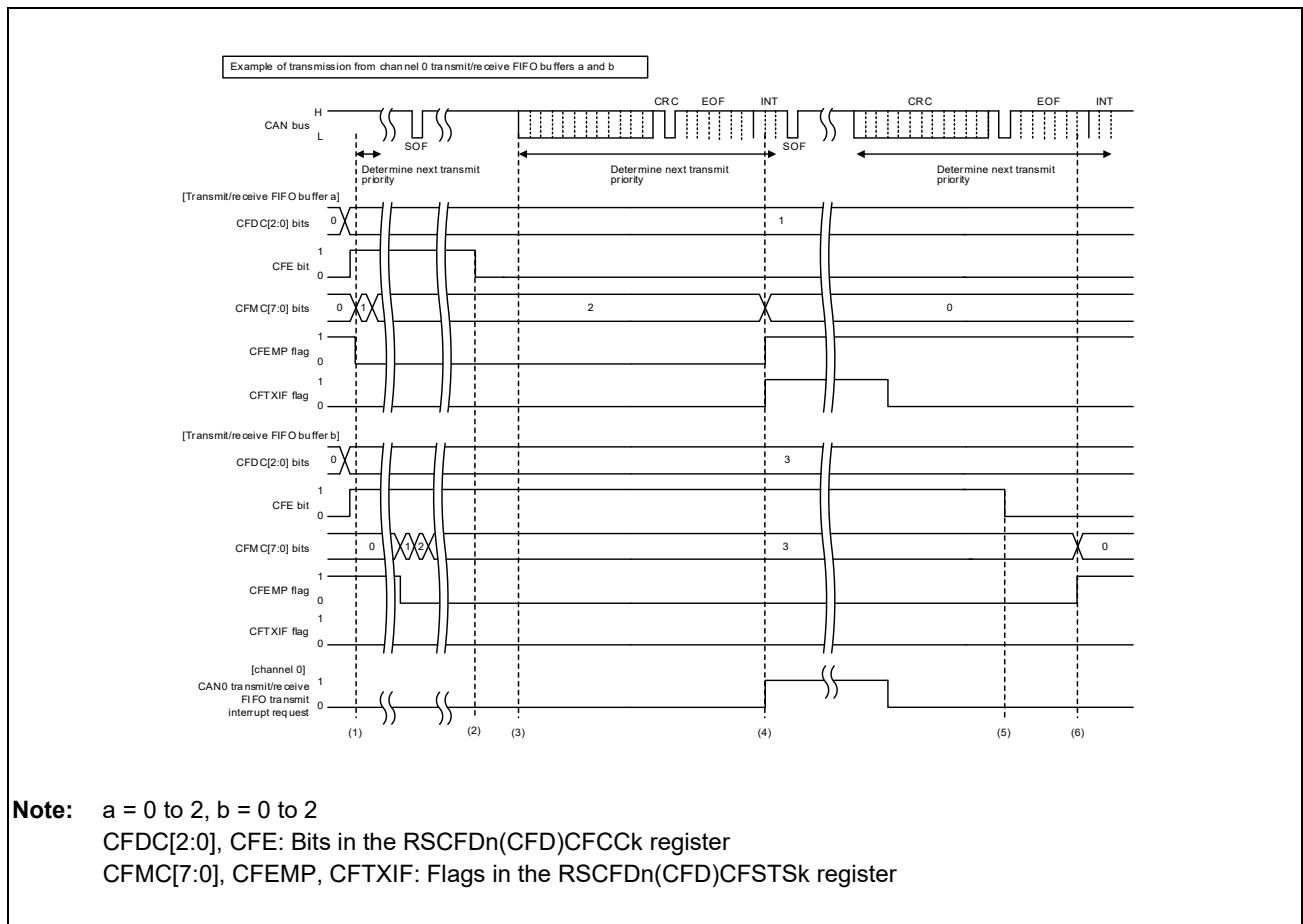


Figure 20.32 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmit Abort Completed)

- (1) While the CAN bus is idle, the priority determination processing starts to determine the highest-priority transmit message when the CFE bit in the RSCFDn(CFD)CFCCa register (a = 0 to 11) is 1 (transmit/receive FIFO buffers are used), the CFDC[2:0] value in the RSCFDn(CFD)CFCCa register is 001_B (4 messages) or more, and the CFMC[7:0] value in the RSCFDn(CFD)CFSTSa register is 01_H or more. When the highest-priority transmit message has been determined, transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer a of channel 0.
- (2) When transmission is in progress or the transmit/receive FIFO buffer is determined for the next transmission, message transmission is not aborted unless an error or arbitration lost occurs even if the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).
- (3) When a transmit request from a buffer is present, the priority determination starts at the first bit of the CRC field for the next transmission. In this figure, transmit/receive FIFO buffer b is not selected as a buffer for the next transmission. Some delay may occur in the determination time if any other channels are performing the transmit priority determination processing. However, it does not affect the transmission interval because the determination will be completed before bit 3 of intermission.

- (4) When transmit completes successfully, the CFMC[7:0] value is cleared to 00_H. Setting the CFIM bit to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CCTXIF flag in the RSCFD_n(CFD)CFSTSa register to 1 (a transmit/receive FIFO transmit interrupt request is present). The CCTXIF flag can be cleared by the program.
- (5) If another CAN node on the CAN bus is transmitting data (not from transmit/receive FIFO buffer "b"), transmit/receive FIFO buffers "b" cannot be disabled immediately even if the CFE bit in the RSCFD_n(CFD)CFCCb register is cleared to 0 (no transmit/receive FIFO buffer is used) during transmit priority determination. (The CFEMP flag in the RSCFD_n(CFD)CFSTSB register is not set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) immediately.)
- (6) After the internal processing time has passed, transmit/receive FIFO buffers "b" are disabled and the CFMC[7:0] bits in the RSCFD_n(CFD)CFSTSB register are cleared to 00_H and the CFEMP flag is set to 1. When the transmit/receive FIFO buffer "b" is neither transmitting data nor selected as the next transmit buffer and when priority determination is not in progress, the transmit/receive FIFO "b" buffer is immediately disabled. (The CFMC[7:0] bits are cleared to 00_H and the CFEMP flag is set to 1.)

20.11.3.3 Procedure for Transmission from the Transmit Queue

Figure 20.33 shows the procedure for transmission from the transmit queue.

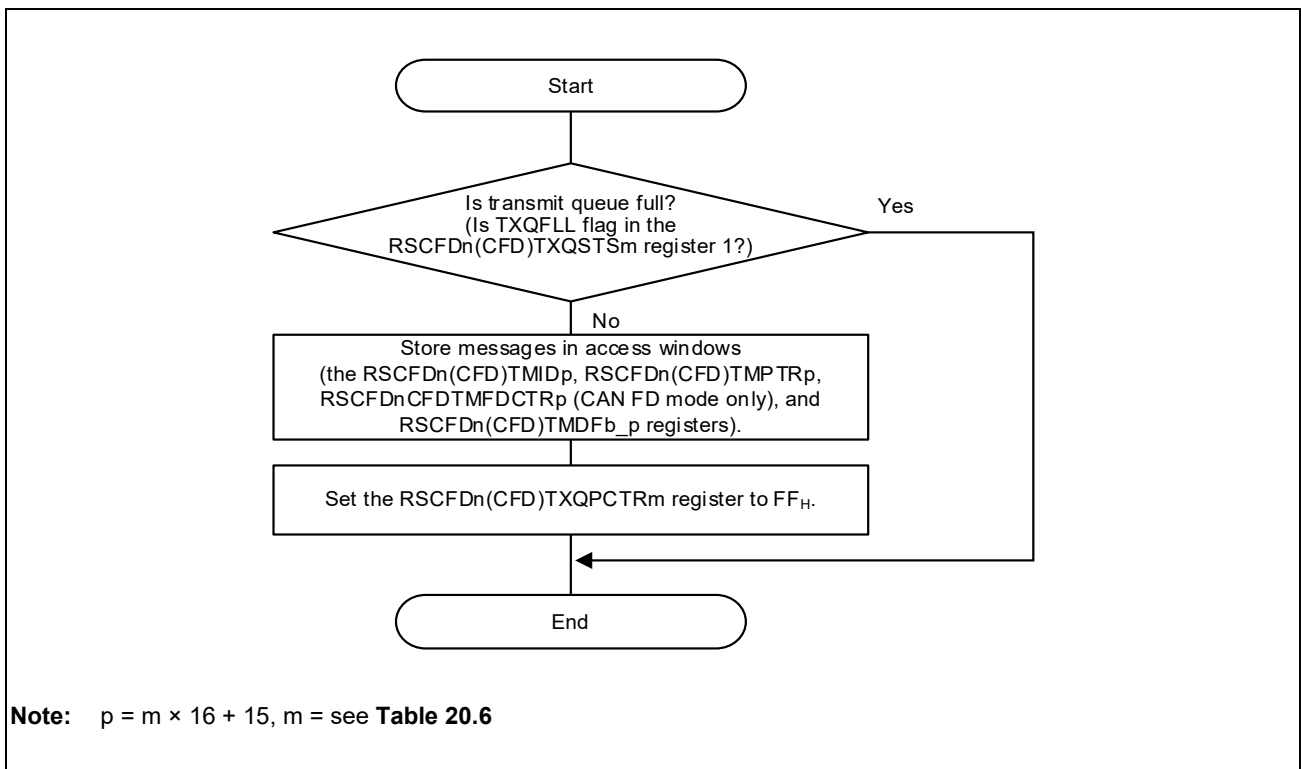


Figure 20.33 Procedure for Transmission from the Transmit Queue

20.11.3.4 Transmit History Buffer Reading Procedure

Transmit history data can be read from the RSCFDn(CFD)THLACCm register. The next data can be accessed by writing FF_H to the corresponding RSCFDn(CFD)THLPCTRm register (m = see **Table 20.6**) after reading a set of data.

Figure 20.34 shows the transmit history buffer reading procedure.

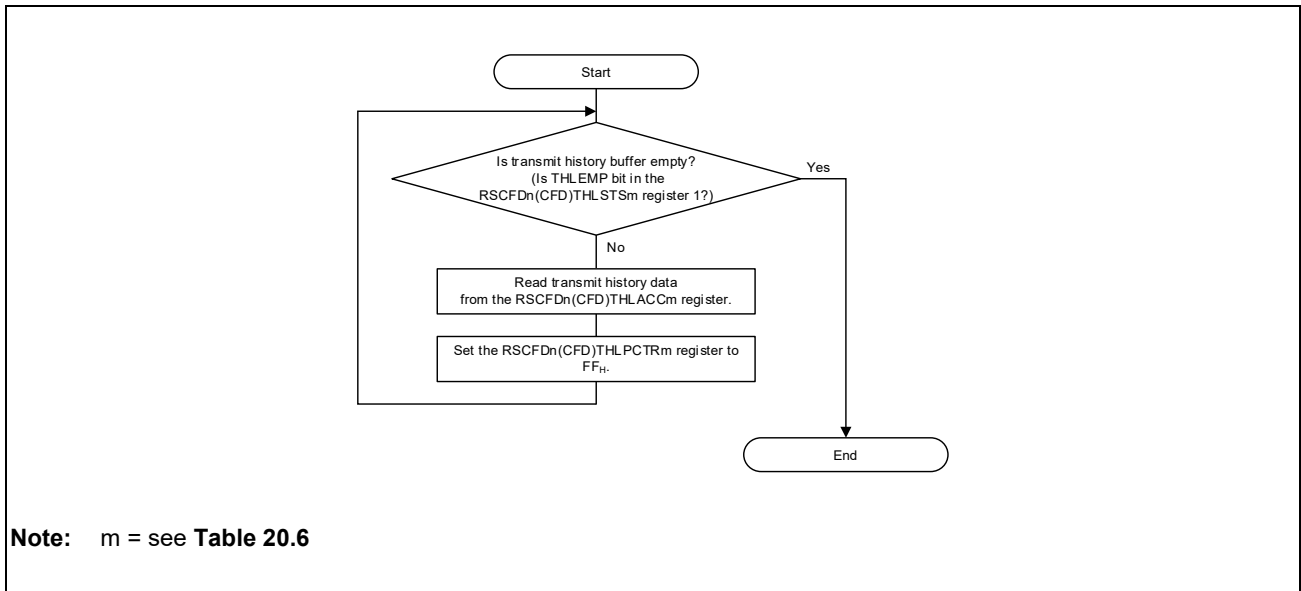


Figure 20.34 Transmit History Buffer Reading Procedure

20.11.4 Test Settings

20.11.4.1 Self-Test Mode Setting Procedure

Self-test mode allows communication test on a channel basis by receiving messages transmitted from the own node.

Figure 20.35 shows the self-test mode setting procedure.

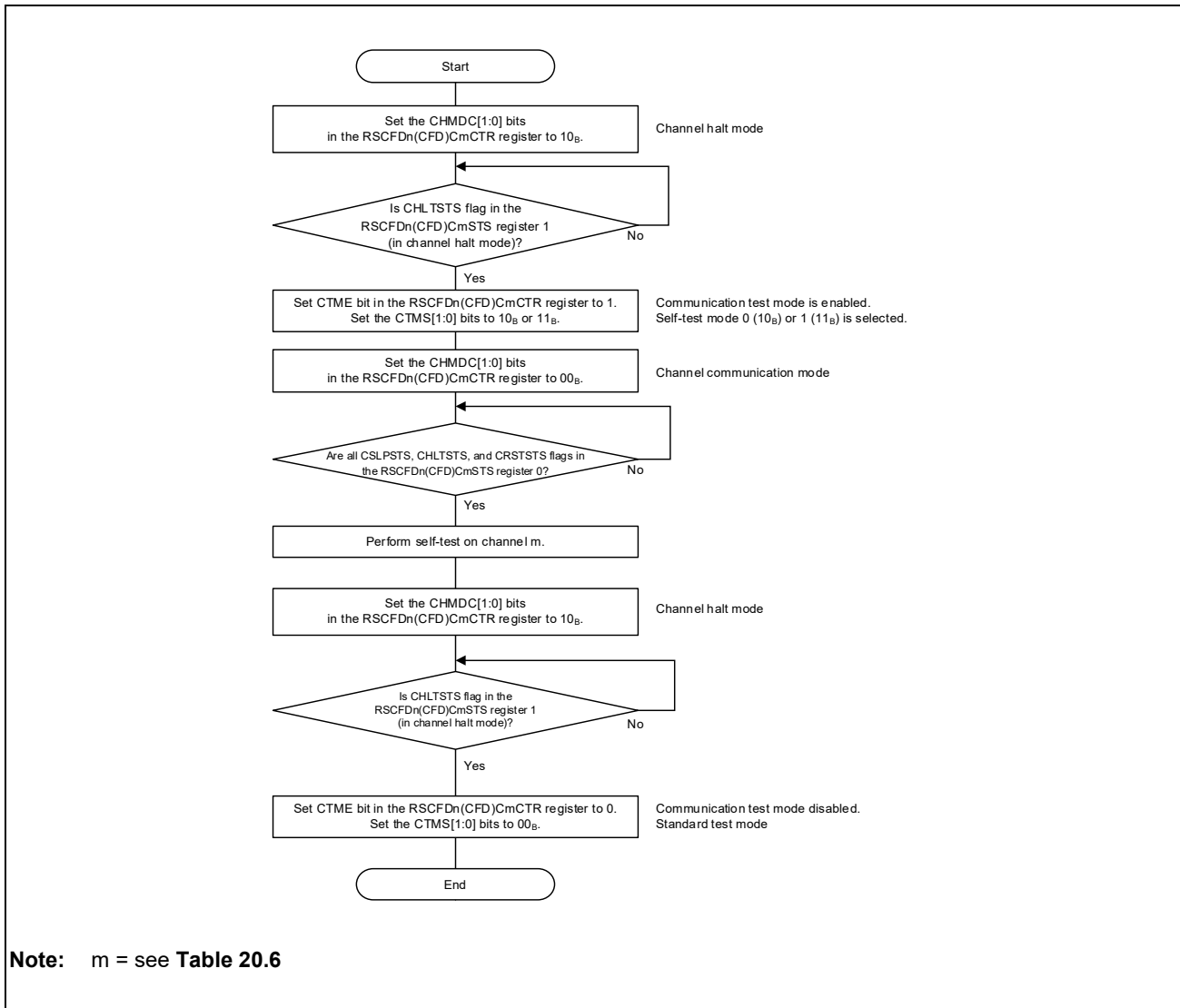


Figure 20.35 Self-Test Mode Setting Procedure

20.11.4.2 Procedure for Releasing the Protection

Since the global test function in Table 20.196 is protected, write the protection release data 1 and release data 2 in succession to the LOCK[15:0] bits in the RSCFDn(CFD)GLOCKK register, then set the target test bit to 1.

Table 20.196 Protection Release Data for Test Function

Test Function	Protection Release Data 1	Protection Release Data 2	Target Bit
RAM test	7575 _H	8A8A _H	RTME bit in the RSCFDn(CFD)GTSTCTR register

If an incorrect value is written to the LOCK[15:0] bits, restart from writing the protection release data 1. **Figure 20.36** shows the procedure for releasing the protection.

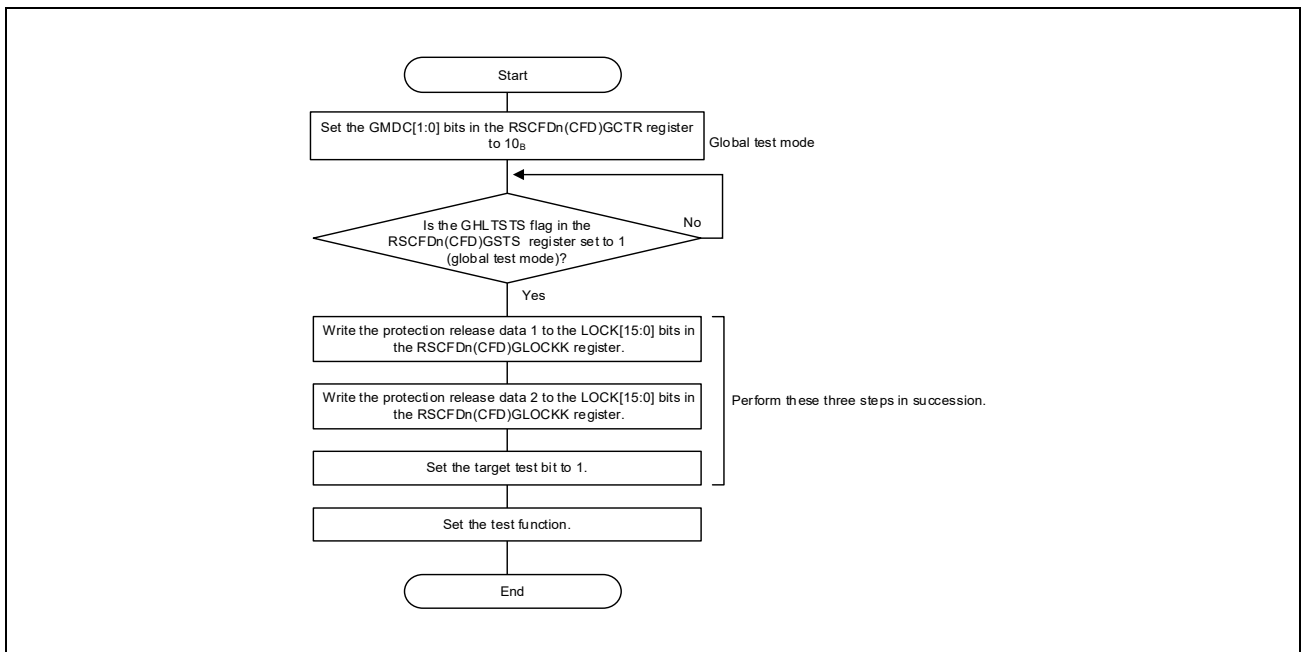


Figure 20.36 Protection Release Procedure

20.11.4.3 RAM Test Setting Procedure

RAM tests include CAN RAM read/write test. The read/write test verifies that data written to the RAM is read correctly. Before ending the RAM test, write 0000 0000_H to all pages of the CAN RAM.

Figure 20.37 shows the RAM test setting procedure.

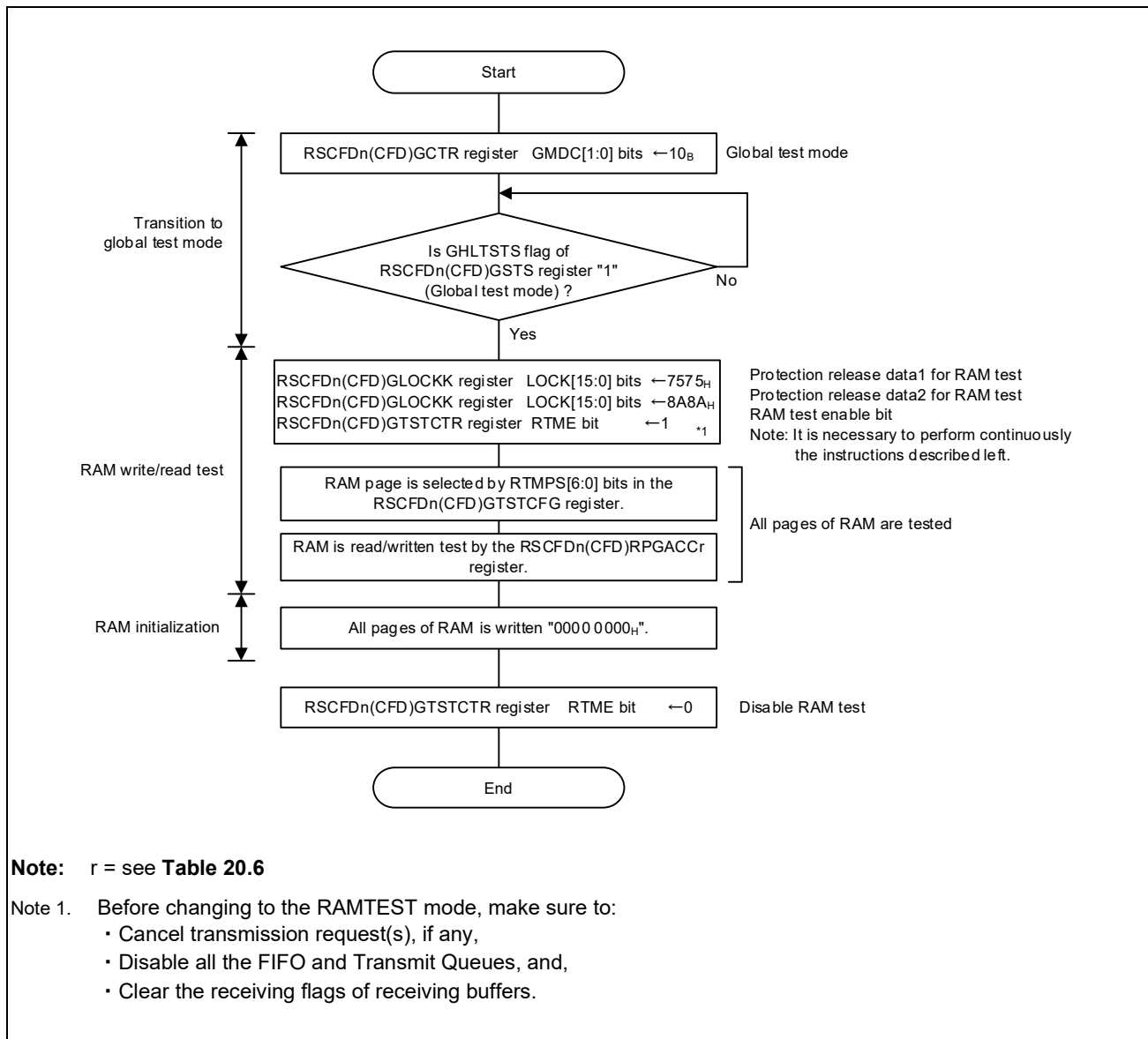


Figure 20.37 RAM Test Setting Procedure

20.11.4.4 Inter-Channel Communication Test Setting Procedure

Communication test can be performed by transmitting and receiving data between different channels.

Figure 20.38 shows the inter-channel communication test setting procedure.

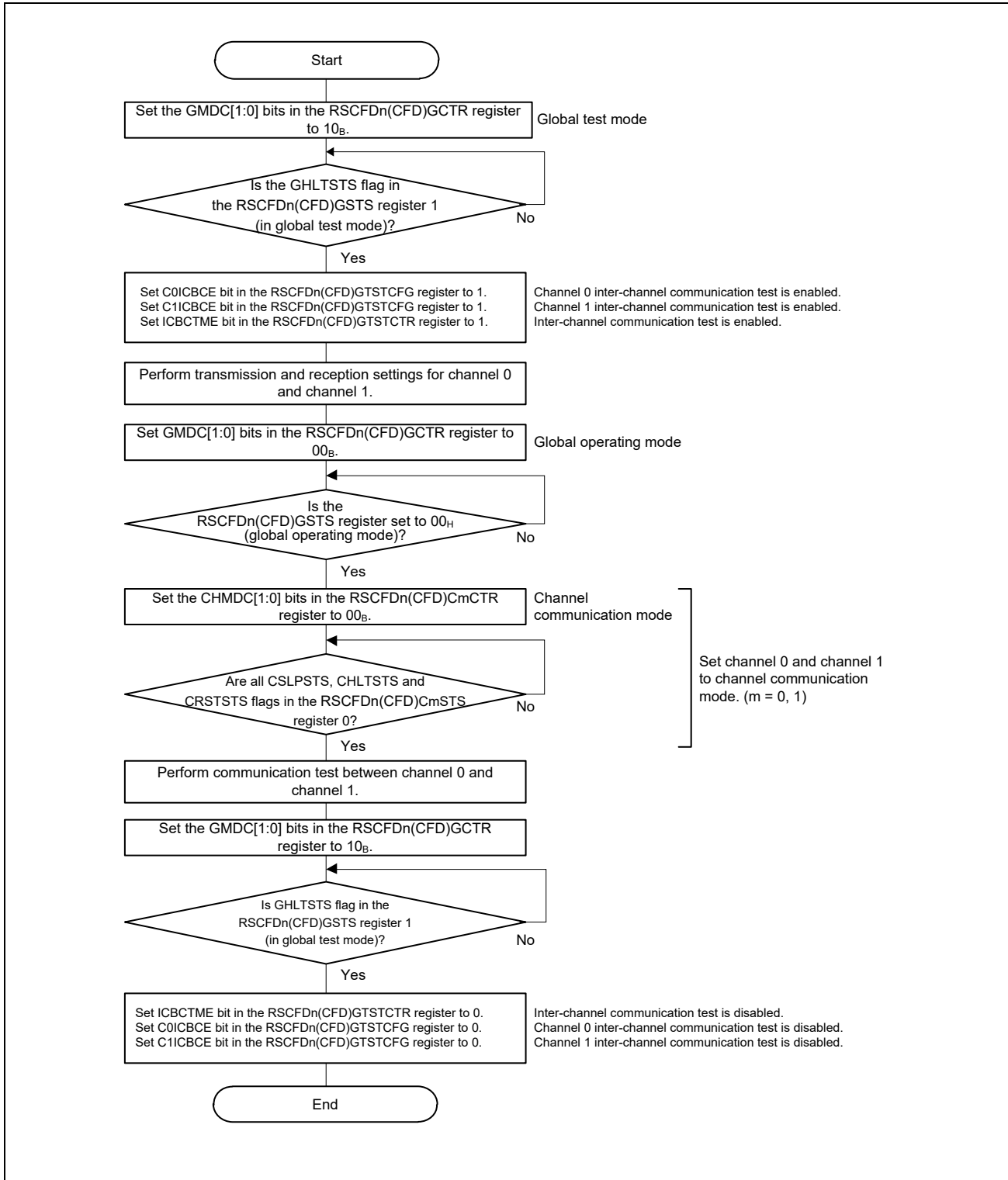


Figure 20.38 Inter-Channel Communication Test Setting Procedure (Example of Communication Test between Channel 0 and Channel 1)

20.12 Notes on the RS-CANFD Module

- When changing interface mode without resetting the RS-CANFD, write the value after reset to all registers and bits that are not allocated to the register map after change and then modify the RSCFDn(CFD)GRMCFG register.
- When changing a global mode, check the GSLPSTS, GHLTSTS, and GRSTSTS flags in the RSCFDn(CFD)GSTS register for transitions. When changing a channel mode, check the CSLPSTS, CHLTSTS, and CRSTSTS flags in the RSCFDn(CFD)CmSTS register (m = see **Table 20.6**) for transitions.
- When only classical CAN frames are used in CAN FD mode, set the RSCFDnCFDCmDCFG register to the value equal to the set RSCFDnCFDCmNCFG register value.
- The acceptance filter processing checks receive rules sequentially in ascending order from the lowest rule number. If the same ID, IDE bit, or RTR bit value is set in multiple receive rules, the lowest-numbered receive rule is used to pass the received message through the acceptance filter processing. If the message does not pass through the subsequent DLC filter processing, the data processing is terminated without returning to the acceptance filter processing and the message is not stored in the buffer.
- When linking transmit buffers to transmit/receive FIFO buffers or allocating transmit buffers to transmit queues, set the control register (RSCFDn(CFD)TMCp register) of the corresponding transmit buffer to 00_H. The status register (RSCFDn(CFD)TMSTSp register) of the corresponding transmit buffer should not be used. Flags in other status registers (RSCFDn(CFD)TMTRSTS0 to RSCFDn(CFD)TMTRSTS2, RSCFDn(CFD)TMTARSTS0 to RSCFDn(CFD)TMTARSTS2, RSCFDn(CFD)TMTCASTS0 to RSCFDn(CFD)TMTCASTS2, and RSCFDn(CFD)TMTASTS0 to RSCFDn(CFD)TMTASTS2 registers), which correspond to transmit buffers linked to transmit/receive FIFO buffers or allocated to transmit queues, remain unchanged. Set the enable bit in the corresponding interrupt enable register (RSCFDn(CFD)TMIEC0 to RSCFDn(CFD)TMIEC2 register) to 0 (transmit buffer interrupt is disabled).
- When using transmit buffer merge mode (in CAN FD mode), write 00_H to the control register (RSCFDn(CFD)TMCp) of the transmit buffer corresponding to the transmit buffer allocated as a payload storage area. Set the enable bit of corresponding interrupt enable registers (RSCFDn(CFD)TMIEC0 to RSCFDn(CFD)TMIEC2) to 0 (to disable interrupts).
- Transmit buffers that are linked to transmit/receive FIFO buffers must not be allocated to transmit queues. Do not allocate a transmit buffer allocated as a payload storage area in transmit buffer merge mode (in CAN FD mode) to the transmit queue either.
- Only a single transmit/receive FIFO buffer can be linked to a transmit buffer. Do not link two or more transmit/receive FIFO buffers to the same transmit buffer.
- When the CANm bit time clock is selected as a timestamp counter clock source, the timestamp counter stops when the corresponding channel has transitioned to channel reset mode or channel halt mode.
- In case of an attempt is made to store a new message when the receive FIFO buffer and the transmit/receive FIFO buffer are full, the new message is discarded. If you wish to store a new transmit message in the transmit/receive FIFO buffer or the transmit queue, check that the transmit/receive FIFO buffer or the transmit queue is not full.
- In the case of registers that access the RAM, the value after reset shown in **Section 20.3, Registers (Classical CAN Mode)** and **Section 20.4, Registers (CAN FD Mode)** indicate the values cleared by initialization of the CAN RAM. Values before clearing are undefined. The following registers apply.
 - Receive rule (RSCFDn(CFD)GAFLIDj, RSCFDn(CFD)GAFLMj, RSCFDn(CFD)GAFLP0_j, RSCFDn(CFD)GAFLP1_j registers)
 - Receive buffers (RSCFDn(CFD)RMIDq, RSCFDn(CFD)RMPTRq, RSCFDnCFDRMFDSTSq, RSCFDn(CFD)RMDfb_q registers)

- Receive FIFO buffer access registers (RSCFDn(CFD)RFIDx, RSCFDn(CFD)RFPTRx, RSCFDnCFDRFFDSTSx, and RSCFDn(CFD)RFDFd_x registers)
- Transmit/receive FIFO buffer access registers (RSCFDn(CFD)CFIDk, RSCFDn(CFD)CFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDn(CFD)CFDFd_k registers)
- Transmit buffers (RSCFDn(CFD)TMIDp, RSCFDn(CFD)TMPTRp, RSCFDnCFDTMFDCTRp, and RSCFDn(CFD)TMDfb_p registers)
- Transmit history access register (RSCFDn(CFD)THLACCm register)
- RAM test page access register (RSCFDn(CFD)RPGACCr register)
- The values of unused receive buffers (the RSCFDn(CFD)RMIDq, RSCFDn(CFD)RMPTRq, RSCFDnCFDRMFDSTSq, and RSCFDn(CFD)RMDfb_q registers), receive FIFO buffer access registers (the RSCFDn(CFD)RFIDx, RSCFDn(CFD)RFPTRx, RSCFDnCFDRFFDSTSx, and RSCFDn(CFD)RFDFd_x registers) and transmit/receive FIFO buffer access registers (RSCFDn(CFD)CFIDk, RSCFDn(CFD)CFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDn(CFD)CFDFd_k registers) are undefined when the RS-CANFD module transitions to global operation mode or global test mode after exiting from global reset mode.

Section 21 FlexRay (FLXA)

This section contains a generic description of the FlexRay (FLXA).

The first part of this section describes the features specified to this product, such as the number of units and register base addresses.

The remainder of this section describes the FLXA functions and registers.

21.1 Features of FLXA

21.1.1 Number of Units

This microcontroller has the following number of FLXA units.

Table 21.1 Number of Units

Product Name	RH850/E2x-FCC1 Series	
	373 Pins	292 Pins
Number of Units	1	1
Name	FLXAn (n = 0)	FLXAn (n = 0)

Table 21.2 Unit Configurations and Channels

Unit Name	No. of Channels	RH850/E2x-FCC1 Series	
		373 Pins	292 Pins
FLXA0	2 (A ch, B ch)	✓	✓

Table 21.3 Index

Index	Description
n	Throughout this section, the individual FLXA units are identified by the index "n"; For example, the FlexRay control register of FLXAn is described as FLXAnFROC.

21.1.2 Register Base Address

The FLXAn base addresses are listed in the following table.

FLXAn register addresses are given as offsets from the base addresses in general.

Table 21.4 Register Base Address

Base Address Name	Base address	Bus Group
<FLXA0_base>	1002 0000 _H	H-Bus Group 1

21.1.3 Clock Supply

The FLXAn clock supply is shown in the following table.

Table 21.5 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
FLXAn	Bus clock	CLK_HBUS
	FlexRay sample clock	CLK_HSB

21.1.4 Interrupt Requests

The FLXAn interrupt requests are listed in the following table.

Table 21.6 Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
FLXA0					
INTFLXA0LINE0	INTFLXAnLINE0 (n = 0)	Universal interrupt for FLXA0	486	—	—
INTFLXA0LINE1	INTFLXAnLINE1 (n = 0)	Universal interrupt for FLXA0	487	—	—
INTFLXA0TIM0	INTFLXAnTIM0 (n = 0)	Timer 0 interrupt for FLXA0	488	—	—
INTFLXA0TIM1	INTFLXAnTIM1 (n = 0)	Timer 1 interrupt for FLXA0	489	—	—
INTFLXA0TIM2	INTFLXAnTIM2 (n = 0)	Timer 2 interrupt for FLXA0	490	—	—
INTFLXA0FDA	INTFLXAnFDA (n = 0)	FIFO data available interrupt for FLXA0	491	—	—
INTFLXA0FW	INTFLXAnFW (n = 0)	FIFO transfer warning interrupt for FLXA0	492	—	—
INTFLXA0OW	INTFLXAnOW (n = 0)	Output transfer warning interrupt for FLXA0	493	—	—
INTFLXA0OT	INTFLXAnOT (n = 0)	Output transfer done interrupt for FLXA0	494	—	—
INTFLXA0IQF	INTFLXAnIQF (n = 0)	Input queue full interrupt for FLXA0	495	—	—
INTFLXA0IQE	INTFLXAnIQE (n = 0)	Input queue empty interrupt for FLXA0	496	—	—

21.1.5 Reset Sources

FLXAn reset sources are listed in the following table. FLXAn is initialized by the following reset signal.

Table 21.7 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
FLXAn	All registers	✓	✓	✓	✓	—	—

21.1.6 External Input/Output Signals

The external input/output signals of FLXAn are listed below.

Table 21.8 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
FLXA0		
FLXA0RXDA	Channel A receive data input	FR0RXDA
FLXA0RXDB	Channel B receive data input	FR0RXDB
FLXA0TXDA	Channel A transmit data output	FR0TXDA
FLXA0TXDB	Channel B transmit data output	FR0TXDB
$\overline{\text{FLXA0TXENA}}$	Channel A transmit enable	$\overline{\text{FR0ENA}}$
$\overline{\text{FLXA0TXENB}}$	Channel B transmit enable	$\overline{\text{FR0ENB}}$

CAUTION

This AC specification is applied to the specific pin groups only. For details, please refer to Appendix file "Limited_conditions_for_AC_specification.xlsx".

21.2 Overview

21.2.1 Functional Overview

For communication on a FlexRay network, individual message buffers with up to 254 data bytes are configurable. The message storage consists of a single-ported Message RAM that holds up to 128 message buffers. All functions concerning the handling of messages are implemented in the message handler. Those functions are the acceptance filtering, the transfer of messages between the two FlexRay channel protocol controllers and the message RAM, maintaining the transmission schedule as well as providing message status information.

The register set of the FlexRay IP module can be accessed directly by an external host via the module's host interface. These registers are used to control/configure/monitor the FlexRay channel protocol controllers, message handler, global time unit, system universal control, frame and symbol processing, network management, and interrupt control, to access the message RAM via input/output buffer, and to control the data transfer between the message RAM and the Local RAM/Cluster RAM.

The FlexRay IP module supports the following features:

Item	Specification
Communication	Conformance with FlexRay protocol specification v2.1
Data transfer rate	Up to 10 Mbit/s on each channel
FlexRay channels	2 (channels A and B)
Message buffer	Up to 128 message buffers configurable
	Configuration of message buffers with different payload lengths possible
	Each message buffer can be configured as receive buffer, as transmit buffer or as part of the receive FIFO
	Filtering for slot counter, cycle counter, and channel
Message RAM	8 KB of message RAM for storage, for example:
	128 message buffers with max. 48 byte data section
	Up to 30 message buffers with 254 byte data section
FIFO	One configurable receive FIFO
Message buffer access	By host CPU via input and output buffer Input buffer: Holds message to be transferred to the message RAM Output buffer: Holds message read from the message RAM
	By data transfer function Input transfer: Message buffer content is transferred from Local RAM/Cluster RAM to message RAM on CPU request Output transfer: Message buffer content is transferred from message RAM to Local RAM/Cluster RAM automatically
Network management	Automatic HW support
Interrupts	Maskable module interrupts
Timer	Two absolute timers (Timer0, Timer2) One relative timer (Timer1) One stop watch timer

21.2.2 Block Diagram

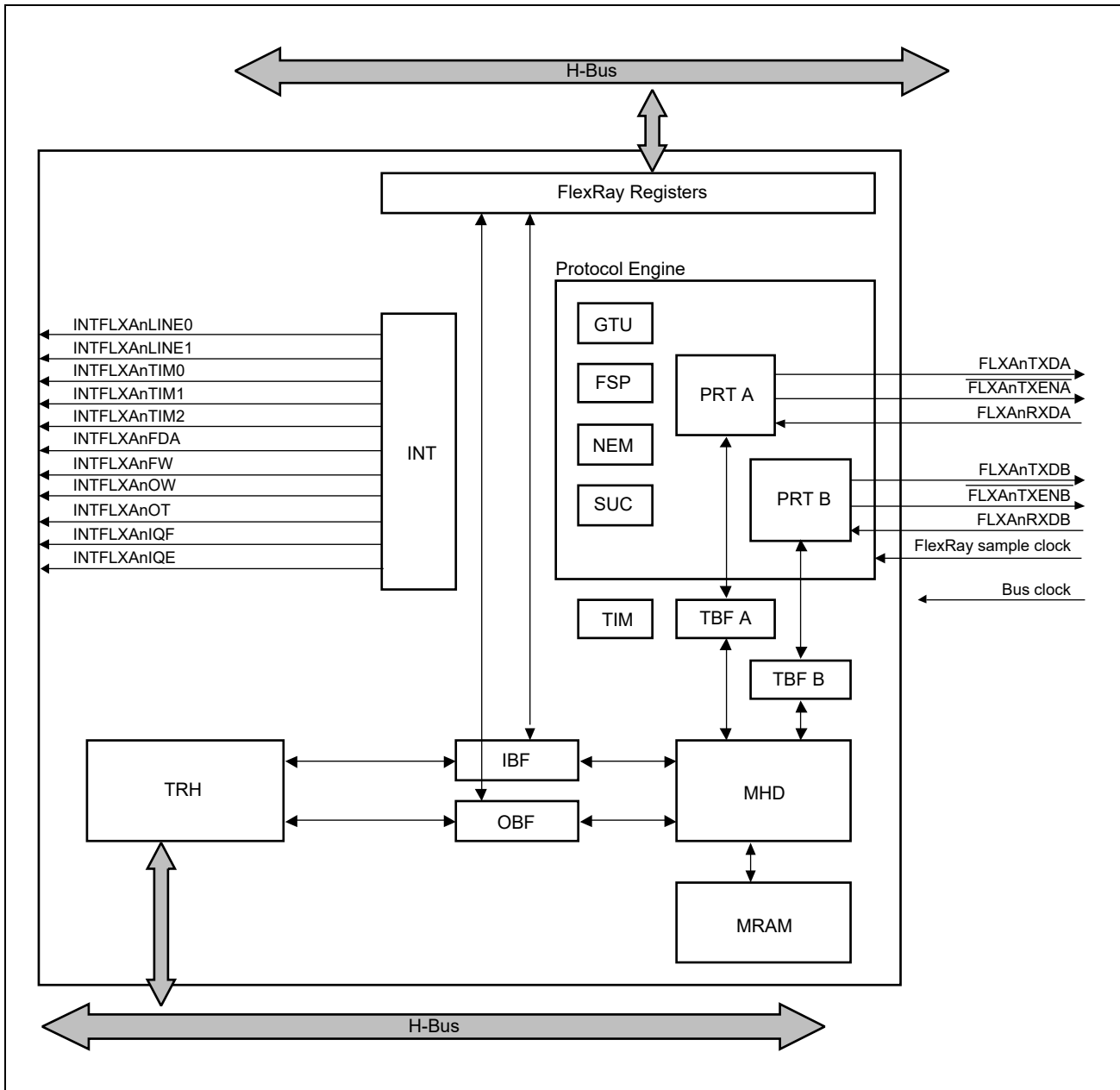


Figure 21.1 FlexRay IP Block Diagram

Input Buffer (IBF)

For write access to the message buffers configured in the message RAM, the host can write the header and data section for a specific message buffer to the input buffer. The message handler then transfers the data from the input buffer to the selected message buffer in the message RAM.

Output Buffer (OBF)

For read access to a message buffer configured in the message RAM, the message handler transfers the selected message buffer to the output buffer. After the transfer has completed, the host can read the header and data section of the transferred message buffer from the output buffer.

Message Handler (MHD)

The FlexRay message handler controls data transfers between the following components:

- Input/output buffer and message RAM
- Transient Buffer RAMs of the two FlexRay protocol controllers and message RAM

Message RAM (MRAM)

The message RAM consists of a single-ported RAM that stores up to 128 FlexRay message buffers together with the related configuration data (header and data partition).

Transient Buffer RAM (TBF A/B)

Stores the data section of two complete messages.

FlexRay Channel Protocol Controllers (PRT A/B)

The FlexRay channel protocol controllers consist of shift register and FlexRay protocol FSM. They are connected to the Transient Buffer RAMs for intermediate message storage and to the physical layer via bus driver (BD).

They have the following functionality:

- Control and check of bit timing
- Reception/transmission of FlexRay frames and symbols
- Check of header CRC
- Generation/checking of frame CRC
- Interfacing to bus driver

Global Time Unit (GTU)

The global time unit performs the following functions:

- Generation of microtick
- Generation of macrotick
- Fault tolerant clock synchronization by FTM algorithm
 - Rate correction
 - Offset correction
- Cycle counter
- Timing control of static segment
- Timing control of dynamic segment (minislottling)
- Support of external clock correction

System Universal Controller (SUC)

The system universal controller controls the following functions:

- Configuration
- Wakeup
- Startup
- Normal Operation
- Passive Operation

Frame and Symbol Processing (FSP)

The frame and symbol processing controls the following functions:

- Checks the correct timing of frames and symbols
- Tests the syntactical and semantical correctness of received frames
- Sets the slot status flags

Network Management (NEM)

Handles the network management vector.

Interrupt Control (INT)

The interrupt controller performs the following functions:

- Provides error and status interrupt flags
- Enable/disable interrupt sources
- Assignment of interrupt sources to one of the two general module interrupt lines
- Enable/disable module interrupt lines

Timer (TIM)

The timer module includes the following macrotick timers:

- Two absolute timers
- One relative timer
- One stop watch timer

Transfer Handler (TRH)

Handles the data transfer between Local RAM/Cluster RAM and FlexRay module.

The Transfer Handler supports the following transfer types:

- Transfer of buffer configuration data from the Local RAM/Cluster RAM to the message RAM
- Transfer of payload data for transmit buffer from the Local RAM/Cluster RAM to the message RAM
- Transfer of buffer configuration data and payload data for transmit buffer from the Local RAM/Cluster RAM to the message RAM
- Automatic transfer of payload data from receive buffer to the Local RAM/Cluster RAM upon frame reception
- Automatic transfer of payload data, buffer configuration data and message buffer status data from receive buffer to the Local RAM/Cluster RAM upon frame reception
- Automatic transfer of buffer configuration data and message buffer status data from the dedicated transmit/receive buffer to the Local RAM/Cluster RAM in response to slot status update
- Manual transfer of payload data, buffer configuration data and message buffer status data from the dedicated transmit/receive buffer to the Local RAM/Cluster RAM

The terms and abbreviations used in this section are given below.

Term	Meaning
BD	Bus driver
BSS	Byte start sequence
CAS	Collision Avoidance Symbol
CC	Communication controller
CHI	Controller host interface
CIF	Host interface
CRC	CRC check (Cyclic Redundancy Check)
FES	Frame end sequence
FSS	Frame Start Sequence
FIFO	First In First Out
FSM	Finite State Machine
FSP	Frame and Symbol Processing Block
FTM	Fault Tolerant Midpoint
GIF	Generic Interface Block
GTU	Global Time Unit Block)
IBF	Input Buffer
INT	Interrupt Control Block
MBF	Message Buffer RAM
MHD	Message Handler Block
MT	Macrotick
MTS	Media Access Test Symbol
NCT	Network Communication Time
NEM	Network Management Block
NIT	Network Idle Time
NM	Network Management
OBF	Output Buffer
PE	Protocol Engine
POC	Protocol Operation Control
PRT	Protocol Controller Block
SDL	Specification and Description Language
SUC	System Universal Control Block
TBF	Transient Buffer
TDMA	Time Division Multiple Access
TRH	Transfer Handler
TSS	Transmission Start Sequence
TT-D	Time Triggered Distributed synchronization
μ T	Microtick
WUP	Wakeup Pattern
WUS	Wakeup Symbol

21.3 Registers

21.3.1 List of Registers

The FLXAn registers are listed in the following table.

For details on <FLXAn_base >, see **Section 21.1.2, Register Base Address**.

Table 21.9 List of Registers (1/4)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection
FLXAn	FlexRay Operation Control Register	FLXAnFROC	<FLXAn_base> + 0004 _H	8, 16, 32	—
	FlexRay Operation Status Register	FLXAnFROS	<FLXAn_base> + 000C _H	8, 16, 32	—
	FlexRay Lock Register	FLXAnFRLCK	<FLXAn_base> + 001C _H	8, 16, 32	—
	FlexRay Error Interrupt Register	FLXAnFREIR	<FLXAn_base> + 0020 _H	8, 16, 32	—
	FlexRay Status Interrupt Register	FLXAnFRSIR	<FLXAn_base> + 0024 _H	8, 16, 32	—
	FlexRay Error Interrupt Line Select Register	FLXAnFREILS	<FLXAn_base> + 0028 _H	8, 16, 32	—
	FlexRay Status Interrupt Line Select Register	FLXAnFRSILS	<FLXAn_base> + 002C _H	8, 16, 32	—
	FlexRay Error Interrupt Enable Set Register	FLXAnFREIES	<FLXAn_base> + 0030 _H	8, 16, 32	—
	FlexRay Error Interrupt Enable Reset Register	FLXAnFREIER	<FLXAn_base> + 0034 _H	8, 16, 32	—
	FlexRay Status Interrupt Enable Set Register	FLXAnFRSIES	<FLXAn_base> + 0038 _H	8, 16, 32	—
	FlexRay Status Interrupt Enable Reset Register	FLXAnFRSIER	<FLXAn_base> + 003C _H	8, 16, 32	—
	FlexRay Interrupt Line Enable Register	FLXAnFRILE	<FLXAn_base> + 0040 _H	8, 16, 32	—
	FlexRay Timer 0 Configuration Register	FLXAnFRT0C	<FLXAn_base> + 0044 _H	8, 16, 32	—
	FlexRay Timer 1 Configuration Register	FLXAnFRT1C	<FLXAn_base> + 0048 _H	8, 16, 32	—
	FlexRay Stop Watch Register 1	FLXAnFRSTPW1	<FLXAn_base> + 004C _H	8, 16, 32	—
	FlexRay Stop Watch Register 2	FLXAnFRSTPW2	<FLXAn_base> + 0050 _H	8, 16, 32	—
	FlexRay SUC Configuration Register 1	FLXAnFRSUCC1	<FLXAn_base> + 0080 _H	8, 16, 32	—
	FlexRay SUC Configuration Register 2	FLXAnFRSUCC2	<FLXAn_base> + 0084 _H	8, 16, 32	—
	FlexRay SUC Configuration Register 3	FLXAnFRSUCC3	<FLXAn_base> + 0088 _H	8, 16, 32	—
	FlexRay NEM Configuration Register	FLXAnFRNEMC	<FLXAn_base> + 008C _H	8, 16, 32	—
	FlexRay PRT Configuration Register 1	FLXAnFRPRTC1	<FLXAn_base> + 0090 _H	8, 16, 32	—
	FlexRay PRT Configuration Register 2	FLXAnFRPRTC2	<FLXAn_base> + 0094 _H	8, 16, 32	—
	FlexRay MHD Configuration Register	FLXAnFRMHDC	<FLXAn_base> + 0098 _H	8, 16, 32	—
	FlexRay GTU Configuration Register 1	FLXAnFRGTUC1	<FLXAn_base> + 00A0 _H	8, 16, 32	—
	FlexRay GTU Configuration Register 2	FLXAnFRGTUC2	<FLXAn_base> + 00A4 _H	8, 16, 32	—
	FlexRay GTU Configuration Register 3	FLXAnFRGTUC3	<FLXAn_base> + 00A8 _H	8, 16, 32	—
	FlexRay GTU Configuration Register 4	FLXAnFRGTUC4	<FLXAn_base> + 00AC _H	8, 16, 32	—
	FlexRay GTU Configuration Register 5	FLXAnFRGTUC5	<FLXAn_base> + 00B0 _H	8, 16, 32	—
	FlexRay GTU Configuration Register 6	FLXAnFRGTUC6	<FLXAn_base> + 00B4 _H	8, 16, 32	—
	FlexRay GTU Configuration Register 7	FLXAnFRGTUC7	<FLXAn_base> + 00B8 _H	8, 16, 32	—
	FlexRay GTU Configuration Register 8	FLXAnFRGTUC8	<FLXAn_base> + 00BC _H	8, 16, 32	—
	FlexRay GTU Configuration Register 9	FLXAnFRGTUC9	<FLXAn_base> + 00C0 _H	8, 16, 32	—
FlexRay GTU Configuration Register 10	FLXAnFRGTUC10	<FLXAn_base> + 00C4 _H	8, 16, 32	—	

Table 21.9 List of Registers (2/4)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection
FLXAn	FlexRay GTU Configuration Register 11	FLXAnFRGTUC11	<FLXAn_base> + 00C8 _H	8, 16, 32	—
	FlexRay CC Status Vector Register	FLXAnFRCCSV	<FLXAn_base> + 0100 _H	8, 16, 32	—
	FlexRay CC Error Vector Register	FLXAnFRCEV	<FLXAn_base> + 0104 _H	8, 16, 32	—
	FlexRay Slot Counter Value Register	FLXAnFRSCV	<FLXAn_base> + 0110 _H	8, 16, 32	—
	FlexRay Macrotick and Cycle Counter Value Register	FLXAnFRMTCCV	<FLXAn_base> + 0114 _H	8, 16, 32	—
	FlexRay Rate Correction Value Register	FLXAnFRRCV	<FLXAn_base> + 0118 _H	8, 16, 32	—
	FlexRay Offset Correction Value Register	FLXAnFROCV	<FLXAn_base> + 011C _H	8, 16, 32	—
	FlexRay Sync Frame Status Register	FLXAnFRSFS	<FLXAn_base> + 0120 _H	8, 16, 32	—
	FlexRay Symbol Window and NIT Status Register	FLXAnFRSWNIT	<FLXAn_base> + 0124 _H	8, 16, 32	—
	FlexRay Aggregated Channel Status Register	FLXAnFRACS	<FLXAn_base> + 0128 _H	8, 16, 32	—
	FlexRay Even Sync ID Register m (m = 1 to 15)	FLXAnFRESIDm (m = 1 to 15)	<FLXAn_base> + 0130 _H to <FLXAn_base> + 0168 _H (<FLXAn_base> + 0130 _H + (m-1) × 4)	8, 16, 32	—
	FlexRay Odd Sync ID Register m (m = 1 to 15)	FLXAnFROSIDm (m = 1 to 15)	<FLXAn_base> + 0170 _H to <FLXAn_base> + 01A8 _H (<FLXAn_base> + 0170 _H + (m-1) × 4)	8, 16, 32	—
	FlexRay Network Management Vector Register m (m = 1 to 3)	FLXAnFRNMVm (m = 1 to 3)	<FLXAn_base> + 01B0 _H to <FLXAn_base> + 01B8 _H (<FLXAn_base> + 01B0 _H + (m-1) × 4)	8, 16, 32	—
	FlexRay Message RAM Configuration Register	FLXAnFRMRC	<FLXAn_base> + 0300 _H	8, 16, 32	—
	FlexRay FIFO Rejection Filter Register	FLXAnFRFRF	<FLXAn_base> + 0304 _H	8, 16, 32	—
	FlexRay FIFO Rejection Filter Mask Register	FLXAnFRFRFM	<FLXAn_base> + 0308 _H	8, 16, 32	—
	FlexRay FIFO Critical Level Register	FLXAnFRFCL	<FLXAn_base> + 030C _H	8, 16, 32	—
	FlexRay Message Handler Status Register	FLXAnFRMHDS	<FLXAn_base> + 0310 _H	8, 16, 32	—
	FlexRay Last Dynamic Transmit Slot Register	FLXAnFRLDTS	<FLXAn_base> + 0314 _H	8, 16, 32	—
	FlexRay FIFO Status Register	FLXAnFRFSR	<FLXAn_base> + 0318 _H	8, 16, 32	—
FlexRay Message Handler Constraints Flags Register	FLXAnFRMHDF	<FLXAn_base> + 031C _H	8, 16, 32	—	
FlexRay Transmission Request Register i (i = 1 to 4)	FLXAnFRTXRQi (i = 1 to 4)	<FLXAn_base> + 0320 _H to <FLXAn_base> + 032C _H (<FLXAn_base> + 0320 _H + (i-1) × 4)	8, 16, 32	—	
FlexRay New Data Register i (i = 1 to 4)	FLXAnFRNDATi (i = 1 to 4)	<FLXAn_base> + 0330 _H to <FLXAn_base> + 033C _H (<FLXAn_base> + 0330 _H + (i-1) × 4)	8, 16, 32	—	

Table 21.9 List of Registers (3/4)

Module Name	Register Name	Symbol	Address	Access Size	Access protection
FLXAn	FlexRay Message Buffer Status Changed Register i (i = 1 to 4)	FLXAnFRMBSCi (i = 1 to 4)	<FLXAn_base> + 0340 _H to <FLXAn_base> + 034C _H (<FLXAn_base> + 0340 _H + (i-1) × 4)	8, 16, 32	—
	FlexRay Write Data Section Register x (x = 1 to 64)	FLXAnFRWRDSx (x = 1 to 64)	<FLXAn_base> + 0400 _H to <FLXAn_base> + 04FC _H (<FLXAn_base> + 0400 _H + (x-1) × 4)	8, 16, 32	—
	FlexRay Write Header Section Register 1	FLXAnFRWRHS1	<FLXAn_base> + 0500 _H	8, 16, 32	—
	FlexRay Write Header Section Register 2	FLXAnFRWRHS2	<FLXAn_base> + 0504 _H	8, 16, 32	—
	FlexRay Write Header Section Register 3	FLXAnFRWRHS3	<FLXAn_base> + 0508 _H	8, 16, 32	—
	FlexRay Input Buffer Command Mask Register	FLXAnFRIBCM	<FLXAn_base> + 0510 _H	8, 16, 32	—
	FlexRay Input Buffer Command Request Register	FLXAnFRIBCR	<FLXAn_base> + 0514 _H	8, 16, 32	—
	FlexRay Read Data Section Register x (x = 1 to 64)	FLXAnFRRDDs (x = 1 to 64)	<FLXAn_base> + 0600 _H to <FLXAn_base> + 06FC _H (<FLXAn_base> + 0600 _H + (x-1) × 4)	8, 16, 32	—
	FlexRay Read Header Section Register 1	FLXAnFRRDHS1	<FLXAn_base> + 0700 _H	8, 16, 32	—
	FlexRay Read Header Section Register 2	FLXAnFRRDHS2	<FLXAn_base> + 0704 _H	8, 16, 32	—
	FlexRay Read Header Section Register 3	FLXAnFRRDHS3	<FLXAn_base> + 0708 _H	8, 16, 32	—
	FlexRay Message Buffer Status Register	FLXAnFRMBS	<FLXAn_base> + 070C _H	8, 16, 32	—
	FlexRay Output Buffer Command Mask Register	FLXAnFROBCM	<FLXAn_base> + 0710 _H	8, 16, 32	—
	FlexRay Output Buffer Command Request Register	FLXAnFROBCR	<FLXAn_base> + 0714 _H	8, 16, 32	—
	FlexRay Input Transfer Configuration Register	FLXAnFRITC	<FLXAn_base> + 0800 _H	8, 16, 32	—
	FlexRay Output Transfer Configuration Register	FLXAnFROTC	<FLXAn_base> + 0804 _H	8, 16, 32	—
	FlexRay Input Pointer Table Base Address Register	FLXAnFRIBA	<FLXAn_base> + 0808 _H	8, 16, 32	—
	FlexRay FIFO Pointer Table Base Address Register	FLXAnFRFBA	<FLXAn_base> + 080C _H	8, 16, 32	—
	FlexRay Output Pointer Table Base Address Register	FLXAnFROBA	<FLXAn_base> + 0810 _H	8, 16, 32	—
	FlexRay Input Queue Control Register	FLXAnFRIQC	<FLXAn_base> + 0814 _H	8, 16, 32	—
	FlexRay User Input Transfer Request Register	FLXAnFRUIR	<FLXAn_base> + 0818 _H	8, 16, 32	—
	FlexRay User Output Transfer Request Register	FLXAnFRUOR	<FLXAn_base> + 081C _H	8, 16, 32	—
	FlexRay Input Transfer Status Register	FLXAnFRITS	<FLXAn_base> + 0820 _H	8, 16, 32	—
	FlexRay Output Transfer Status Register	FLXAnFROTS	<FLXAn_base> + 0824 _H	8, 16, 32	—

Table 21.9 List of Registers (4/4)

Module Name	Register Name	Symbol	Address	Access Size	Access protection
FLXAn	FlexRay Access Error Status Register	FLXAnFRAES	<FLXAn_base> + 0828 _H	8, 16, 32	—
	FlexRay Access Error Address Register	FLXAnFRAEA	<FLXAn_base>+ 082C _H	8, 16, 32	—
	FlexRay Message Data Available Register i (i = 0 to 3)	FLXAnFRDAi (i = 0 to 3)	<FLXAn_base> + 0830 _H to <FLXAn_base> + 083C _H (<FLXAn_base> + 0830 _H + I × 4)	8, 16, 32	—
	FlexRay Timer 2 Configuration Register	FLXAnFRT2C	<FLXAn_base> + 0844 _H	8, 16, 32	—

21.3.2 FlexRay Operation Register

21.3.2.1 FLXAnFROC — FlexRay Operation Control Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	T2IE	T1IE	T0IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OEP	—	—	—	—	—	BEC	OE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W

Table 21.10 FLXAnFROC Register Contents

Bit Position	Bit Name	Function
31 to 19	—	Reserved These bits are always read as 0. When writing, always write 0.
18	T2IE	Timer 2 Interrupt Enable Bit 0: Disabled 1: Enabled
17	T1IE	Timer 1 Interrupt Enable Bit 0: Disabled 1: Enabled
16	T0IE	Timer 0 Interrupt Enable Bit 0: Disabled 1: Enabled
15 to 8	—	Reserved These bits are always read as 0. When writing, always write 0.
7	OEP	Operation Enable Bit Protection Bit 0: OE bit is unprotected 1: OE bit is protected
6 to 2	—	Reserved These bits are always read as 0. When writing, always write 0.
1	BEC	Byte Endian Control Bit 0: Little endian 1: Big endian
0	OE	Operation Enable Bit 0: Operation stopped SW reset 1: Operation Enabled

(1) FLXAnFROC.T2IE

Timer 2 interrupt enable bit

This bit controls the timer 2 interrupt.

0: Disabled

No interrupt will be requested and the timer 2 interrupt line will be released if held pending.

1: Enabled

Timer 2 interrupt will be asserted when FLXAnFROTS.T2IS is '1'.

(2) FLXAnFROC.T1IE

Timer 1 interrupt enable bit

The user should only set this bit to '1' when timer 1 interrupt is not enabled in the FlexRay Status interrupt enable register (FLXAnFRSIES.T1IE should be '0').

This bit controls the timer 1 interrupt.

0: Disabled

No interrupt will be requested and the timer 1 interrupt line will be released if held pending.

1: Enabled

Timer 1 interrupt will be asserted when FLXAnFROTS.T1IS is '1'.

(3) FLXAnFROC.T0IE

Timer 0 interrupt enable bit

The user should only set this bit to '1' when timer 0 interrupt is not enabled in the FlexRay Status interrupt enable register (FLXAnFRSIES.TI0E should be '0').

This bit controls the timer 0 interrupt.

0: Disabled

No interrupt will be requested and the timer 0 interrupt line will be released if held pending.

1: Enabled

Timer 0 interrupt will be asserted when FLXAnFROTS.T0IS is '1'.

(4) FLXAnFROC.OEP

Operation enable bit protection bit

This bit protects against unintended write access to the OE bit.

0: OE bit is unprotected

Write access to the OE bit is enabled

1: OE bit is protected

Write access to the OE bit is disabled

(5) FLXAnFROC.BEC

Byte endian control bit

The user should only change this bit when FLXAnFROS.OS is '1'.

This bit controls the byte order on reading and writing the FlexRay network management vector register (FLXAnFRNMVx), FlexRay write data section (FLXAnFRWRDSx) and FlexRay read data section (FLXAnFRRDDSx). This bit also controls the byte order when reading or writing FlexRay payload data using the data transfer function.

For details about the byte alignment, please refer to **Section 21.4.17, Byte Alignment**.

0: Little endian

Byte alignment in FLXAnFRNMVx, FLXAnFRWRDSx and FLXAnFRRDDSx is in little endian style.

1: Big endian

Byte alignment in FLXAnFRNMVx, FLXAnFRWRDSx and FLXAnFRRDDSx is in big endian style.

(6) FLXAnFROC.OE

Operation enable bit

The user can only write to this bit when FLXAnFROC.OEP is '0'.

The user should only write this bit with '0' when FLXAnFROS.OS is '1'.

The user should only write this bit with '1' when FLXAnFROS.OS is '0' and the FlexRay sample clock is enabled.

This bit controls the operation state and serves the software reset of the FlexRay module. The operation status bit (FLXAnFROS.OS) indicates whether or not the FlexRay module is in reset state.

0: Operation stopped SW reset

Forcibly moves the FlexRay module to its reset state, whatever the state of the FlexRay module is.

1: Operation Enabled

Reset state of the FlexRay module is released.

21.3.2.2 FLXAnFROS — FlexRay Operation Status Register

Do not rewrite this register using bit manipulation instructions.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	T2IS	T1IS	T0IS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.11 FLXAnFROS Register Contents

Bit Position	Bit Name	Function
31 to 19	—	Reserved These bits are always read as 0. When writing, always write 0.
18	T2IS	Timer 2 Interrupt Status Bit 0: Timer 2 has not matched the conditions configured in the FLXAnFRT2C register 1: Timer 2 matched the conditions configured in the FLXAnFRT2C register
17	T1IS	Timer 1 Interrupt Status Bit 0: Timer 1 has not matched the conditions configured in the FLXAnFRT1C register 1: Timer 1 matched the conditions configured in the FLXAnFRT1C register
16	T0IS	Timer 0 Interrupt Status Bit 0: Timer 0 has not matched the conditions configured in the FLXAnFRTOC register 1: Timer 0 matched the conditions configured in the FLXAnFRTOC register
15 to 1	—	Reserved These bits are always read as 0. When writing, always write 0.
0	OS	Operation Status Bit 0: Operation disabled, reset state 1: Operation enabled

(1) FLXAnFROS.T2IS

Timer 2 Interrupt Status Bit

Writing '0' has no effect on the bit value.

This bit represents that Timer 2 has matched the state configured in the FLXAnFRT2C register.

If enabled in FLXAnFROC.T2IE the timer 2 interrupt is generated when FLXAnFROS.T2IS is '1'.

[Clearing condition]

This bit is cleared by writing '1' to FLXAnFROS.T2IS.

This bit is cleared when FLXAnFROS.OS changes from '1' to '0'.

[Setting condition]

This bit is set when Timer 2 matches the state configured in FLXAnFRT2C register.

(2) FLXAnFROS.T1IS

Timer 1 Interrupt Status Bit

Writing '0' has no effect on the bit value.

This bit indicates that Timer 1 has matched the state configured in FLXAnFRT1C register.

If enabled in FLXAnFROC.T1IE, the timer 1 interrupt is generated when FLXAnFROS.T1IS is '1'.

[Clearing condition]

This bit is cleared by writing '1' to FLXAnFROS.T1IS.

This bit is cleared when FLXAnFROS.OS changes from '1' to '0'.

[Setting condition]

This bit is set when Timer 1 matches the state configured in the FLXAnFRT1C register.

(3) FLXAnFROS.T0IS

Timer 0 Interrupt Status Bit

Writing '0' has no effect on the bit value.

This bit represents that the Timer 0 has matched the state configured in the FLXAnFRT0C register.

If enabled in FLXAnFROC.T0IE, the timer 1 interrupt is generated when FLXAnFROS.T0IS is '1'.

[Clearing condition]

This bit is cleared by writing '1' to FLXAnFROS.T0IS.

This bit is cleared when FLXAnFROS.OS changes from '1' to '0'.

[Setting condition]

This bit is set when Timer 0 matches the state configured in the FLXAnFRT0C register.

(4) FLXAnFROS.OS

Operation Status Bit

This bit indicates whether the FlexRay module is in the reset or the operation state.

When FLXAnFROS.OS is '0', the FlexRay module gets initialized and registers mapped to the address area $\langle \text{FLXAn_base} \rangle + 0010_{\text{H}}$ to $\langle \text{FLXAn_base} \rangle + 0\text{FFF}_{\text{H}}$ cannot be accessed; read access from these registers will return undefined data.

When FLXAnFROS.OS is '1', it is possible to access the address area $\langle \text{FLXAn_base} \rangle + 0010_{\text{H}}$ to $\langle \text{FLXAn_base} \rangle + 0\text{FFF}_{\text{H}}$ and to perform FlexRay communication.

When FLXAnFROS.OS changes from '0' to '1', all registers in the address range $\langle \text{FLXAn_base} \rangle + 0010_{\text{H}}$ to $\langle \text{FLXAn_base} \rangle + 0\text{FFF}_{\text{H}}$ are set to the "Values after reset".

[Clearing condition]

When FLXAnFROC.OE is set to '0', it takes up to two peripheral bus clock cycles until FLXAnFROS.OS is set to '0'.

[Setting condition]

When FLXAnFROC.OE is set to '1', it takes up to four peripheral clock cycles of the clock with the lower frequency out of the FlexRay sample clock and peripheral bus clock until FLXAnFROS.OS is set to '1'.

21.3.3 Special Registers

21.3.3.1 FLXAnFRLCK — FlexRay Lock Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CLK[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.12 FLXAnFRLCK Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are always read as 0. When writing, always write 0.
7 to 0	CLK[7:0]	Configuration Lock Key Bit

(1) FLXAnFRLCK.CLK

Configuration Lock Key

The lock register is write-only. Reading the register will return 0000 0000_H.

To leave CONFIG state by writing FLXAnFRSUCC1.CMD (command READY), the write operation has to be directly preceded by two write accesses to the Configuration Lock Key (unlock sequence). If the write sequence below is interrupted by other write accesses between the second write to the configuration lock key and the write access to the FLXAnFRSUCC1 register, the CC remains in CONFIG state and the sequence has to be repeated.

First write: FLXAnFRLCK.CLK = “1100 1110_B” (CE_H)

Second write: FLXAnFRLCK.CLK = “0011 0001_B” (31_H)

Third write: FLXAnFRSUCC1.CMD

CAUTION

In case that the host uses 8/16-bit accesses to write the listed bit fields, the programmer has to ensure that no “dummy accesses”, e.g. to the remaining register bytes/words, are inserted by the compiler.

21.3.4 Interrupt Registers

21.3.4.1 FLXAnFREIR — FlexRay Error Interrupt Register

Do not rewrite this register using bit manipulation instructions.

The flags are set when the CC detects one of the listed error conditions. The flags remain set until the host clears them.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABB	LTVB	EDB	—	—	—	—	—	TABA	LTVA	EDA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHF	IOBA	IIBA	EFA	RFO	AERR	CCL	CCF	SFO	SFBM	CNA	PEMC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.13 FLXAnFREIR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	—	Reserved These bits are always read as 0. When writing, always write 0.
26	TABB	Transmission Across Boundary Channel B Flag 0: No transmission across slot boundary detected on channel B 1: Transmission across slot boundary detected on channel B
25	LTVB	Latest Transmit Violation Channel B Flag 0: No latest transmit violation detected on channel B 1: Latest transmit violation detected on channel B
24	EDB	Error Detected on Channel B Flag 0: No error detected on channel B 1: Error detected on channel B
23 to 19	—	Reserved These bits are always read as 0. When writing, always write 0.
18	TABA	Transmission Across Boundary Channel A Flag 0: No transmission across slot boundary detected on channel A 1: Transmission across slot boundary detected on channel A
17	LTVA	Latest Transmit Violation Channel A Flag 0: No latest transmit violation detected on channel A 1: Latest transmit violation detected on channel A
16	EDA	Error Detected on Channel A Flag 0: No error detected on channel A 1: Error detected on channel A
15 to 12	—	Reserved These bits are always read as 0. When writing, always write 0.

Table 21.13 FLXAnFREIR Register Contents (2/2)

Bit Position	Bit Name	Function
11	MHF	Message Handler Constraints Flag 0: No message handler failure detected 1: Message handler failure detected
10	IOBA	Illegal Output Buffer Access Flag 0: No illegal host access to output buffer occurred 1: Illegal host access to output buffer occurred
9	IIBA	Illegal Input Buffer Access Flag 0: No illegal host access to input buffer occurred 1: Illegal host access to input buffer occurred
8	EFA	Empty FIFO Access Flag 0: No host access to empty FIFO occurred 1: Host access to empty FIFO occurred
7	RFO	Receive FIFO Overrun Flag 0: No receive FIFO overrun detected 1: A receive FIFO overrun has been detected
6	AERR	Access Error Flag 0: Access error is not detected. 1: Access error is detected.
5	CCL	CHI Command Locked Flag 0: CHI command accepted 1: CHI command not accepted
4	CCF	Clock Correction Failure Flag 0: No clock correction error 1: Clock correction failed
3	SFO	Sync Frame Overflow Flag 0: Number of received sync frames <= FLXAnFRGTUC2.SNM 1: More sync frames received than configured by FLXAnFRGTUC2.SNM
2	SFBM	Sync Frames Below Minimum Flag 0: Sync node: 1 or more sync frames received, non-sync node: 2 or more sync frames received 1: Less than the required minimum of sync frames received
1	CNA	Command Not Accepted Flag 0: CHI command accepted 1: CHI command not accepted
0	PEMC	POC Error Mode Changed Flag 0: Error mode has not changed 1: Error mode has changed

(1) FLXAnFREIR.TABB

Transmission Across Boundary Channel B Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

The flag signals to the host that a transmission across a slot boundary occurred for channel B.

(2) FLXAnFREIR.LTVB

Latest Transmit Violation Channel B Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

The flag signals the latest transmit violation on channel B to the host.

(3) FLXAnFREIR.EDB

Error Detected on Channel B Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This bit is set whenever one of flags FLXAnFRACS.SEDB, FLXAnFRACS.CEDB, FLXAnFRACS.CIB, FLXAnFRACS.SBVB changes from '0' to '1'.

(4) FLXAnFREIR.TABA

Transmission Across Boundary Channel A Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

The flag signals to the host that a transmission across a slot boundary occurred for channel A.

(5) FLXAnFREIR.LTVA

Latest Transmit Violation Channel A Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

The flag signals the latest transmit violation on channel A to the host.

(6) FLXAnFREIR.EDA

Error Detected on Channel A Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This bit is set whenever one of flags FLXAnFRACS.SEDA, FLXAnFRACS.CEDA, FLXAnFRACS.CIA, FLXAnFRACS.SBVA changes from '0' to '1'.

(7) FLXAnFREIR.MHF

Message Handler Constraints Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

The flag signals a message handler constraints violation condition. It is set whenever one of flags FLXAnFRMHDF.SNUA, FLXAnFRMHDF.SNUB, FLXAnFRMHDF.FNFA, FLXAnFRMHDF.FNFB, FLXAnFRMHDF.TBFA, FLXAnFRMHDF.TBFB, FLXAnFRMHDF.WAHP changes from '0' to '1'.

(8) FLXAnFREIR.IOBA

Illegal Output Buffer Access Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set by the CC when the host requests the transfer of a message buffer from the message RAM to the output buffer while FLXAnFROBCR.OBSYS is set to '1'.

(9) FLXAnFREIR.IIBA

Illegal Input Buffer Access Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set by the CC when the host wants to modify a message buffer via input buffer and one of the following conditions applies:

1. The CC is not in CONFIG or DEFAULT_CONFIG state and the host writes to the input buffer command request register to modify the:
 - Header section of message buffer 0, 1 if configured for transmission in key slot
 - Header section of static message buffers with buffer number < FLXAnFRMRC.FDB while FLXAnFRMRC.SEC = "01"
 - Header section of any static or dynamic message buffer while FLXAnFRMRC.SEC = "1x"
 - Header and/or data section of any message buffer belonging to the receive FIFO
2. The host writes to any register of the input buffer while FLXAnFRIBCR.IBSYH is set to '1'.

(10) FLXAnFREIR.EFA

Empty FIFO Access Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set by the CC when the host requests the transfer of a message from the receive FIFO via output buffer while the receive FIFO is empty.

(11) FLXAnFREIR.RFO

Receive FIFO Overrun Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

The flag is set by the CC when a receive FIFO overrun is detected. When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. The actual state of the FIFO is monitored in register FLXAnFRFSR.

(12) FLXAnFREIR.AERR

Access Error Flag

Writing 0 in this bit has no effect.

This bit is cleared by writing '1' to it.

Notifies of an access error.

When the AMR, ATBF1, or ATBF2 bit in the FLXAnFRMHDS register changes from 0 to 1, this bit is set to 1.

(13) FLXAnFREIR.CCL

CHI Command Locked Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

The flag signals that the write access to the CHI command vector FLXAnFRSUCC1.CMD was not successful because

the execution of the previous CHI command has not yet completed. In this case, bit FLXAnFREIR.CNA is also set to '1'.

(14) FLXAnFREIR.CCF

Clock Correction Failure Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set at the end of the communication cycle whenever one of the following errors occurs:

- Missing offset and/or rate correction
- Clock correction limit reached

The clock correction status is monitored in registers FLXAnFRCCEV and FLXAnFRSFS. A failure may occur during startup, therefore bit FLXAnFREIR.CCF should be set to "0" after the CC enters NORMAL_ACTIVE state.

(15) FLXAnFREIR.SFO

Sync Frame Overflow Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

It is set to "1" when either the number of sync frames received during the last communication cycle or the total number of different sync frame IDs received during the last double cycle exceeds the maximum number of sync frames as defined by FLXAnFRGTUC2.SNM.

(16) FLXAnFREIR.SFBM

Sync Frames Below Minimum Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set to "1" at the end of a cycle if the number of sync frames received during the last communication cycle was below the limit required for rate or offset correction term calculation (i.e. missing offset and/or missing rate correction). The clock correction status is monitored in FLXAnFRCCEV and FLXAnFRSFS.

This flag may be set to "1" during startup. Therefore this flag should be set to "0" by the host after the CC enters NORMAL_ACTIVE state.

(17) FLXAnFREIR.CNA

Command Not Accepted Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

The flag signals that the write access to the CHI command vector FLXAnFRSUCC1.CMD was not successful because the requested command was not valid in the actual POC state, or because the CHI command was locked (CCL = '1').

(18) FLXAnFREIR.PEMC

POC Error Mode Changed Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set to "1" whenever the error mode signaled by FLXAnFRCCEV.ERRM has changed.

21.3.4.2 FLXAnFRSIR — FlexRay Status Interrupt Register

Do not rewrite this register using bit manipulation instructions.

The flags are set when the CC detects one of the listed events. The flags remain set until the host clears them.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSB	WUPB	—	—	—	—	—	—	MTSA	WUPA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDS	MBSI	SUCS	SWE	TOBC	TIBC	TI1	TI0	NMVC	RFCL	RFNE	RXI	TXI	CYCS	CAS	WST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.14 FLXAnFRSIR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	—	Reserved These bits are always read as 0. When writing, always write 0.
25	MTSB	MTS Received on Channel B Flag (vSS!ValidMTSB) 0: No MTS symbol received on channel B 1: MTS symbol received on channel B
24	WUPB	Wakeup Pattern Channel B Flag 0: No wakeup pattern received on channel B 1: Wakeup pattern received on channel B
23 to 18	—	Reserved These bits are always read as 0. When writing, always write 0.
17	MTSA	MTS Received on Channel A Flag (vSS!ValidMTSA) 0: No MTS symbol received on channel A 1: MTS symbol received on channel A
16	WUPA	Wakeup Pattern Channel A Flag 0: No wakeup pattern received on channel A 1: Wakeup pattern received on channel A
15	SDS	Start of Dynamic Segment Flag 0: Dynamic segment not yet started 1: Dynamic segment started
14	MBSI	Message Buffer Status Interrupt Flag 0: No message buffer status change of message buffer with MBI = '1' 1: Message buffer status of at least one message buffer with MBI = '1' has changed
13	SUCS	Startup Normal End Flag 0: No startup completed successfully 1: Startup completed successfully
12	SWE	Stop Watch Event Flag 0: No Stop Watch Event 1: Stop Watch Event occurred

Table 21.14 FLXAnFRSIR Register Contents (2/2)

Bit Position	Bit Name	Function
11	TOBC	Transfer Output Buffer Completed Flag 0: No transfer completed 1: Transfer between message RAM and output buffer completed
10	TIBC	Transfer Input Buffer Completed Flag 0: No transfer completed 1: Transfer between input buffer and message RAM completed
9	TI1	Timer Interrupt 1 Flag 0: No timer interrupt 1 1: Timer interrupt 1 occurred
8	TI0	Timer Interrupt 0 Flag 0: No timer interrupt 0 1: Timer interrupt 0 occurred
7	NMVC	Network Management Vector Changed Flag 0: No change in the network management vector 1: Network management vector changed
6	RFCL	Receive FIFO Critical Level Flag 0: Receive FIFO below critical level 1: Receive FIFO critical level reached
5	RFNE	Receive FIFO Not Empty Flag 0: Receive FIFO is empty 1: Receive FIFO is not empty
4	RXI	Receive Interrupt Flag 0: No ND flag of a receive buffer with MBI = '1' has been set to '1' 1: At least one ND flag of a receive buffer with MBI = 1 has been set to 1
3	TXI	Transmit Interrupt Flag 0: No frame transmitted from a transmit buffer with MBI = '1' 1: At least one frame was transmitted from a transmit buffer with MBI = '1'
2	CYCS	Cycle Start Interrupt Flag 0: No communication cycle started 1: Communication cycle started
1	CAS	Collision Avoidance Symbol Flag 0: No bit pattern matching the CAS symbol received 1: Bit pattern matching the CAS symbol received
0	WST	Wakeup Status Flag 0: Wakeup status unchanged 1: Wakeup status changed

(1) FLXAnFRSIR.MTSB

MTS Received on Channel B Flag (vSS!ValidMTSB)

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

Media Access Test Symbol (MTS) received on channel B during the preceding symbol window.

Updated by the CC for each channel at the end of the symbol window.

(2) FLXAnFRSIR.WUPB

Wakeup Pattern Channel B Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set to "1" when a wakeup pattern is received on channel B in either of the following states:

- WAKEUP
- READY
- STARTUP

(3) FLXAnFRSIR.MTSA

MTS Received on Channel A Flag (vSS!ValidMTSA)

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

Media Access Test Symbol (MTS) received on channel A during the preceding symbol window.

Updated by the CC for each channel at the end of the symbol window.

(4) FLXAnFRSIR.WUPA

Wakeup Pattern Channel A Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set to "1" when a wakeup pattern was received on channel A in either of the following states:

- WAKEUP
- READY
- STARTUP

(5) FLXAnFRSIR.SDS

Start of Dynamic Segment Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set by the CC when the dynamic segment starts.

(6) FLXAnFRSIR.MBSI

Message Buffer Status Interrupt Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set by the CC when the message buffer status FLXAnFRMBS has changed and if bit MBI of that message buffer is "1" (see **Table 21.106, Header Section of a Message Buffer in the Message RAM**).

(7) FLXAnFRSIR.SUCS

Startup Normal End Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set whenever a startup completed successfully and the CC entered NORMAL_ACTIVE state.

(8) FLXAnFRSIR.SWE

Stop Watch Event Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set after a stop watch activation when the actual cycle counter and macrotick value are stored in the Stop Watch register (see **Section 21.3.5.4, FLXAnFRSTPW1 — FlexRay Stop Watch Register 1**).

(9) FLXAnFRSIR.TOBC

Transfer Output Buffer Completed Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set whenever a transfer from the message RAM to the output buffer has completed and FLXAnFROBCR.OBSYS has been reset by the message handler.

(10) FLXAnFRSIR.TIBC

Transfer Input Buffer Completed Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set whenever a transfer from input buffer to the message RAM has completed and FLXAnFRIBCR.IBSYS has been reset by the message handler.

(11) FLXAnFRSIR.TI1

Timer Interrupt 1 Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set whenever timer 1 matches the conditions configured in register FLXAnFRT1C. INTFLXAnTIM1 is generated when the T1IE bit in the FLXAnFROC register is effective.

(12) FLXAnFRSIR.TI0

Timer Interrupt 0 Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set whenever timer 0 matches the conditions configured in register FLXAnFRT0C. INTFLXAnTIM0 is generated when the T0IE bit in the FLXAnFROC register is effective.

(13) FLXAnFRSIR.NMVC

Network Management Vector Changed Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This bit is set when a change in the Network Management Vector occurs.

(14) FLXAnFRSIR.RFCL

Receive FIFO Critical Level Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set when the receive FIFO fill level FLXAnFRFSR.RFFL is equal to or greater than the critical level as configured by FLXAnFRFCL.CL.

(15) FLXAnFRSIR.RFNE

Receive FIFO Not Empty Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set when a received valid frame is transferred to the empty receive FIFO. The current state of the receive FIFO can be read in register FLXAnFRFSR.

(16) FLXAnFRSIR.RXI

Receive Interrupt Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set whenever the set condition of a message buffers ND flag is fulfilled (see **Section 21.3.9.6, FLXAnFRNDAT_i — FlexRay New Data Register i (i = 1 to 4)**), and if bit MBI of that message buffer is set to 1 (see **Table 21.106, Header Section of a Message Buffer in the Message RAM**).

(17) FLXAnFRSIR.TXI

Transmit Interrupt Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set at the end of frame transmission if bit MBI in the respective message buffer is set to 1 (see **Table 21.106, Header Section of a Message Buffer in the Message RAM**)

(18) FLXAnFRSIR.CYCS

Cycle Start Interrupt Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set to "1" every time a communication cycle starts.

(19) FLXAnFRSIR.CAS

Collision Avoidance Symbol Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set during STARTUP state when a CAS or an expected CAS was received.

(20) FLXAnFRSIR.WST

Wakeup Status Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set when FLXAnFRCCSV.WSV changes to a value other than UNDEFINED.

21.3.4.3 FLXAnFREILS — FlexRay Error Interrupt Line Select Register

The FlexRay error interrupt line select register assigns an interrupt generated by a specific error interrupt flag from register FLXAnFREIR to one of the two module interrupt lines (INTFLXAnLINE0 or INTFLXAnLINE1).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABBL	LTVBL	EDBL	—	—	—	—	—	TABAL	LTVAL	EDAL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHFL	IOBAL	IIBAL	EFAL	RFOL	AERRL	CCLL	CCFL	SFOL	SFBML	CNAL	PEMCL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.15 FLXAnFREILS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	—	Reserved These bits are always read as 0. When writing, always write 0.
26	TABBL	Transmission Across Boundary Channel B Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
25	LTVBL	Latest Transmit Violation Channel B Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
24	EDBL	Error Detected on Channel B Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
23 to 19	—	Reserved These bits are always read as 0. When writing, always write 0.
18	TABAL	Transmission Across Boundary Channel A Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
17	LTVAL	Latest Transmit Violation Channel A Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
16	EDAL	Error Detected on Channel A Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
15 to 12	—	Reserved These bits are always read as 0. When writing, always write 0.
11	MHFL	Message Handler Constraints Flag Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1

Table 21.15 FLXAnFREILS Register Contents (2/2)

Bit Position	Bit Name	Function
10	IOBAL	Illegal Output Buffer Access Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
9	IIBAL	Illegal Input Buffer Access Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
8	EFAL	Empty FIFO Access Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
7	RFOL	Receive FIFO Overrun Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
6	AERRL	Access Error Interrupt Output Select Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
5	CCLL	CHI Command Locked Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
4	CCFL	Clock Correction Failure Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
3	SFOL	Sync Frame Overflow Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
2	SFBML	Sync Frames Sync Frame Shortfall Interrupt Output Selecting Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
1	CNAL	Command Not Accepted Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
0	PEMCL	POC Error Mode Changed Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1

21.3.4.4 FLXAnFRSILS — FlexRay Status Interrupt Line Select Register

The FlexRay status interrupt line select register assigns an interrupt generated by a specific status interrupt flag from register FLXAnFRSIR to one of the two module interrupt lines (INTFLXAnLINE0, INTFLXAnLINE1).

Value after reset: 0303 FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSBL	WUPBL	—	—	—	—	—	—	MTSAL	WUPAL
Value after reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDSL	MBSIL	SUCSL	SWEL	TOBCL	TIBCL	TI1L	TI0L	NMVCL	RFCLL	RFNEL	RXIL	TXIL	CYCSL	CASL	WSTL
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.16 FLXAnFRSILS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	—	Reserved These bits are always read as 0. When writing, always write 0.
25	MTSBL	Media Access Test Symbol Channel B Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
24	WUPBL	Wakeup Pattern Channel B Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
23 to 18	—	Reserved These bits are always read as 0. When writing, always write 0.
17	MTSAL	Media Access Test Symbol Channel A Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
16	WUPAL	Wakeup Pattern Channel A Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
15	SDSL	Start of Dynamic Segment Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
14	MBSIL	Message Buffer Status Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
13	SUCSL	Startup Completed Successfully Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
12	SWEL	Stop Watch Event Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1

Table 21.16 FLXAnFRSILS Register Contents (2/2)

Bit Position	Bit Name	Function
11	TOBCL	Transfer Output Buffer Completed Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
10	TIBCL	Transfer Input Buffer Completed Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
9	T11L	Timer Interrupt 1 Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
8	T10L	Timer Interrupt 0 Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
7	NMVCL	Network Management Vector Changed Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
6	RFCLL	Receive FIFO Critical Level Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
5	RFNEL	Receive FIFO Not Empty Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
4	RXIL	Receive Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
3	TXIL	Transmit Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
2	CYCSL	Cycle Start Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
1	CASL	Collision Avoidance Symbol Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1
0	WSTL	Wakeup Status Interrupt Line Bit 0: Interrupt assigned to INTFLXAnLINE0 1: Interrupt assigned to INTFLXAnLINE1

21.3.4.5 FLXAnFREIES — FlexRay Error Interrupt Enable Set Register

The settings in the FlexRay error interrupt enable set register (FLXAnFREIES) and FlexRay error interrupt enable reset register (FLXAnFREIER) determine which status changes in the FlexRay error interrupt register will result in an interrupt.

The enable bits are set by writing to FLXAnFREIES and reset by writing to FLXAnFREIER. Reading from both addresses will result in the same value.

Writing '0' has no effect on the bit value.

Writing '1' sets the interrupt enable bit.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABBE	LTVBE	EDBE	—	—	—	—	—	TABAE	LTVAE	EDAE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHFE	IOBAE	IIBAE	EFAE	RFOE	AERRE	CCLC	CCFE	SFOE	SFBME	CNAE	PEMCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.17 FLXAnFREIES Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	—	Reserved These bits are always read as 0. When writing, always write 0.
26	TABBE	Transmission Violation Across Boundary Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
25	LTVBE	Latest Transmit Violation Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
24	EDBE	Error Detected on Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 19	—	Reserved These bits are always read as 0. When writing, always write 0.
18	TABAE	Transmission Violation Across Boundary Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
17	LTVAE	Latest Transmit Violation Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled

Table 21.17 FLXAnFREIES Register Contents (2/2)

Bit Position	Bit Name	Function
16	EDAE	Error Detected on Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
15 to 12	—	Reserved These bits are always read as 0. When writing, always write 0.
11	MHFE	Message Handler Constraints Flag Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
10	IOBAE	Illegal Output Buffer Access Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
9	IIBAE	Illegal Input Buffer Access Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
8	EFAE	Empty FIFO Access Interrupt Enable ³ Bit 0: Interrupt disabled 1: Interrupt enabled
7	RFOE	Receive FIFO Overrun Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
6	AERRE	Access error interrupt Enable Bit 0: Interrupt is disabled. 1: Interrupt is enabled.
5	CCLE	CHI Command Locked Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
4	CCFE	Clock Correction Failure Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
3	SFOE	Sync Frame Overflow Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
2	SFBME	Sync Frames Below Minimum Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CNAE	Command Not Accepted Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
0	PEMCE	POC Error Mode Changed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled

21.3.4.6 FLXAnFREIER — FlexRay Error Interrupt Enable Reset Register

The settings in the FlexRay error interrupt enable set register (FLXAnFREIES) and FlexRay error interrupt enable reset register (FLXAnFREIER) determine which status changes in the FlexRay error interrupt register will result in an interrupt.

The enable bits are set by writing to FLXAnFREIES and reset by writing to FLXAnFREIER. Reading from both addresses will result in the same value.

Writing '0' has no effect on the bit value.

Writing '1' clears the interrupt enable bit.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABBD	LTVBD	EDBD	—	—	—	—	—	TABAD	LTVAD	EDAD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHFD	IOBAD	IIBAD	EFAD	RFOD	AERRD	CCLD	CCFD	SFOD	SFBMD	CNAD	PEMCD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.18 FLXAnFREIER Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	—	Reserved These bits are always read as 0. When writing, always write 0.
26	TABBD	Transmission Violation Across Boundary Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
25	LTVBD	Latest Transmit Violation Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
24	EDBD	Error Detected on Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 19	—	Reserved These bits are always read as 0. When writing, always write 0.
18	TABAD	Transmission Violation Across Boundary Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
17	LTVAD	Latest Transmit Violation Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled

Table 21.18 FLXAnFREIER Register Contents (2/2)

Bit Position	Bit Name	Function
16	EDAD	Error Detected on Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
15 to 12	—	Reserved These bits are always read as 0. When writing, always write 0.
11	MHFD	Message Handler Constraints Flag Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
10	IOBAD	Illegal Output Buffer Access Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
9	IIBAD	Illegal Input Buffer Access Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
8	EFAD	Empty FIFO Access Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
7	RFOD	Receive FIFO Overrun Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
6	AERRD	Access Error Interrupt Disable Bit 0: Interrupt is disabled. 1: Interrupt is enabled.
5	CCLD	CHI Command Locked Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
4	CCFD	Clock Correction Failure Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
3	SFOD	Sync Frame Overflow Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
2	SFBMD	Sync Frames Below Minimum Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CNAD	Command Not Accepted Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
0	PEMCD	POC Error Mode Changed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled

21.3.4.7 FLXAnFRSIES — FlexRay Status Interrupt Enable Set Register

The settings in the FlexRay status interrupt enable set register (FLXAnFRSIES) and FlexRay status interrupt enable reset register (FLXAnFRSIER) determine which status changes in the FlexRay status interrupt register will result in an interrupt.

The enable bits are set by writing to FLXAnFRSIES and reset by writing to FLXAnFRSIER. Reading from both addresses will result in the same value.

Writing '0' has no effect on the bit value.

Writing '1' sets the interrupt enable bit.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSBE	WUPBE	—	—	—	—	—	—	MTSAE	WUPAE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDSE	MBSIE	SUCSE	SWEE	TOBCE	TIBCE	TI1E	TI0E	NMVCE	RFCLE	RFNEE	RXIE	TXIE	CYCSE	CASE	WSTE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.19 FLXAnFRSIES Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	—	Reserved These bits are always read as 0. When writing, always write 0.
25	MTSBE	MTS Received on Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
24	WUPBE	Wakeup Pattern Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 18	—	Reserved These bits are always read as 0. When writing, always write 0.
17	MTSAE	MTS Received on Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
16	WUPAE	Wakeup Pattern Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
15	SDSE	Start of Dynamic Segment Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled

Table 21.19 FLXAnFRSIES Register Contents (2/2)

Bit Position	Bit Name	Function
14	MBSIE	Message Buffer Status Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
13	SUCSE	Startup Completed Successfully Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
12	SWEE	Stop Watch Event Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
11	TOBCE	Transfer Output Buffer Completed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
10	TIBCE	Transfer Input Buffer Completed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
9	TI1E	Timer Interrupt 1 Enable Bit 0: Interrupt disabled 1: Interrupt enabled
8	TI0E	Timer Interrupt 0 Enable Bit 0: Interrupt disabled 1: Interrupt enabled
7	NMVCE	Network Management Vector Changed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
6	RFCLE	Receive FIFO Critical Level Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
5	RFNEE	Receive FIFO Not Empty Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
4	RXIE	Receive Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
3	TXIE	Transmit Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
2	CYCSE	Cycle Start Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CASE	Collision Avoidance Symbol Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
0	WSTE	Wakeup Status Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled

21.3.4.8 FLXAnFRSIER — FlexRay Status Interrupt Enable Reset Register

The settings in the FlexRay status interrupt enable set register (FLXAnFRSIES) and FlexRay status interrupt enable reset register (FLXAnFRSIER) determine which status changes in the FlexRay status interrupt register will result in an interrupt.

The enable bits are set by writing to FLXAnFRSIES and reset by writing to FLXAnFRSIER. Reading from both addresses will result in the same value.

Writing '0' has no effect on the bit value.

Writing '1' clears the interrupt enable bit.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSBD	WUPBD	—	—	—	—	—	—	MTSAD	WUPAD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDSD	MBSID	SUCSD	SWED	TOBCD	TIBCD	TI1D	TI0D	NMVCD	RFCLD	RFNED	RXID	TXID	CYCSD	CASD	WSTD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.20 FLXAnFRSIER Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	—	Reserved These bits are always read as 0. When writing, always write 0.
25	MTSBD	MTS Received on Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
24	WUPBD	Wakeup Pattern Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 18	—	Reserved These bits are always read as 0. When writing, always write 0.
17	MTSAD	MTS Received on Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
16	WUPAD	Wakeup Pattern Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
15	SDSD	Start of Dynamic Segment Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled

Table 21.20 FLXAnFRSIER Register Contents (2/2)

Bit Position	Bit Name	Function
14	MBSID	Message Buffer Status Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
13	SUCSD	Startup Completed Successfully Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
12	SWED	Stop Watch Event Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
11	TOBCD	Transfer Output Buffer Completed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
10	TIBCD	Transfer Input Buffer Completed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
9	TI1D	Timer Interrupt 1 Disable Bit 0: Interrupt disabled 1: Interrupt enabled
8	TI0D	Timer Interrupt 0 Disable Bit 0: Interrupt disabled 1: Interrupt enabled
7	NMVCD	Network Management Vector Changed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
6	RFCLD	Receive FIFO Critical Level Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
5	RFNED	Receive FIFO Not Empty Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
4	RXID	Receive Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
3	TXID	Transmit Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
2	CYCSD	Cycle Start Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CASD	Collision Avoidance Symbol Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
0	WSTD	Wakeup Status Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled

21.3.4.9 FLXAnFRILE — FlexRay Interrupt Line Enable Register

Each of the two module interrupt lines (INTFLXAnLINE0, INTFLXAnLINE1) can be enabled or disabled separately by programming bit FLXAnFRILE.EINT0 and FLXAnFRILE.EINT1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EINT1	EINT0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 21.21 FLXAnFRILE Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved These bits are always read as 0. When writing, always write 0.
1	EINT1	Enable FlexRay Interrupt Line 1 Bit 0: INTFLXAnLINE1 disabled 1: INTFLXAnLINE1 enabled
0	EINT0	Enable FlexRay Interrupt Line 0 Bit 0: INTFLXAnLINE0 disabled 1: INTFLXAnLINE0 enabled

21.3.5 FlexRay Timer Registers

21.3.5.1 FLXAnFRT0C — FlexRay Timer 0 Configuration Register

This register is an absolute timer. It specifies the point in time when a FlexRay timer 0 interrupt occurs as the values of cycle count and macrotick (MT). When the FlexRay timer 0 expires, FLXAnFRSIR.TI0 and FLXAnFROS.T0IS are set to '1'. A timer 0 interrupt then occurs while the FLXAnFROC.T0IE bit is effective.

CAUTION

The configuration of timer 0 is compared against the macrotick counter value, there is no separate counter for timer 0.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T0MO[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	T0CC[6:0]						—	—	—	—	—	—	—	T0MS	T0RC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 21.22 FLXAnFRT0C Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved These bits are always read as 0. When writing, always write 0.
29 to 16	T0MO[13:0]	Timer 0 Macrotick Offset Bit Timer 0 macrotick offset
15	—	Reserved This bit is always read as 0. When writing, always write 0.
14 to 8	T0CC[6:0]	Timer 0 Cycle Code Bit Timer 0 cycle code
7 to 2	—	Reserved These bits are always read as 0. When writing, always write 0.
1	T0MS	Timer 0 Mode Select Bit 0: Single-shot mode 1: Continuous mode
0	T0RC	Timer 0 Run Control Bit 0: Timer 0 halted 1: Timer 0 running

(1) FLXAnFRT0C.T0MO

Timer 0 Macrotick Offset Bit

Before reconfiguration of the timer, the timer has to be halted first by writing '0' to bit FLXAnFRT0C.T0RC.

Configures the macrotick offset from the beginning of the communication cycle where the interrupt is to occur. The FlexRay timer 0 interrupt occurs at this offset for each cycle of the cycle set.

(2) FLXAnFRT0C.T0CC

Timer 0 Cycle Code Bit

Before reconfiguration of the timer, the timer has to be halted first by writing '0' to bit FLXAnFRT0C.T0RC.

The 7-bit timer 0 cycle code determines the cycle set used for generation of the FlexRay timer 0 interrupt. For details about the configuration of the cycle code, see **Section 21.4.8.2, Cycle Counter Filtering**.

(3) FLXAnFRT0C.T0MS

Timer 0 Mode Select Bit

Before reconfiguration of the timer, the timer has to be halted first by writing '0' to bit FLXAnFRT0C.T0RC.

Configures the timer run mode. In Single-shot mode, the timer is deactivated when the timer configuration matches the configured cycle counter and macrotick value.

(4) FLXAnFRT0C.T0RC

Timer 0 Run Control Bit

Timer 0 can be activated (set FLXAnFRT0C.T0RC to '1') when the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state.

Timer 0 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

21.3.5.2 FLXAnFRT1C — FlexRay Timer 1 Configuration Register

This register is a relative timer. After the specified number of macroticks (MT) has expired, a FlexRay timer 1 interrupt is asserted. When the FlexRay timer 1 expires, FLXAnFRSIR.TI1 and FLXAnFROS.TI1S are set to '1'. A timer 1 interrupt then occurs while the FLXAnFROC.TI1E bit is effective.

Value after reset: 0002 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	T1MC[13:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	T1MS	T1RC	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	

Table 21.23 FLXAnFRT1C Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved These bits are always read as 0. When writing, always write 0.
29 to 16	T1MC[13:0]	Timer 1 Macrotick Count Bit Timer 1 macrotick count
15 to 2	—	Reserved These bits are always read as 0. When writing, always write 0.
1	T1MS	Timer 1 Mode Select Bit 0: Single-shot mode 1: Continuous mode
0	T1RC	Timer 1 Run Control Bit 0: Timer 1 halted 1: Timer 1 running

(1) FLXAnFRT1C.T1MC

Timer 1 Macrotick Count Bit

Before reconfiguration of the timer, the timer has to be halted first by writing '0' to bit FLXAnFRT1C.T1RC.

Valid values are 2 to 16383 MT in continuous mode

Valid values are 1 to 16383 MT in single-shot mode

When the configured macrotick count is reached, the FlexRay timer 1 interrupt is generated.

(2) FLXAnFRT1C.T1MS

Timer 1 Mode Select Bit

Before reconfiguration of the timer, the timer has to be halted first by writing '0' to bit FLXAnFRT1C.T1RC.

Configures the timer run mode. In Single-shot mode, the timer is deactivated when the timer configuration matches the configured cycle counter and macrotick value.

(3) FLXAnFRT1C.T1RC

Timer 1 Run Control Bit

Timer 1 can be activated (set FLXAnFRT1C.T1RC to '1') as long as the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state.

Timer 1 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

21.3.5.3 FLXAnFRT2C — FlexRay Timer 2 Configuration Register

This register is an absolute timer. It specifies the point in time when a FlexRay timer 2 interrupt occurs as the values of cycle count and macrotick (MT). When the FlexRay timer 2 expires, FLXAnFROS.T2IS are set to '1'. A timer 2 interrupt then occurs while the FLXAnFROC.T2IE bit is effective.

FlexRay timer 2 has the same absolute timer features as FlexRay timer 0.

CAUTION

The configuration of timer 2 is compared against the macrotick counter value, there is no separate counter for timer 2.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T2MO[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	T2CC[6:0]						—	—	—	—	—	—	—	T2MS	T2RC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 21.24 FLXAnFRT2C Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved These bits are always read as 0. When writing, always write 0.
29 to 16	T2MO[13:0]	Timer 2 Macrotick Offset Bit Timer 2 macrotick offset
15	—	Reserved This bit is always read as 0. When writing, always write 0.
14 to 8	T2CC[6:0]	Timer 2 Cycle Code Bit Timer 2 cycle code
7 to 2	—	Reserved These bits are always read as 0. When writing, always write 0.
1	T2MS	Timer 2 Mode Select Bit 0: Single-shot mode 1: Continuous mode
0	T2RC	Timer 2 Run Control Bit 0: Timer 2 halted 1: Timer 2 running

(1) FLXAnFRT2C.T2MO

Timer 2 Macrotick Offset Bit

Write 0 to the FLXAnFRT2C.T2RC bit and halt the timer before changing the value of the timer.

Configures the macrotick offset from the beginning of the communication cycle where the timer 2 interrupt is generated. The timer 2 interrupt is generated at this offset for each cycle of the cycle set.

(2) FLXAnFRT2C.T2CC

Timer 2 Cycle Code Bit

Before reconfiguration of the timer, the timer has to be halted first by writing 0 to bit FLXAnFRT2C.T2RC.

Configures the cycle set that generates timer 2 interrupt with the 7-bit timer 2 cycle code. For details, see **Section 21.4.8.2, Cycle Counter Filtering**.

(3) FLXAnFRT2C.T2MS

Timer 2 Mode Select Bit

Before reconfiguration of the timer, the timer has to be halted first by writing 0 to bit FLXAnFRT2C.T2RC.

Configures the timer run mode. In single-shot mode, the timer is deactivated when the timer configuration matches the configured cycle counter and macrotick value.

(4) FLXAnFRT2C.T2RC

Timer 2 Run Control Bit

Timer 2 can be operated when the POC is in NORMAL_ACTIVE or NORMAL_PASSIVE state. (set FLXAnFRT2C.T2RC to '1')

Timer 2 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

21.3.5.4 FLXAnFRSTPW1 — FlexRay Stop Watch Register 1

The stop watch is activated by the following trigger events.

- INTFLXAnLINE0 or INTFLXAnLINE1
- Writing bit FLXAnFRSTPW1.SSWT to '1'

With the macrotick counter increment following next to the stop watch activation, the actual cycle counter and macrotick values are captured in register FLXAnFRSTPW1 while the slot counter values for channels A and B are captured in register FLXAnFRSTPW2.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SMTV[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SCCV[5:0]					—	EINT1	EINT0	—	SSWT	EDGE	SWMS	ESWT	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W

Table 21.25 FLXAnFRSTPW1 Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	—	Reserved These bits are always read as 0. When writing, always write 0.
29 to 16	SMTV[13:0]	Stop Watch Captured Macrotick Value Bit Stop watch captured macrotick value
15, 14	—	Reserved These bits are always read as 0. When writing, always write 0.
13 to 8	SCCV[5:0]	Stop Watch Captured Cycle Counter Value Bit Stop watch captured cycle counter value
7	—	Reserved This bit is always read as 0. When writing, always write 0.
6	EINT1	Enable FlexRay Interrupt 1 Trigger Bit 0: Stop watch trigger by INTFLXAnLINE1 disabled 1: INTFLXAnLINE1 event triggers stop watch
5	EINT0	Enable FlexRay Interrupt 0 Trigger Bit 0: Stop watch trigger by INTFLXAnLINE0 disabled 1: INTFLXAnLINE0 event triggers stop watch

Table 21.25 FLXAnFRSTPW1 Register Contents (2/2)

Bit Position	Bit Name	Function
4	—	Reserved This bit is always read as 0. When writing, always write 0.
3	SSWT	Software Stop Watch Trigger Bit 0: Software trigger reset 1: Stop watch activated by software trigger
2	EDGE	Stop Watch Trigger Edge Select Bit 0: Falling edge 1: Rising edge
1	SWMS	Stop Watch Mode Select Bit 0: Single-shot mode 1: Continuous mode
0	ESWT	Enable Hardware Stop Watch Trigger Bit 0: Stop watch trigger disabled 1: Stop watch trigger enabled

(1) FLXAnFRSTPW1.SMTV

Stop Watch Captured Macrotick Value Bit

State of the macrotick counter when the stop watch event occurred.

(2) FLXAnFRSTPW1.SCCV

Stop Watch Captured Cycle Counter Value Bit

State of the cycle counter when the stop watch event occurred.

(3) FLXAnFRSTPW1.EINT1

Enable FlexRay Interrupt 1 Trigger Bit

Enables stop watch trigger by INTFLXAnLINE1 when FLXAnFRSTPW1.ESWT = '1'.

(4) FLXAnFRSTPW1.EINT0

Enable FlexRay Interrupt 0 Trigger Bit

Enables stop watch trigger by INTFLXAnLINE0 when FLXAnFRSTPW1.ESWT = '1'.

(5) FLXAnFRSTPW1.SSWT

Software Stop Watch Trigger Bit

Bits FLXAnFRSTPW1.ESWT and FLXAnFRSTPW1.SSWT cannot be set to '1' simultaneously. In this case, the write access to the register is ignored, and both bits keep their previous values. Either the external stop watch trigger or the software stop watch trigger may be used.

Writing 1 in this bit activates the stop watch. This bit is reset to '0' after the cycle count and slot count, and macrotick (MT) value are stored in the FlexRay stop watch register.

(6) FLXAnFRSTPW1.EDGE

Stop Watch Trigger Edge Select Bit

(7) FLXAnFRSTPW1.SWMS

Stop Watch Mode Select Bit

(8) FLXAnFRSTPW1.ESWT

Enable Stop Watch Trigger Bit

Bits FLXAnFRSTPW1.ESWT and FLXAnFRSTPW1.SSWT cannot be set to '1' simultaneously. In this case, the write access to the register is ignored, and both bits keep their previous values. Either the external stop watch trigger or the software stop watch trigger may be used.

If enabled, an INTFLXAnLINE0 event or an INTFLXAnLINE1 event activates the stop watch.

In single-shot mode, this bit is reset to '0' after the cycle count and slot count, and macrotick (MT) value are stored in the FlexRay stop watch register.

21.3.5.5 FLXAnFRSTPW2 — FlexRay Stop Watch Register 2

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SSCVB[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SSCVA[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.26 FLXAnFRSTPW2 Register Contents

Bit Position	Bit Name	Function
31 to 27	—	Reserved These bits are always read as 0. When writing, always write 0.
26 to 16	SSCVB[10:0]	Stop Watch Captured Slot Counter Value Channel B
15 to 11	—	Reserved These bits are always read as 0. When writing, always write 0.
10 to 0	SSCVA[10:0]	Stop Watch Captured Slot Counter Value Channel A

(1) FLXAnFRSTPW2.SSCVB

Stop Watch Captured Slot Counter Value Channel B

State of the slot counter for channel B when the stop watch event occurred.

(2) FLXAnFRSTPW2.SSCVA

Stop Watch Captured Slot Counter Value Channel A

State of the slot counter for channel A when the stop watch event occurred.

21.3.6 CC Control Registers

This section describes the registers provided by the CC to allow the host to control the operation of the CC. The FlexRay protocol specification requires the host to write application configuration data in CONFIG state only. Please note that the configuration registers are not locked for writing in DEFAULT_CONFIG state.

The configuration data is reset when DEFAULT_CONFIG state is entered from reset. To change POC state from DEFAULT_CONFIG to CONFIG state, the host has to apply CHI command CONFIG. If the host wants the CC to leave CONFIG state, the host has to execute the lock release sequence as described in **Section 21.3.3.1, FLXAnFRLCK — FlexRay Lock Register.**

21.3.6.1 FLXAnFRSUCC1 — FlexRay SUC Configuration Register 1

Value after reset: 0C40 1080_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	CCHB	CCHA	MTSB	MTSA	HCSE	TSM	WUCS	PTA[4:0]				
Value after reset	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSA[4:0]				—	TXSY	TXST	PBSY	—	—	—	CMD[3:0]				
Value after reset	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 21.27 FLXAnFRSUCC1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	—	Reserved These bits are always read as 0. When writing, always write 0.
27	CCHB	Connected to Channel B Bit Configures pChannels 0: Node not connected to channel B 1: Node connected to channel B (default after reset)
26	CCHA	Connected to Channel A Bit Configures pChannels 0: Node not connected to channel A 1: Node connected to channel A (default after reset)
25	MTSB	Select Channel B for MTS Transmission Bit 0: Channel B disabled for MTS transmission 1: Channel B selected for MTS transmission
24	MTSA	Select Channel A for MTS Transmission Bit 0: Channel A disabled for MTS transmission 1: Channel A selected for MTS transmission
23	HCSE	Halt due to Clock Sync Error Bit Configures pAllowHaltDueToClock 0: CC will enter or remain in NORMAL_PASSIVE 1: CC will enter HALT state

Table 21.27 FLXAnFRSUCC1 Register Contents (2/2)

Bit Position	Bit Name	Function
22	TSM	Transmission Slot Mode Bit Configures pSingleSlotEnabled 0: ALL Slot Mode 1: SINGLE Slot Mode (default after hard reset)
21	WUCS	Wakeup Channel Select Bit Configures pWakeupChannel 0: Send wakeup pattern on channel A 1: Send wakeup pattern on channel B
20 to 16	PTA[4:0]	Passive to Active Bit Configure pAllowPassiveToActive
15 to 11	CSA[4:0]	Cold Start Attempts Bit Configure gColdStartAttempts
10	—	Reserved This bit is always read as 0. When writing, always write 0.
9	TXSY	Transmit Sync Frame in Key Slot Bit Configure pKeySlotUsedForSync 0: No sync frame transmission in key slot, node is neither sync nor coldstart node 1: Key slot used to transmit sync frame, node is sync node
8	TXST	Transmit Startup Frame in Key Slot Bit Configure pKeySlotUsedForStartup 0: No startup frame transmission in key slot, node is non-coldstarter 1: Key slot used to transmit startup frame, node is leading or following coldstarter
7	PBSY	POC Busy Flag 0: POC not busy, FLXAnFRSUCC1.CMD writeable 1: POC is busy, FLXAnFRSUCC1.CMD locked
6 to 4	—	Reserved These bits are always read as 0. When writing, always write 0.
3 to 0	CMD[3:0]	CHI Command Vector Bit 0000: command_not_accepted 0001: CONFIG 0010: READY 0011: WAKEUP 0100: RUN 0101: ALL_SLOTS 0110: HALT 0111: FREEZE 1000: SEND_MTS 1001: ALLOW_COLDSTART 1010: RESET_STATUS_INDICATORS 1100: CLEAR_RAMs others: reserved

(1) FLXAnFRSUCC1.CCHB

Connected to Channel B Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Configures whether the node is connected to channel B (pChannels).

(2) FLXAnFRSUCC1.CCHA

Connected to Channel A Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Configures whether the node is connected to channel A (pChannels).

(3) FLXAnFRSUCC1.MTSB

Select Channel B for MTS Transmission Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

FLXAnFRSUCC1.MTSB may also be changed outside DEFAULT_CONFIG or CONFIG state when the write to FLXAnFRSUCC1 register is directly preceded by the unlock sequence for the Configuration Lock Key as described in **Section 21.3.3.1, FLXAnFRLCK — FlexRay Lock Register**. This may be combined with CHI command SEND_MTS. If both bits FLXAnFRSUCC1.MTSA and FLXAnFRSUCC1.MTSB are set to '1', an MTS symbol will be transmitted on both channels when requested by writing FLXAnFRSUCC1.CMD = "1000_B".

The bit selects channel B for MTS symbol transmission.

(4) FLXAnFRSUCC1.MTSA

Select Channel A for MTS Transmission Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

FLXAnFRSUCC1.MTSA may also be changed outside DEFAULT_CONFIG or CONFIG state when the write to FLXAnFRSUCC1 register is directly preceded by the unlock sequence for the Configuration Lock Key as described in **Section 21.3.3.1, FLXAnFRLCK — FlexRay Lock Register**. This may be combined with CHI command SEND_MTS. If both bits FLXAnFRSUCC1.MTSA and FLXAnFRSUCC1.MTSB are set to '1', an MTS symbol will be transmitted on both channels when requested by writing FLXAnFRSUCC1.CMD = "1000_B".

The bit selects channel A for MTS symbol transmission.

(5) FLXAnFRSUCC1.HCSE

Halt due to Clock Sync Error Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Controls the transition to HALT state due to a clock synchronization error (pAllowHaltDueToClock).

(6) FLXAnFRSUCC1.TSM

Transmission Slot Mode Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Selects the initial transmission slot mode (pSingleSlotEnabled).

In SINGLE slot mode, the CC may only transmit in the preconfigured key slot. The key slot ID is configured in the header section of message buffer 0 or that of message buffers 0 and 1 depending on bit FLXAnFRMRC.SPLM.

In case FLXAnFRSUCC1.TSM = '1', message buffer 0 or message buffers 0,1 can be (re)configured in DEFAULT_CONFIG or CONFIG state only. In ALL slot mode, the CC may transmit in all slots.

FLXAnFRSUCC1.TSM is a configuration bit that can only be set or reset by the host.

The CC changes to ALL slot mode when the host successfully applied the ALL_SLOTS command by writing FLXAnFRSUCC1.CMD = "0101_B" in POC state NORMAL_ACTIVE or NORMAL_PASSIVE. The actual slot mode is monitored by FLXAnFRCCSV.SLM.

(7) FLXAnFRSUCC1.WUCS

Wakeup Channel Select Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
With this bit, the host selects the channel on which the CC sends the Wakeup pattern (pWakeupChannel).

(8) FLXAnFRSUCC1.PTA

Passive to Active Condition Setting Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 0 to 31 even/odd cycle pairs.

Defines the number of consecutive even/odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state (pAllowPassiveToActive).
If set to “00000_B”, the CC is not allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state.

(9) FLXAnFRSUCC1.CSA

Cold Start Attempt Count Setting Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Must be identical in all nodes of a cluster.
Valid values are 2 to 31.

Configures the maximum number of attempts that a cold starting node is permitted to try to start up the network without receiving any valid response from another node (gColdStartAttempts).

(10) FLXAnFRSUCC1.TXSY

Transmit Sync Frame in Key Slot Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Defines whether the key slot is used to transmit sync frames (pKeySlotUsedForSync).

CAUTION

The protocol requires that both bits FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY be set for coldstart nodes.

(11) FLXAnFRSUCC1.TXST

Transmit Startup Frame in Key Slot Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Defines whether the key slot is used to transmit startup frames (pKeySlotUsedForStartup).

CAUTION

The protocol requires that both bits FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY be set for coldstart nodes.

(12) FLXAnFRSUCC1.PBSY

POC Busy Flag

Signals that the POC is busy and cannot accept a command from the host. FLXAnFRSUCC1.CMD is locked against write accesses.

Set to ‘1’ after reset during initialization of internal RAM blocks.

(13) FLXAnFRSUCC1.CMD**CHI Command Vector Bit**

The host may write any CHI command at any time, but certain commands are enabled only in certain POC states. If a command is not enabled, it will not be executed, the CHI command vector FLXAnFRSUCC1.CMD will be reset to “0000” = command_not_accepted, and flag FLXAnFREIR.CNA will be set to ‘1’.

In general the host must check FLXAnFRSUCC1.PBSY before writing a new CHI command.

In case the previous CHI command has not yet completed, FLXAnFREIR.CCL is set to ‘1’ together with FLXAnFREIR.CNA; the CHI command needs to be repeated.

Except for HALT state, a POC state change command applied while the CC is already in the requested POC state neither causes a state change nor will FLXAnFREIR.CNA be set.

Reading FLXAnFRSUCC1.CMD shows whether the last CHI command was accepted. The actual POC state is monitored by FLXAnFRCCSV.POCS.

Command_not_accepted

FLXAnFRSUCC1.CMD is reset to “0000_B” due to one of the following conditions:

- Illegal command applied by the host
- Host applied command to leave CONFIG state without preceding config lock key
- Host applied new command while execution of the previous host command has not completed
- Host writes command_not_accepted

When FLXAnFRSUCC1.CMD is reset to “0000_B”, FLXAnFREIR.CNA is set to ‘1’ an interrupt is generated when it is enabled. Commands that are not accepted are not executed.

CONFIG

Go to POC state CONFIG when called in POC state DEFAULT_CONFIG or READY. When called in HALT state, the CC transits to POC state DEFAULT_CONFIG. When called in any other state, FLXAnFRSUCC1.CMD will be reset to “0000_B” = command_not_accepted.

READY

Go to POC state READY when called in POC state CONFIG, NORMAL_ACTIVE, NORMAL_PASSIVE, STARTUP, or WAKEUP. When called in any other state, FLXAnFRSUCC1.CMD will be reset to “0000_B” = command_not_accepted.

WAKEUP

Go to POC state WAKEUP when called in POC state READY. When called in any other state, FLXAnFRSUCC1.CMD will be reset to “0000_B” = command_not_accepted.

RUN

Go to POC state STARTUP when called in POC state READY. When called in any other state, FLXAnFRSUCC1.CMD will be reset to “0000_B” = command_not_accepted.

ALL_SLOTS

Leave SINGLE slot mode and go to ALL-SLOTS mode after successful startup/integration at the next end of cycle when called in POC state NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, FLXAnFRSUCC1.CMD will be reset to “0000_B” = command_not_accepted.

HALT

Set halt request FLXAnFRCCSV.HRQ to '1' and go to POC state HALT at the next end of cycle when called in POC state NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, FLXAnFRSUCC1.CMD will be reset to "0000_B" = command_not_accepted.

FREEZE

Set the freeze status indicator FLXAnFRCCSV.FSI to '1' and go to POC state HALT immediately. Can be called from any state.

SEND_MTS

Send single MTS symbol during the next following symbol window on the channel configured by FLXAnFRSUCC1.MTSA, FLXAnFRSUCC1.MTSB, when called in POC state NORMAL_ACTIVE after CC entered ALL slot mode (FLXAnFRCCSV.SLM = "11"). When called in any other state, or when called while a previously requested MTS has not yet been transmitted, FLXAnFRSUCC1.CMD will be reset to "0000_B" = command_not_accepted.

ALLOW_COLDSTART

The command resets FLXAnFRCCSV.CSI to enable the node to become leading coldstarter. When called in state DEFAULT_CONFIG, CONFIG, or HALT, FLXAnFRSUCC1.CMD will be reset to "0000_B" = command_not_accepted. To become leading coldstarter, it is also required that both FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY be set.

RESET_STATUS_INDICATORS

Resets status flags FLXAnFRCCSV.CSNI, FLXAnFRCCSV.CSAI, and FLXAnFRCCSV.WSV, according to *IP sdm and E-Ray 1.0.3 specification v1.2.7* to their default values. Can be called in POC states READY and STARTUP. When called in any other state, FLXAnFRSUCC1.CMD will be reset to "0000_B" = command_not_accepted.

CLEAR_RAMs

Sets FLXAnFRMHDS.CRAM to '1' when called in DEFAULT_CONFIG or CONFIG state. When called in any other state, FLXAnFRSUCC1.CMD will be reset to "0000_B" = command_not_accepted.

FLXAnFRMHDS.CRAM is also set to '1' when the CC leaves reset. By setting FLXAnFRMHDS.CRAM, all internal RAM blocks are initialized to zero. During the initialization of the RAMs, FLXAnFRSUCC1.PBSY will show POC busy. Access to the configuration and status registers is possible during execution of CHI command CLEAR_RAMs. The initialization of the internal RAM block requires 2048 bus clock cycles. There should be no host access to input buffer or output buffer during initialization of the internal RAM blocks after reset or after assertion of CHI command CLEAR_RAMs.

Before asserting CHI command CLEAR_RAMs, confirm that no transfer between message RAM and input buffer or output buffer or the Transient Buffer RAMs is ongoing and that the data transfer handler is disabled (FLXAnFRITS.ITS = 0 and FLXAnFROTS.OTS = 0). This command also resets the message buffer status registers FLXAnFRMHDS, FLXAnFRLDTS, FLXAnFRFSR, FLXAnFRMHDF, FLXAnFRTXRQ1/2/3/4, FLXAnFRNDAT1/2/3/4, and FLXAnFRMBSC1/2/3/4.

CAUTIONS

1. All accepted commands with the exception of CLEAR_RAMs and SEND_MTS will cause a change of the POC state in the FlexRay domain after at most 8 cycles of the slower of the two clocks “bus clock” and “FlexRay sample clock”, assuming that POC was not busy when the command was applied and that no POC state change was forced by bus activity in that time frame. Reading register FLXAnFRCCSV will show data that is additionally delayed by synchronization from the FlexRay domain to the bus clock domain. The maximum additional delay is 12 cycles of the slower of the two clocks “bus clock” and “FlexRay sample clock”.
2. When a terminal restarts transfer as a leading coldstart node after transfer was stopped by using the FREEZE or READY command, the startup frame may not be transmitted in cycle 0. This depends on the internal state of the FlexRay module. This phenomenon occurs when the startup frame is set in a slot with a number from 1 to 7. This does not occur in ColdStart after a hardware reset. Even if this occurs, the second trial of ColdStart will succeed. ColdStart time becomes longer, but ColdStart will not be affected by the occurrence. To avoid this, allocate the Startup/Sync frame in static slot 8 or higher.

Table 21.28 below references the CHI commands from the FlexRay Protocol Specification to the FlexRay CHI command vector FLXAnFRSUCC1.CMD.

Table 21.28 Reference to CHI Host Command Summary from FlexRay Protocol Specification

CHI Command	Where Processed (POC States)	CHI Command Vector CMD
ALL_SLOTS	POC: normal active, POC: normal passive	ALL_SLOTS
ALLOW_COLDSTART	All except POC: default config, POC: config, POC: halt	ALLOW_COLDSTART
CONFIG	POC: default config, POC: ready	CONFIG
CONFIG_COMPLETE	POC: config	Unlock sequence & READY
DEFAULT_CONFIG	POC: halt	CONFIG
FREEZE	All	FREEZE
HALT	POC: normal active, POC: normal passive	HALT
READY	All except POC: default config, POC: config, POC: ready, POC: halt	READY
RUN	POC: ready	RUN
WAKEUP	POC: ready	WAKEUP

21.3.6.2 FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2

Value after reset: 0100 0504_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	LTN[3:0]				—	—	—	LT[20:16]				
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LT[15:0]															
Value after reset	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.29 FLXAnFRSUCC2 Register Contents

Bit Position	Bit Name	Function
31 to 28	—	Reserved These bits are always read as 0. When writing, always write 0.
27 to 24	LTN[3:0]	Listen Timeout Noise Bit Configure (gListenNoise - 1)
23 to 21	—	Reserved These bits are always read as 0. When writing, always write 0.
20 to 0	LT[20:0]	Listen Timeout Bit Configure pdListenTimeout

(1) FLXAnFRSUCC2.LTN

Listen Timeout Noise Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 16.

FLXAnFRSUCC2.LTN must be configured identically in all nodes of a cluster.

Configures the upper limit for startup and wakeup listen timeout in the presence of noise expressed as a multiple of pdListenTimeout.

CAUTION

The wakeup/startup noise timeout is calculated as follows:

$$\text{pdListenTimeout} \times \text{gListenNoise} = \text{FLXAnFRSUCC2.LT} \times (\text{FLXAnFRSUCC2.LTN} + 1).$$

(2) FLXAnFRSUCC2.LT

Listen Timeout Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 1284 to 1283846 μ T.

Configures wakeup/startup listen timeout in μ T.

21.3.6.3 FLXAnFRSUCC3 — FlexRay SUC Configuration Register 3

Value after reset: 0000 0011_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	WCF[3:0]			WCP[3:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.30 FLXAnFRSUCC3 Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are always read as 0. When writing, always write 0.
7 to 4	WCF[3:0]	Maximum Without Clock Correction Fatal Bit (condition for transition to HALT state) Configure gMaxWithoutClockCorrectionFatal
3 to 0	WCP[3:0]	Maximum Without Clock Correction Passive Bit (condition for transition to NORMAL_PASSIVE state) Configure gMaxWithoutClockCorrectionPassive

(1) FLXAnFRSUCC3.WCF

Maximum Without Clock Correction Fatal Bit (condition for transition to HALT state)

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 15 cycle pairs.

Setting must be identical in all nodes of a cluster.

Defines the number of consecutive even/odd cycle pairs with missing clock correction terms that will cause a transition from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state.

CAUTION

The transition to HALT state is prevented if FLXAnFRSUCC1.HCSE is not set.

(2) FLXAnFRSUCC3.WCP

Maximum Without Clock Correction Passive Bit (transition to NORMAL_PASSIVE state)

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 15 cycle pairs.

Setting must be identical in all nodes of a cluster.

Defines the number of consecutive even/odd cycle pairs with missing clock correction terms that will cause a transition from NORMAL_ACTIVE to NORMAL_PASSIVE state.

21.3.6.4 FLXAnFRNEMC — FlexRay NEM Configuration Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	NML[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 21.31 FLXAnFRNEMC Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved These bits are always read as 0. When writing, always write 0.
3 to 0	NML[3:0]	Network Management Vector Length Bit Configure gNetworkManagementVectorLength

(1) FLXAnFRNEMC.NML

Network Management Vector Length Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 12 bytes.

The configured length must be identical in all nodes of a cluster.

These bits configure the length of the NM vector.

21.3.6.5 FLXAnFRPRTC1 — FlexRay PRT Configuration Register 1

Value after reset: 084C 0633_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RWP[5:0]						—	RXW[8:0]								
Value after reset	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BRP[1:0]		SPP[1:0]		—	CASM[6:0]						TSST[3:0]				
Value after reset	0	0	0	0	0	1	1	0	0	0	1	1	0	0	1	1
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.32 FLXAnFRPRTC1 Register Contents

Bit Position	Bit Name	Function
31 to 26	RWP[5:0]	Repetitions of Tx Wakeup Pattern Bit Configure pWakeupPattern
25	—	Reserved This bit is always read as 0. When writing, always write 0.
24 to 16	RXW[8:0]	Wakeup Symbol Receive Window Length Bit Configure gdWakeupSymbolRxWindow
15, 14	BRP[1:0]	Baud Rate Prescaler Bit Configure gdSampleClockPeriod and pSamplesPerMicrotick 00: 10 Mbps 01: 5 Mbps 10: 2.5 Mbps 11: 2.5 Mbps
13, 12	SPP[1:0]	Strobe Point Position Bit Configure Strobe point position 00: Sample 5 01: Sample 4 10: Sample 6 11: Sample 5
11	—	Reserved This bit is always read as 0. When writing, always write 0.
10 to 4	CASM[6:0]	Collision Avoidance Symbol Max Bit Configure gdCASRxLowMax
3 to 0	TSST[3:0]	Transmission Start Sequence Transmitter Bit Configure gdTSSTransmitter

(1) FLXAnFRPRTC1.RWP

Repetitions of Tx Wakeup Pattern Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 63.

Configures the number of repetitions (sequences) of the Tx wakeup symbol.

(2) FLXAnFRPRTC1.RXW

Wakeup Symbol Receive Window Length Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 76 to 301 bit times.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to test the duration of the received wakeup pattern.

(3) FLXAnFRPRTC1.BRP

Baud Rate Prescaler Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

The Baud Rate Prescaler configures the baud rate on the FlexRay bus. The baud rates listed below are valid with a sample clock set to 80 MHz. One bit time always consists of 8 samples independent of the configured baud rate.

00 = 10 MBit/s

$gdSampleClockPeriod = 12.5 \text{ ns} = 1 * \text{'sample clock'}$

$pSamplesPerMicrotick = 2 (1 \mu\text{T} = 25 \text{ ns})$

01 = 5 MBit/s

$gdSampleClockPeriod = 25 \text{ ns} = 2 * \text{'sample clock'}$

$pSamplesPerMicrotick = 1 (1 \mu\text{T} = 25 \text{ ns})$

10, 11 = 2.5 MBit/s

$gdSampleClockPeriod = 50 \text{ ns} = 4 * \text{'sample clock'}$

$pSamplesPerMicrotick = 1 (1 \mu\text{T} = 50 \text{ ns})$

(4) FLXAnFRPRTC1.SPP

Strobe Point Position Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Defines the sample count value for strobing. The strobed bit value is set to the voted value when the sample count is incremented to the value configured by FLXAnFRPRTC1.SPP.

CAUTION

The current revision 2.1 of the FlexRay protocol requires that FLXAnFRPRTC1.SPP = "00_B". The alternate strobe point positions could be used to compensate for asymmetries in the physical layer.

(5) FLXAnFRPRTC1.CASM

Collision Avoidance Symbol Max Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

CASM6 is fixed to '1'.

Valid values are 67 to 99 bit times.

Configures the upper limit of the acceptance window for a collision avoidance symbol (CAS).

(6) FLXAnFRPRTC1.TSST

Transmission Start Sequence Transmitter Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 3 to 15 bit times.

Must be identical in all nodes of a cluster.

Configures the duration of the Transmission Start Sequence (TSS) in terms of bit times (1 bit time = $4 \mu\text{T} = 100 \text{ ns}$ at 10 Mbps).

21.3.6.6 FLXAnFRPRTC2 — FlexRay PRT Configuration Register 2

Value after reset: 0F2D 0A0E_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	TXL[5:0]						TXI[7:0]							
Value after reset	0	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RXL[5:0]						—	—	RXI[5:0]					
Value after reset	0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.33 FLXAnFRPRTC2 Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved These bits are always read as 0. When writing, always write 0.
29 to 24	TXL[5:0]	Wakeup Symbol Transmit Low Bit Configure gdWakeupSymbolTxLow
23 to 16	TXI[7:0]	Wakeup Symbol Transmit Idle Bit Configure gdWakeupSymbolTxIdle
15, 14	—	Reserved These bits are always read as 0. When writing, always write 0.
13 to 8	RXL[5:0]	Wakeup Symbol Receive Low Bit Configure gdWakeupSymbolRxLow
7, 6	—	Reserved These bits are always read as 0. When writing, always write 0.
5 to 0	RXI[5:0]	Wakeup Symbol Rx Idle Bit Configure gdWakeupSymbolRxIdle

(1) FLXAnFRPRTC2.TXL

Wakeup Symbol Transmit Low Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 15 to 60 bit times.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to transmit the low phase of the wakeup symbol.

(2) FLXAnFRPRTC2.TXI

Wakeup Symbol Transmit Idle Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 45 to 180 bit times.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to transmit the idle phase of the wakeup symbol.

(3) FLXAnFRPRTC2.RXL

Wakeup Symbol Receive Low Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 10 to 55 bit times.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to test the duration of the low phase of the received wakeup symbol.

(4) FLXAnFRPRTC2.RXI

Wakeup Symbol Rx Idle Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 14 to 59 bit times.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to test the duration of the idle phase of the received wakeup symbol.

21.3.6.7 FLXAnFRMHDC — FlexRay MHD Configuration Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	SLT[12:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SFDL[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.34 FLXAnFRMHDC Register Contents

Bit Position	Bit Name	Function
31 to 29	—	Reserved These bits are always read as 0. When writing, always write 0.
28 to 16	SLT[12:0]	Start of Latest Transmit Bit Configure pLatestTx
15 to 7	—	Reserved These bits are always read as 0. When writing, always write 0.
6 to 0	SFDL[6:0]	Static Frame Data Length Bit Configure gPayloadLengthStatic

(1) FLXAnFRMHDC.SLT

Start of Latest Transmit Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7981 minislots.

Configures the maximum minislot value allowed before inhibiting frame transmission in the dynamic segment of the cycle. There is no transmission in dynamic segment if FLXAnFRMHDC.SLT is set to zero.

(2) FLXAnFRMHDC.SFDL

Static Frame Data Length Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 127.

The payload length must be identical in all nodes of a cluster.

Configures the cluster-wide payload length for all frames sent in the static segment in double bytes.

21.3.6.8 FLXAnFRGTUC1 — FlexRay GTU Configuration Register 1

Value after reset: 0000 0280_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	UT[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UT[15:0]															
Value after reset	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.35 FLXAnFRGTUC1 Register Contents

Bit Position	Bit Name	Function
31 to 20	—	Reserved These bits are always read as 0. When writing, always write 0.
19 to 0	UT[19:0]	Microtick per Cycle Bit Configure pMicroPerCycle

(1) FLXAnFRGTUC1.UT

Microtick per Cycle Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 640 to 640000 μ T.

Configures the duration of the communication cycle in microticks.

21.3.6.9 FLXAnFRGTUC2 — FlexRay GTU Configuration Register 2

Value after reset: 0002 000A_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	—	—	—	—	—	—	—	—	—	SNM[3:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	MPC[13:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.36 FLXAnFRGTUC2 Register Contents

Bit Position	Bit Name	Function
31 to 20	—	Reserved These bits are always read as 0. When writing, always write 0.
19 to 16	SNM[3:0]	Sync Node Max
15, 14	—	Reserved These bits are always read as 0. When writing, always write 0.
13 to 0	MPC[13:0]	Macrotick Per Cycle Bit Configure gMacroPerCycle

(1) FLXAnFRGTUC2.SNM

Sync Node Max Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 15.

Must be identical in all nodes of a cluster.

Maximum number of frames within a cluster with sync frame indicator bit SYN set to '1'.

(2) FLXAnFRGTUC2.MPC

Macrotick Per Cycle Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 10 to 16000 MT.

The cycle length must be identical in all nodes of a cluster.

Configures the duration of one communication cycle in macroticks.

21.3.6.10 FLXAnFRGTUC3 — FlexRay GTU Configuration Register 3

Value after reset: 0202 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	MIOB[6:0]						—	MIOA[6:0]							
Value after reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UIOB[7:0]							UIOA[7:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.37 FLXAnFRGTUC3 Register Contents

Bit Position	Bit Name	Function
31	—	Reserved This bit is always read as 0. When writing, always write 0.
30 to 24	MIOB[6:0]	Macrotick Initial Offset Channel B Bit Configure pMacroInitialOffset[B]
23	—	Reserved This bit is always read as 0. When writing, always write 0.
22 to 16	MIOA[6:0]	Macrotick Initial Offset Channel A Bit Configure pMacroInitialOffset[A]
15 to 8	UIOB[7:0]	Microtick Initial Offset Channel B Bit Configure pMicroInitialOffset[B]
7 to 0	UIOA[7:0]	Microtick Initial Offset Channel A Bit Configure pMicroInitialOffset[A]

(1) FLXAnFRGTUC3.MIOB

Macrotick Initial Offset Channel B Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 72 MT.

Must be identical in all nodes of a cluster

Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration

(2) FLXAnFRGTUC3.MIOA

Macrotick Initial Offset Channel A Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 72 MT.

Must be identical in all nodes of a cluster

Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration

(3) FLXAnFRGTUC3.UIOB

Microtick Initial Offset Channel B Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 240 μ T.

Configures the number of microticks between the actual time reference point on channel B and the subsequent macrotick boundary of the secondary time reference point. The parameter depends on pDelayCompensation [B] and therefore has to be set for each channel independently.

(4) FLXAnFRGTUC3.UIOA

Microtick Initial Offset Channel A Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 240 μ T.

Configures the number of microticks between the actual time reference point on channel A and the subsequent macrotick boundary of the secondary time reference point. The parameter depends on pDelayCompensation [A] and therefore has to be set for each channel independently.

21.3.6.11 FLXAnFRGTUC4 — FlexRay GTU Configuration Register 4

For details about configuration of FLXAnFRGTUC4.NIT and FLXAnFRGTUC4.OCS, see **Section 21.4.2(5), Configuration of NIT Start and Offset Correction Start.**

Value after reset: 0008 0007_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	OCS[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	NIT[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.38 FLXAnFRGTUC4 Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved These bits are always read as 0. When writing, always write 0.
29 to 16	OCS[13:0]	Offset Correction Start Bit Configure (gOffsetCorrectionStart - 1)
15, 14	—	Reserved These bits are always read as 0. When writing, always write 0.
13 to 0	NIT[13:0]	Network Idle Time Start Bit Configure (gMacroPerCycle -gdNIT - 1)

(1) FLXAnFRGTUC4.OCS

Offset Correction Start Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 8 to 15998 MT.

For cluster consisting of E-Ray implementations only, it is sufficient to program FLXAnFRGTUC4.OCS = FLXAnFRGTUC4.NIT + 1.

Must be identical in all nodes of a cluster.

Determines the start of the offset correction within the NIT phase, calculated from start of cycle.

(2) FLXAnFRGTUC4.NIT

Network Idle Time Start Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 7 to 15997 MT.

Must be identical in all nodes of a cluster.

Configures the starting point of the Network Idle Time NIT at the end of the communication cycle expressed in terms of macroticks from the beginning of the cycle. The start of NIT is recognized if MacroTick = gMacroPerCycle -gdNIT -1 and the increment pulse of MacroTick is set.

21.3.6.12 FLXAnFRGTUC5 — FlexRay GTU Configuration Register 5

Value after reset: 0E00 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEC[7:0]							—	—	—	CDD[4:0]					
Value after reset	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCB[7:0]							DCA[7:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.39 FLXAnFRGTUC5 Register Contents

Bit Position	Bit Name	Function
31 to 24	DEC[7:0]	Decoding Correction Bit Configure pDecodingCorrection
23 to 21	—	Reserved These bits are always read as 0. When writing, always write 0.
20 to 16	CDD[4:0]	Cluster Drift Damping Bit Configure pClusterDriftDamping
15 to 8	DCB[7:0]	Delay Compensation Channel B Bit Configure pDelayCompensation[B]
7 to 0	DCA[7:0]	Delay Compensation Channel A Bit Configure pDelayCompensation[A]

(1) FLXAnFRGTUC5.DEC

Decoding Correction Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 14 to 143 μ T.

Configures the decoding correction value in microticks used to determine the primary time reference point.

(2) FLXAnFRGTUC5.CDD

Cluster Drift Damping Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 20 μ T.

Configures the cluster drift damping value in microticks used in clock synchronization to minimize accumulation of rounding errors.

(3) FLXAnFRGTUC5.DCB

Delay Compensation Channel B Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 200 μ T.

Used to compensate for reception delays on channel B. This covers assumed propagation delay up to

cPropagationDelayMax for microticks in the range of 0.0125 to 0.05 μ s. In practice, the minimum of the propagation delays of all sync nodes should be applied.

(4) FLXAnFRGTUC5.DCA

Delay Compensation Channel A Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 200 μ T.

Used to compensate for reception delays on channel A. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 to 0.05 μ s. In practice, the minimum of the propagation delays of all sync nodes should be applied.

21.3.6.13 FLXAnFRGTUC6 — FlexRay GTU Configuration Register 6

Value after reset: 0002 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	MOD[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ASR[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.40 FLXAnFRGTUC6 Register Contents

Bit Position	Bit Name	Function
31 to 27	—	Reserved These bits are always read as 0. When writing, always write 0.
26 to 16	MOD[10:0]	Maximum Oscillator Drift Bit Configure pdMaxDrift
15 to 11	—	Reserved These bits are always read as 0. When writing, always write 0.
10 to 0	ASR[10:0]	Accepted Startup Range Bit Configure pdAcceptedStartupRange

(1) FLXAnFRGTUC6.MOD

Maximum Oscillator Drift Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 1923 μ T.

Maximum drift offset between two nodes that operate with unsynchronized clocks over one communication cycle in μ T.

(2) FLXAnFRGTUC6.ASR

Accepted Startup Range Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 1875 μ T.

Number of microticks constituting the expanded range of measured deviation for startup frames during integration.

21.3.6.14 FLXAnFRGTUC7 — FlexRay GTU Configuration Register 7

Value after reset: 0002 0004_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	NSS[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SSL[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.41 FLXAnFRGTUC7 Register Contents

Bit Position	Bit Name	Function
31 to 26	—	Reserved These bits are always read as 0. When writing, always write 0.
25 to 16	NSS[9:0]	Number of Static Slots Bit Configure gNumberOfStaticSlots
15 to 10	—	Reserved These bits are always read as 0. When writing, always write 0.
9 to 0	SSL[9:0]	Static Slot Length Bit Configure gdStaticSlot

(1) FLXAnFRGTUC7.NSS

Number of Static Slots Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 1023.

The number of static slots must be identical in all nodes of a cluster.

Configures the number of static slots in a cycle.

(2) FLXAnFRGTUC7.SSL

Static Slot Length Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 4 to 659 MT.

The static slot length must be identical in all nodes of a cluster.

Configures the length of a static slot in macroticks.

21.3.6.15 FLXAnFRGTUC8 — FlexRay GTU Configuration Register 8

Value after reset: 0000 0002_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	NMS[12:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	MSL[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.42 FLXAnFRGTUC8 Register Contents

Bit Position	Bit Name	Function
31 to 29	—	Reserved These bits are always read as 0. When writing, always write 0.
28 to 16	NMS[12:0]	Number of Minislots Bit Configure gNumberOfMinislots
15 to 6	—	Reserved These bits are always read as 0. When writing, always write 0.
5 to 0	MSL[5:0]	Minislot Length Bit Configure gdMinislot

(1) FLXAnFRGTUC8.NMS

Number of Minislots Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7986.

The number of minislots must be identical in all nodes of a cluster.

Configures the number of minislots within the dynamic segment of a cycle.

(2) FLXAnFRGTUC8.MSL

Minislot Length Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 63 MT.

The minislot length must be identical in all nodes of a cluster.

Configures the length of a minislot in macroticks.

21.3.6.16 FLXAnFRGTUC9 — FlexRay GTU Configuration Register 9

Value after reset: 0000 0101_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSI[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	MAPO[4:0]				—	—	APO[5:0]						
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.43 FLXAnFRGTUC9 Register Contents

Bit Position	Bit Name	Function
31 to 18	—	Reserved These bits are always read as 0. When writing, always write 0.
17, 16	DSI[1:0]	Dynamic Slot Idle Phase Bit Configure gdDynamicSlotIdlePhase
15 to 13	—	Reserved These bits are always read as 0. When writing, always write 0.
12 to 8	MAPO[4:0]	Minislot Action Point Offset Bit Configure gdMinislotActionPointOffset
7, 6	—	Reserved These bits are always read as 0. When writing, always write 0.
5 to 0	APO[5:0]	Action Point Offset Bit Configure gdActionPointOffset

(1) FLXAnFRGTUC9.DSI

Dynamic Slot Idle Phase Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 2 Minislots.

Must be identical in all nodes of a cluster.

Configures the duration of the dynamic slot idle phase in the number of minislots. The duration has to be greater than or equal to the idle detection time.

(2) FLXAnFRGTUC9.MAPO

Minislot Action Point Offset Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 31 MT.

Must be identical in all nodes of a cluster.

Configures the action point offset in macroticks within the minislots of the dynamic segment.

(3) FLXAnFRGTUC9.APO

Action Point Offset Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 63 MT.

Must be identical in all nodes of a cluster.

Configures the action point offset in macroticks within static slots and symbol window.

21.3.6.17 FLXAnFRGTUC10 — FlexRay GTU Configuration Register 10

Value after reset: 0002 0005_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	MRC[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MOC[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.44 FLXAnFRGTUC10 Register Contents

Bit Position	Bit Name	Function
31 to 27	—	Reserved These bits are always read as 0. When writing, always write 0.
26 to 16	MRC[10:0]	Maximum Rate Correction Bit Configure pRateCorrectionOut
15, 14	—	Reserved These bits are always read as 0. When writing, always write 0.
13 to 0	MOC[13:0]	Maximum Offset Correction Bit Configure pOffsetCorrectionOut

(1) FLXAnFRGTUC10.MRC

Maximum Rate Correction Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 1923 μ T.

Holds the maximum permitted rate correction value to be applied by the internal clock synchronization algorithm. The CC checks only the internal rate correction value against the maximum rate correction value (absolute value).

(2) FLXAnFRGTUC10.MOC

Maximum Offset Correction Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 5 to 15266 μ T.

Holds the maximum permitted offset correction value (absolute value) to be applied by the internal clock synchronization algorithm (absolute value). The CC checks only the internal offset correction value against the maximum offset correction value.

21.3.6.18 FLXAnFRGTUC11 — FlexRay GTU Configuration Register 11

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ERC[2:0]			—	—	—	—	—	EOC[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ERCC[1:0]		—	—	—	—	—	—	EOCC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 21.45 FLXAnFRGTUC11 Register Contents

Bit Position	Bit Name	Function
31 to 27	—	Reserved These bits are always read as 0. When writing, always write 0.
26 to 24	ERC[2:0]	External Rate Correction Bit Configure pExternRateCorrection
23 to 19	—	Reserved These bits are always read as 0. When writing, always write 0.
18 to 16	EOC[2:0]	External Offset Correction Bit Configure pExternOffsetCorrection
15 to 10	—	Reserved These bits are always read as 0. When writing, always write 0.
9, 8	ERCC[1:0]	External Rate Correction Control Bit Configure vExternRateControl 00: External rate correction is prohibited. 01: External rate correction is prohibited. 10: Subtract 11: Add
7 to 2	—	Reserved These bits are always read as 0. When writing, always write 0.
1, 0	EOCC[1:0]	External Offset Correction Control Bit Configure vExternOffsetControl 00: External offset correction is prohibited. 01: External offset correction is prohibited. 10: Subtract 11: Add

(1) FLXAnFRGTUC11.ERC

External Rate Correction Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7 μ T.

Holds the external rate correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted or added from or to the calculated rate correction value. The value is applied during NIT.

(2) FLXAnFRGTUC11.EOC

External Offset Correction Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7 μ T.

Holds the external offset correction value in microticks to be applied by the internal clock synchronization algorithm.

The value is subtracted from or added to the calculated offset correction value. The value is applied during NIT.

(3) FLXAnFRGTUC11.ERCC

External Rate Correction Control Bit

Should be modified only outside NIT (Network Idle Time).

By writing to FLXAnFRGTUC11.ERCC the external rate correction is enabled as specified below.

00_B = External rate correction is prohibited.

01_B = External rate correction is prohibited.

10_B = Subtract

External rate correction value subtracted from calculated rate correction value

11_B = Add

External rate correction value added to calculated rate correction value

(4) FLXAnFRGTUC11.EOCC

External Offset Correction Control Bit

Should be modified only outside NIT (Network Idle Time).

By writing to FLXAnFRGTUC11.EOCC the external offset correction is enabled as specified below.

00_B = External offset correction is prohibited.

01_B = External offset correction is prohibited.

10_B = Subtract

External offset correction value subtracted from calculated offset correction value

11_B = Add

External offset correction value added to calculated offset correction value

21.3.7 CC Status Registers

During 8/16-bit accesses to status variables coded with more than 8/16-bit, the variable might be updated by the CC between two accesses (non-atomic read accesses).

21.3.7.1 FLXAnFRCCSV — FlexRay CC Status Vector Register

Value after reset: 0010 4000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	PSL[5:0]					RCA[4:0]				WSV[2:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	CSI	CSAI	CSNI	—	—	SLM[1:0]		HRQ	FSI	POCS[5:0]					
Value after reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.46 FLXAnFRCCSV Register Contents (1/2)

Bit	Bit Name	Function
31, 30	—	Reserved These bits are always read as 0. When writing, always write 0.
29 to 24	PSL[5:0]	POC Status Log Flags Status of FLXAnFRCCSV.POCS immediately before entering HALT state.
23 to 19	RCA[4:0]	Remaining Coldstart Attempts Flags Indicate vRemainingColdstartAttempts
18 to 16	WSV[2:0]	Wakeup Status Flags Indicate vPOC!WakeupStatus
15	—	Reserved This bit is always read as 0. When writing, always write 0.
14	CSI	Cold Start Inhibit Flag Indicates vColdStartInhibit 0: Cold starting of node enabled 1: Cold starting of node disabled
13	CSAI	Coldstart Abort Indicator Flag
12	CSNI	Coldstart Noise Indicator Flag Indicates vPOC!ColdstartNoise
11, 10	—	Reserved These bits are always read as 0. When writing, always write 0.
9, 8	SLM[1:0]	Slot Mode Flags Indicate vPOC!SlotMode 00: SINGLE 01: reserved 10: ALL_PENDING 11: ALL

Table 21.46 FLXAnFRCCSV Register Contents (2/2)

Bit	Bit Name	Function
7	HRQ	Halt Request Flag Indicates vPOC!CHI!HaltRequest
6	FSI	Freeze Status Indicator Flag Indicates vPOC!Freeze
5 to 0	POCS[5:0]	Protocol Operation Control Status Flags

(1) FLXAnFRCCSV.PSL

POC Status Log Flags

Set the value of FLXAnFRCCSV.POCS immediately before entering HALT state.

Set to HALT when FREEZE command is applied during HALT state and FLXAnFRCCSV.FSI is not already set, i.e. the HALT state was not reached by FREEZE command.

Reset to “000000_B” when leaving HALT state.

(2) FLXAnFRCCSV.RCA

Remaining Coldstart Attempts Flags

Indicates the number of remaining coldstart attempts (vRemainingColdstartAttempts).

The initial value of FLXAnFRCCSV.RCA during CONFIG and DEFAULT_CONFIG state is also FLXAnFRSUCC1.CSA.

The RUN command resets this counter to the maximum number of coldstart attempts as configured by FLXAnFRSUCC1.CSA.

(3) FLXAnFRCCSV.WSV

Wakeup Status Flags

Indicates the status of the current wakeup attempt (vPOC!WakeupStatus).

Reset to “0” when entering Wakeup state, by CHI command RESET_STATUS_INDICATORS, or by transition from DEFAULT_CONFIG to CONFIG state

000_B = UNDEFINED

Wakeup not yet executed by the CC.

001_B = RECEIVED_HEADER

Set when the CC finishes wakeup due to the reception of a frame header without coding violation on either channel in WAKEUP_LISTEN state.

010_B = RECEIVED_WUP

Set when the CC finishes wakeup due to the reception of a valid wakeup pattern on the configured wakeup channel in WAKEUP_LISTEN state.

011_B = COLLISION_HEADER

Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid header on either channel.

100_B = COLLISION_WUP

Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid wakeup pattern on the configured wakeup channel.

101_B = COLLISION_UNKNOWN

Set when the CC stops wakeup by leaving WAKEUP_DETECT state after expiration of the wakeup timer without receiving a valid wakeup pattern or a valid frame header.

110_B = TRANSMITTED

Set when the CC has successfully completed the transmission of the wakeup pattern.

111_B = reserved

(4) FLXAnFRCCSV.CSI

Cold Start Inhibit Flag

Indicates that the node is disabled from cold starting (vColdStartInhibit).

The flag is set to “1” whenever the POC enters READY state due to CHI command READY.

The flag has to be reset under control of the host by CHI command ALLOW_COLDSTART (FLXAnFRSUCC1.CMD = “1001_B”).

(5) FLXAnFRCCSV.CSAI

Coldstart Abort Indicator Flag

Indicates that a coldstart attempt was aborted.

Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state.

(6) FLXAnFRCCSV.CSNI

Coldstart Noise Indicator Flag

Indicates that the cold start procedure occurred under noisy conditions (vPOC!ColdstartNoise).

Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state.

(7) FLXAnFRCCSV.SLM

Slot Mode Flags

Indicates the actual slot mode of the POC (vPOC!SlotMode) in states READY, WAKEUP, STARTUP, NORMAL_ACTIVE, and NORMAL_PASSIVE.

Default value is SINGLE. Changes to ALL, depending on FLXAnFRSUCC1.TSM.

In NORMAL_ACTIVE or NORMAL_PASSIVE state the CHI command ALL_SLOTS will change the slot mode from SINGLE over ALL_PENDING to ALL.

Set FLXAnFRSUCC1.TSM to SINGLE except for NORMAL_ACTIVE or NORMAL_PASSIVE.

(8) FLXAnFRCCSV.HRQ

Halt Request Flag

Indicates that a request from the host has been received to halt the POC at the end of the communication cycle (vPOC!CHIHaltRequest).

Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

(9) FLXAnFRCCSV.FSI

Freeze Status Indicator Flag

Indicates that the POC has entered the HALT state due to CHI command FREEZE or due to an error condition requiring an immediate POC halt (vPOC!Freeze).

Reset by transition from HALT to DEFAULT_CONFIG state.

(10) FLXAnFRCCSV.POCS

Protocol Operation Control Status Flag

Indicates the actual state of operation of the CC Protocol Operation Control

00 0000_B = DEFAULT_CONFIG state

00 0001_B = READY state

00 0010_B = NORMAL_ACTIVE state

00 0011_B = NORMAL_PASSIVE state

00 0100_B = HALT state

00 0110_B ... 00 1110_B = reserved

00 1111_B = CONFIG state

Indicates the actual state of operation of the POC in the wakeup path

01 0000_B = WAKEUP_STANDBY state

01 0001_B = WAKEUP_LISTEN state

01 0010_B = WAKEUP_SEND state

01 0011_B = WAKEUP_DETECT state

Indicates the actual state of operation of the POC in the startup path

10 0000_B = STARTUP_PREPARE state

10 0001_B = COLDSTART_LISTEN state

10 0010_B = COLDSTART_COLLISION_RESOLUTION state

10 0011_B = COLDSTART_CONSISTENCY_CHECK state

10 0100_B = COLDSTART_GAP state

10 0101_B = COLDSTART_JOIN State

10 0110_B = INTEGRATION_COLDSTART_CHECK state

10 0111_B = INTEGRATION_LISTEN state

10 1000_B = INTEGRATION_CONSISTENCY_CHECK state

10 1001_B = INITIALIZE_SCHEDULE state

10 1010_B = ABORT_STARTUP state

10 1011_B = STARTUP_SUCCESS state

Other than above = reserved

21.3.7.2 FLXAnFRCCEV — FlexRay CC Error Vector Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PTAC[4:0]				ERRM[1:0]		—	—	CCFC[3:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.47 FLXAnFRCCEV Register Contents

Bit Position	Bit Name	Function
31 to 13	—	Reserved These bits are always read as 0. When writing, always write 0.
12 to 8	PTAC[4:0]	Passive to Active Count Flag Indicate vAllowPassiveToActive
7, 6	ERRM[1:0]	Error Mode Flags Indicate vPOC!ErrorMode 00: ACTIVE 01: PASSIVE 10: COMM_HALT 11: reserved
5, 4	—	Reserved These bits are always read as 0. When writing, always write 0.
3 to 0	CCFC[3:0]	Clock Correction Failed Counter Indicate vClockCorrectionFailed

(1) FLXAnFRCCEV.PTAC

Passive to Active Count Flag

Indicates the number of consecutive even/odd cycle pairs that have passed with valid rate and offset correction terms, while the node is waiting to transit from NORMAL_PASSIVE state to NORMAL_ACTIVE state. The transition takes place when FLXAnFRCCEV.PTAC equals FLXAnFRSUCC1.PTA -1.

Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

(2) FLXAnFRCCEV.ERRM

Error Mode Flags

Indicates the actual error mode of the POC (vPOC!ErrorMode).

Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

(3) FLXAnFRCCEV.CCFC

Clock Correction Failed Counter

Indicates the clock correction failed counter of the POC (vClockCorrectionFailed).

The Clock Correction Failed Counter is incremented by one at the end of any odd communication cycle where either the

missing offset correction error or missing rate correction error are active.

The Clock Correction Failed Counter is reset to '0' at the end of an odd communication cycle if neither the offset correction failed nor the rate correction failed errors are active.

The Clock Correction Failed Counter stops at 15.

Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

21.3.7.3 FLXAnFRSCV — FlexRay Slot Counter Value Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SCCB[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SCCA[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.48 FLXAnFRSCV Register Contents

Bit Position	Bit Name	Function
31 to 27	—	Reserved These bits are always read as 0. When writing, always write 0.
26 to 16	SCCB[10:0]	Slot Counter Channel B Indicates vSlotCounter[B]
15 to 11	—	Reserved These bits are always read as 0. When writing, always write 0.
10 to 0	SCCA[10:0]	Slot Counter Channel A Indicates vSlotCounter[A]

(1) FLXAnFRSCV.SCCB

Slot Counter Channel B

Current slot counter value on channel B (vSlotCounter[B]). The value is incremented by the CC and reset at the start of a communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRSCV.SCCA

Slot Counter Channel A

Current slot counter value on channel A (vSlotCounter[A]). The value is incremented by the CC and reset at the start of a communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

21.3.7.4 FLXAnFRMTCCV — FlexRay Macrotick and Cycle Counter Value Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	CCV[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MTV[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.49 FLXAnFRMTCCV Register Contents

Bit Position	Bit Name	Function
31 to 22	—	Reserved These bits are always read as 0. When writing, always write 0.
21 to 16	CCV[5:0]	Cycle Counter Value Indicates vCycleCounter
15, 14	—	Reserved These bits are always read as 0. When writing, always write 0.
13 to 0	MTV[13:0]	Macrotick Value Indicates vMacrotick

(1) FLXAnFRMTCCV.CCV

Cycle Counter Value

Current cycle counter value (vCycleCounter). The value is incremented by the CC at the start of a communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRMTCCV.MTV

Macrotick Value

Current macrotick value (vMacrotick). The value is incremented by the CC and reset at the start of a communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

21.3.7.5 FLXAnFRRCV — FlexRay Rate Correction Value Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RCV[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.50 FLXAnFRRCV Register Contents

Bit Position	Bit Name	Function
31 to 12	—	Reserved These bits are always read as 0. When writing, always write 0.
11 to 0	RCV[11:0]	Rate Correction Value Flags Indicates vRateCorrection

(1) FLXAnFRRCV.RCV

Rate Correction Value Flags

Indicates internal rate correction value (vRateCorrection/ two's complement) before limitation. If the FLXAnFRRCV.RCV value exceeds the limits defined by FLXAnFRGTUC10.MRC, flag FLXAnFRSFS.RCLR is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

CAUTION

The external rate correction value is added to the limited rate correction value.

21.3.7.6 FLXAnFROCV — FlexRay Offset Correction Value Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	OCV[18:16]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.51 FLXAnFROCV Register Contents

Bit Position	Bit Name	Function
31 to 19	—	Reserved These bits are always read as 0. When writing, always write 0.
18 to 0	OCV[18:0]	Offset Correction Value Flags Indicate vOffsetCorrection

(1) FLXAnFROCV.OCV

Offset Correction Value Flags

Indicate offset correction value (vOffsetCorrection/ two's complement) before limitation. If the FLXAnFROCV.OCV value exceeds the limits defined by FLXAnFRGTUC10.MOC, flag FLXAnFRSFS.OCLR is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

CAUTION

The external offset correction value is added to the limited offset correction value.

21.3.7.7 FLXAnFRSFS — FlexRay Sync Frame Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RCLR	MRCS	OCLR	MOCS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VSBO[3:0]				VSBE[3:0]				VSAO[3:0]				VSAE[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.52 FLXAnFRSFS Register Contents

Bit Position	Bit Name	Function
31 to 20	—	Reserved These bits are always read as 0. When writing, always write 0.
19	RCLR	Rate Correction Limit Reached Flag 0: Rate correction below limit 1: Rate correction limit reached
18	MRCS	Missing Rate Correction Signal Flag 0: Rate correction signal valid 1: Missing rate correction signal
17	OCLR	Offset Correction Limit Reached Flag 0: Offset correction below limit 1: Offset correction limit reached
16	MOCS	Missing Offset Correction Signal Flag 0: Offset correction signal valid 1: Missing offset correction signal
15 to 12	VSBO[3:0]	Valid Sync Frames Channel B, odd communication cycle
11 to 8	VSBE[3:0]	Valid Sync Frames Channel B, even communication cycle
7 to 4	VSAO[3:0]	Valid Sync Frames Channel A, odd communication cycle
3 to 0	VSAE[3:0]	Valid Sync Frames Channel A, even communication cycle

(1) FLXAnFRSFS.RCLR

Rate Correction Limit Reached Flag

The rate correction limit reached flag signals to the host that the rate correction value has exceeded its limit as defined by FLXAnFRGTUC10.MRC10 - MRC0. The flag is updated by the CC at start of offset correction phase.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(2) FLXAnFRSFS.MRCS

Missing Rate Correction Signal Flag

The missing rate correction flag signals to the host that no rate correction calculation can be performed because no pairs of even/odd sync frames were received. The flag is updated by the CC at start of offset correction phase.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(3) FLXAnFRSFS.OCLR

Offset Correction Limit Reached Flag

The offset correction limit reached flag signals to the host that the offset correction value has exceeded its limit as defined by FLXAnFRGTUC10.MOC. The flag is updated by the CC at start of offset correction phase.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(4) FLXAnFRSFS.MOCS

Missing Offset Correction Signal Flag

The missing offset correction flag signals to the host that no offset correction calculation can be performed because no sync frames were received. The flag is updated by the CC at start of offset correction phase.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(5) FLXAnFRSFS.VSBO

Valid Sync Frames Channel B, odd communication cycle

These bits are only valid when FLXAnFRSUCC1.CCHB is '1'.

Holds the number of valid sync frames received on channel B in the odd communication cycle. If transmission of sync frames is enabled by FLXAnFRSUCC1.TXSY, the value is incremented by one. The value is updated during the NIT of each odd communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(6) FLXAnFRSFS.VSBE

Valid Sync Frames Channel B, even communication cycle

These bits are only valid when FLXAnFRSUCC1.CCHB is '1'.

Holds the number of valid sync frames received on channel B in the even communication cycle. If transmission of sync frames is enabled by FLXAnFRSUCC1.TXSY, the value is incremented by one. The value is updated during the NIT of each even communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(7) FLXAnFRSFS.VSAO

Valid Sync Frames Channel A, odd communication cycle

These bits are only valid when FLXAnFRSUCC1.CCHA is '1'.

Holds the number of valid sync frames received on channel A in the odd communication cycle. If transmission of sync frames is enabled by FLXAnFRSUCC1.TXSY, the value is incremented by one. The value is updated during the NIT of each odd communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(8) FLXAnFRSFS.VSAE

Valid Sync Frames Channel A, even communication cycle

These bits are only valid when FLXAnFRSUCC1.CCHA is '1'.

Holds the number of valid sync frames received on channel A in the even communication cycle. If transmission of sync frames is enabled by FLXAnFRSUCC1.TXSY, the value is incremented by one. The value is updated during the NIT of each even communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

21.3.7.8 FLXAnFRSWNIT — FlexRay Symbol Window and NIT Status Register

Symbol window related status information is updated by the CC at the end of the symbol window for each channel. NIT related status information is updated by the CC at the end of the NIT for each channel.

During startup the status data is not updated.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SBNB	SENB	SBNA	SENA	MTSB	MTSA	TCSB	SBSB	SESB	TCSA	SBSA	SESA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.53 FLXAnFRSWNIT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 12	—	Reserved These bits are always read as 0. When writing, always write 0.
11	SBNB	Slot Boundary Violation during NIT Channel B Flag 0: No slot boundary violation detected 1: Slot boundary violation during NIT detected on channel B
10	SENB	Syntax Error during NIT Channel B Flag 0: No syntax error detected 1: Syntax error during NIT detected on channel B
9	SBNA	Slot Boundary Violation during NIT Channel A Flag 0: No slot boundary violation detected 1: Slot boundary violation during NIT detected on channel A
8	SENA	Syntax Error during NIT Channel A Flag 0: No syntax error detected 1: Syntax error during NIT detected on channel A
7	MTSB	MTS Received on Channel B Flag 0: No MTS symbol received on channel B 1: MTS symbol received on channel B
6	MTSA	MTS Received on Channel A Flag 0: No MTS symbol received on channel A 1: MTS symbol received on channel A
5	TCSB	Transmission Conflict in Symbol Window Channel B Flag 0: No transmission conflict detected 1: Transmission conflict in symbol window detected on channel B
4	SBSB	Slot Boundary Violation in Symbol Window Channel B Flag 0: No slot boundary violation detected 1: Slot boundary violation during symbol window detected on channel B

Table 21.53 FLXAnFRSWNIT Register Contents (2/2)

Bit Position	Bit Name	Function
3	SESB	Syntax Error in Symbol Window Channel B Flag 0: No syntax error detected 1: Syntax error during symbol window detected on channel B
2	TCSA	Transmission Conflict in Symbol Window Channel A Flag 0: No transmission conflict detected 1: Transmission conflict in symbol window detected on channel A
1	SBSA	Slot Boundary Violation in Symbol Window Channel A Flag 0: No slot boundary violation detected 1: Slot boundary violation during symbol window detected on channel A
0	SESA	Syntax Error in Symbol Window Channel A Flag 0: No syntax error detected 1: Syntax error during symbol window detected on channel A

(1) FLXAnFRSWNIT.SBNB

Indicates a Slot Boundary Violation during NIT Channel B Flag (vSS!BViolationB).

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRSWNIT.SENB

Indicates a Syntax Error during NIT Channel B Flag (vSS!SyntaxErrorB).

Reset when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFRSWNIT.SBNA

Indicates a Slot Boundary Violation during NIT Channel A Flag (vSS!BViolationA).

Reset when leaving CONFIG state or when entering STARTUP state.

(4) FLXAnFRSWNIT.SENA

Indicates a Syntax Error during NIT Channel A Flag (vSS!SyntaxErrorA).

Reset when leaving CONFIG state or when entering STARTUP state.

(5) FLXAnFRSWNIT.MTSB

Indicates a MTS Received on Channel B Flag (vSS!ValidMTSB).

Media Access Test symbol received on channel B during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window.

When this bit is set to '1', also interrupt flag FLXAnFRSIR.MTSB is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

(6) FLXAnFRSWNIT.MTSA

Indicates a MTS Received on Channel A Flag (vSS!ValidMTSA).

Media Access Test symbol received on channel A during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window.

When this bit is set to '1', also interrupt flag FLXAnFRSIR.MTSA is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

(7) FLXAnFRSWNIT.TCSB

Indicates a Transmission Conflict in Symbol Window Channel B Flag (vSS!TxConflictB).
Reset when leaving CONFIG state or when entering STARTUP state.

(8) FLXAnFRSWNIT.SBSB

Indicates a Slot Boundary Violation in Symbol Window Channel B Flag (vSS!BViolationB).
Reset when leaving CONFIG state or when entering STARTUP state.

(9) FLXAnFRSWNIT.SESB

Indicates a Syntax Error in Symbol Window Channel B Flag (vSS!SyntaxErrorB).
Reset when leaving CONFIG state or when entering STARTUP state.

(10) FLXAnFRSWNIT.TCSA

Indicates a Transmission Conflict in Symbol Window Channel A Flag (vSS!TxConflictA).
Reset when leaving CONFIG state or when entering STARTUP state.

(11) FLXAnFRSWNIT.SBSA

Indicates a Slot Boundary Violation in Symbol Window Channel A Flag (vSS!BViolationA).
Reset when leaving CONFIG state or when entering STARTUP state.

(12) FLXAnFRSWNIT.SESA

Indicates a Syntax Error in Symbol Window Channel A Flag (vSS!SyntaxErrorA).
Reset when leaving CONFIG state or when entering STARTUP state.

21.3.7.9 FLXAnFRACS — FlexRay Aggregated Channel Status Register

Do not rewrite this register using bit manipulation instructions.

The aggregated channel status provides the host with an accrued status of channel activity for all communication slots regardless of whether they are assigned for transmission or subscribed for reception.

The aggregated channel status also includes status data from the symbol window and the network idle time.

The status data is updated (set) after each slot and aggregated until it is reset by the host.

During startup, the status data is not updated.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SBVB	CIB	CEDB	SEDB	VFRB	—	—	—	SBVA	CIA	CEDA	SEDA	VFRA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 21.54 FLXAnFRACS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 13	—	Reserved These bits are always read as 0. When writing, always write 0.
12	SBVB	Slot Boundary Violation on Channel B Flag 0: No slot boundary violation observed 1: Slot boundary violation(s) observed on channel B
11	CIB	Communication Indicator Channel B Flag 0: No valid frame(s) received in slots containing any additional communication 1: Valid frame(s) received on channel B in slots containing any additional communication
10	CEDB	Content Error Detected on Channel B Flag 0: No frame with content error received 1: Frame(s) with content error received on channel B
9	SEDB	Syntax Error Detected on Channel B Flag 0: No syntax error observed 1: Syntax error(s) observed on channel B
8	VFRB	Valid Frame Received on Channel B Flag 0: No valid frame received 1: Valid frame(s) received on channel B
7 to 5	—	Reserved These bits are always read as 0. When writing, always write 0.
4	SBVA	Slot Boundary Violation on Channel A Flag 0: No slot boundary violation observed 1: Slot boundary violation(s) observed on channel A

Table 21.54 FLXAnFRACS Register Contents (2/2)

Bit Position	Bit Name	Function
3	CIA	Communication Indicator Channel A Flag 0: No valid frame(s) received in slots containing any additional communication 1: Valid frame(s) received on channel A in slots containing any additional communication
2	CEDA	Content Error Detected on Channel A Flag 0: No frame with content error received 1: Frame(s) with content error received on channel A
1	SEDA	Syntax Error Detected on Channel A Flag 0: No syntax error observed 1: Syntax error(s) observed on channel A
0	VFRA	Valid Frame Received on Channel A Flag 0: No valid frame received 1: Valid frame(s) received on channel A

(1) FLXAnFRACS.SBVB

Slot Boundary Violation on Channel B Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

One or more slot boundary violations were observed on channel B at any time during the observation period (static or dynamic slots, symbol window, and NIT).

When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.EDB is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRACS.CIB

Communication Indicator Channel B Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

One or more valid frames were received on channel B in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error OR content error OR slot boundary violation.

When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.EDB is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

The set condition of the flag FLXAnFRACS.CIB is also fulfilled if there is only a single frame in the slot and the slot boundary at the end of the slot is reached during the frames channel idle recognition phase.

(3) FLXAnFRACS.CEDB

Content Error Detected on Channel B Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

One or more frames with a content error were received on channel B in any static or dynamic slot during the observation period.

When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.EDB is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

(4) FLXAnFRACS.SEDB

Syntax Error Detected on Channel B Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

One or more syntax errors in static or dynamic slots, symbol window, and NIT were observed on channel B.

When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.EDB is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

(5) FLXAnFRACS.VFRB

Valid Frame Received on Channel B Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

One or more valid frames were received on channel B in any static or dynamic slot during the observation period.

Reset when leaving CONFIG state or when entering STARTUP state.

(6) FLXAnFRACS.SBVA

Slot Boundary Violation on Channel A Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

Slot boundary violations were observed on channel A at any time during the observation period (static or dynamic slots, symbol window, and NIT).

When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.EDA is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

(7) FLXAnFRACS.CIA

Communication Indicator Channel A Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

One or more valid frames were received on channel A in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame, and had one of syntax error, content error, or slot boundary violation.

When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.EDA is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

The set condition of the flag FLXAnFRACS.CIA is also fulfilled if there is only a single frame in the slot and the slot boundary at the end of the slot is reached during the frames channel idle recognition phase.

(8) FLXAnFRACS.CEDA

Content Error Detected on Channel A Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

One or more frames with a content error were received on channel A in any static or dynamic slot during the observation period.

When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.EDA is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

(9) FLXAnFRACS.SEDA

Syntax Error Detected on Channel A Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

One or more syntax errors in static or dynamic slots, symbol window, and NIT were observed on channel A.

When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.EDA is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

(10) FLXAnFRACS.VFRA

Valid Frame Received on Channel A Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

One or more valid frames were received on channel A in any static or dynamic slot during the observation period.

Reset when leaving CONFIG state or when entering STARTUP state.

21.3.7.10 FLXAnFRESIDm — FlexRay Even Sync ID Register m (m = 1 to 15)

Registers FLXAnFRESID1 to FLXAnFRESID15 hold the frame IDs of the sync frames received in even communication cycles used for clock synchronization up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register FLXAnFRESID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an even communication cycle, register FLXAnFRESID1 holds the respective sync frame ID as configured in message buffer 0 and flags FLXAnFRESID1.RXEA, FLXAnFRESID1.RXEB are set. The value is updated during the NIT of each even communication cycle.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXEB	RXEA	—	—	—	—	EID[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.55 FLXAnFRESIDm (m = 1 to 15) Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved These bits are always read as 0. When writing, always write 0.
15	RXEB	Received/Configured Even Sync ID on Channel B Flag 0: No sync frame received on channel B or node not configured to transmit sync frames 1: Sync frame received on channel B or node configured to transmit sync frames
14	RXEA	Received/Configured Even Sync ID on Channel A Flag 0: No sync frame received on channel A or node not configured to transmit sync frames 1: Sync frame received on channel A or node configured to transmit sync frames
13 to 10	—	Reserved These bits are always read as 0. When writing, always write 0.
9 to 0	EID[9:0]	Even Sync ID Flags (vsSyncIDListA,B even)

(1) FLXAnFRESIDm.RXEB

Received/Configured Even Sync ID on Channel B Flag

Signals that a sync frame corresponding to the stored even sync ID was received on channel B or that the node is configured to be a sync node with key slot indicated by the FLXAnFRESID1.EID.

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRESIDm.RXEA

Received/Configured Even Sync ID on Channel A Flag

Signals that a sync frame corresponding to the stored even sync ID was received on channel A or that the node is configured to be a sync node with key slot indicated by FLXAnFRESID1.EID.

Reset when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFRESIDm.EID

Even Sync ID Flags (vsSyncIDListA, B even)

Sync frame ID even communication cycle

Reset when leaving CONFIG state or when entering STARTUP state.

21.3.7.11 FLXAnFROSIDm — FlexRay Odd Sync ID Register m (m = 1 to 15)

Registers FLXAnFROSID1 to FLXAnFROSID15 hold the frame IDs of the sync frames received in odd communication cycles used for clock synchronization up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register FLXAnFROSID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an odd communication cycle, register FLXAnFROSID1 holds the respective sync frame ID as configured in message buffer 0 and flags FLXAnFROSID1.RXOA and FLXAnFROSID1.RXOB are set. The value is updated during the NIT of each odd communication cycle.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXOB	RXOA	—	—	—	—	OID[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.56 FLXAnFROSIDm Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved These bits are always read as 0. When writing, always write 0.
15	RXOB	Received/Configured Odd Sync ID on Channel B Flag 0: No sync frame received on channel B or node not configured to transmit sync frames 1: Sync frame received on channel B or node configured to transmit sync frames
14	RXOA	Received/Configured Odd Sync ID on Channel A Flag 0: No sync frame received on channel A or node not configured to transmit sync frames 1: Sync frame received on channel A or node configured to transmit sync frames
13 to 10	—	Reserved These bits are always read as 0. When writing, always write 0.
9 to 0	OID[9:0]	Odd Sync ID Flags (vsSyncIDListA,B odd)

(1) FLXAnFROSIDm.RXOB

Received/Configured Odd Sync ID on Channel B Flag

Signals that a sync frame corresponding to the stored odd sync ID was received on channel B or that the node is configured to be a sync node with key slot indicated by the FLXAnFROSID1.OID.

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFROSIDm.RXOA

Received/Configured Odd Sync ID on Channel A Flag

Signals that a sync frame corresponding to the stored odd sync ID was received on channel A or that the node is configured to be a sync node with key slot indicated by the FLXAnFRISID1.OID.

Reset when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFROSIDm.OID

Odd Sync ID Flags (vsSyncIDListA,B odd)

Sync frame ID odd communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state

21.3.7.12 FLXAnFRNMVm — FlexRay Network Management Vector Register m (m = 1 to 3)

The three network management registers hold the accrued NM vector (see **Section 21.4.7, Network Management**).

The CC updates the NM vector at the end of each communication cycle as long as the CC is either in NORMAL_ACTIVE or NORMAL_PASSIVE state.

NMVn-bytes exceeding the configured NM vector length are not valid.

For information about the byte alignment of the received NM vector in this register (see **Section 21.4.17, Byte Alignment**).

Value after reset: 0000 0000 _H																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NM[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.57 FLXAnFRNMVm Register Contents

Bit Position	Bit Name	Function
31 to 0	NM[31:0]	<p>NM Vector the three network management vector registers hold the accrued NM vector (configurable 0 to 12 bytes). The NM vector to be held is generated by bit-wise logic OR for each NM vector received on each channel (valid static frames with PPI = '1') (see Section 21.4.7, Network Management).</p> <p>For information about the byte alignment of the received NM vector in this register, (see Section 21.4.17, Byte Alignment).</p> <p>NMVn-bytes exceeding the configured NM vector length are not valid.</p> <p>The register contents are updated at the end of each communication cycle as long as the CC is either in NORMAL_ACTIVE or NORMAL_PASSIVE state.</p> <p>These bits are cleared when leaving CONFIG state or when entering STARTUP state.</p>

21.3.8 Message Buffer Control Registers

21.3.8.1 FLXAnFRMRC — FlexRay Message RAM Configuration Register

The message RAM configuration register defines the number of message buffers assigned to the static segment, dynamic segment, and FIFO.

The message RAM can be divided into up three different areas: static buffer area, static and dynamic buffer area, FIFO area. If present, the static buffer area starts at message buffer 0.

The start of the static and dynamic buffer area is configured by FLXAnFRMRC.FDB. FLXAnFRMRC.FDB defines the end of the static buffer area. If no static buffer area is present, the static and dynamic buffer area starts at message buffer 0.

The start of the FIFO area is configured by FLXAnFRMRC.FFB. FLXAnFRMRC.FFB defines the end of the previous area, which can be either the static buffer area or the static and dynamic buffer area.

If no static buffer area and no static and dynamic buffer area is present, the FIFO area starts at message buffer 0.

With FLXAnFRMRC.LCB, the end of the last configured area is configured. This can be the static buffer area, the static and dynamic buffer area, or the FIFO area.

Figure 21.2 shows an example configuration of the message RAM where all the areas are configured.

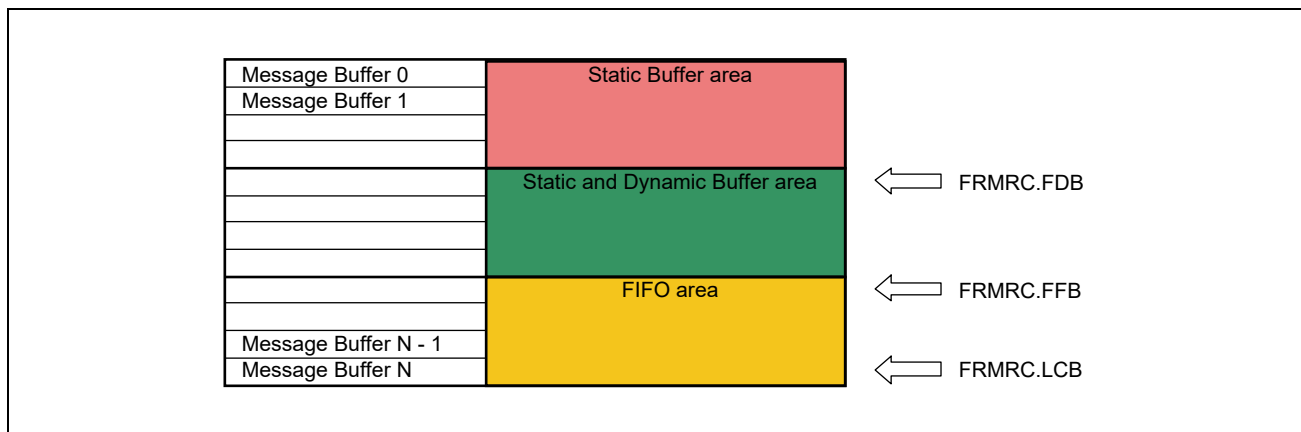


Figure 21.2 Message RAM Organization

CAUTIONS

1. If the node is configured as a sync node (FLXAnFRSUCC1.TXSY = '1') or for single slot mode operation (FLXAnFRSUCC1.TSM = '1'), message buffers 0 and 1 are reserved for sync frames or single slot frames and have to be configured with the node-specific key slot ID. If the node is neither configured as a sync node nor for single slot operation, message buffers 0 and 1 are treated like all other message buffers.
 2. The maximum number of header sections is 128. This means a maximum of 128 message buffers can be configured. The maximum length of a data section is 254 bytes. The length of the data section can be configured separately for each message buffer. For details, see **Section 21.4.13, Message RAM**.
 3. If two or more message buffers are assigned to slot 1 by use of cycle filtering, all of them must be located either in the "static buffers" or at the beginning of the "static + dynamic buffers" section.
 4. The FlexRay protocol specification requires that each node send a frame in its key slot. Therefore, at least message buffer 0 is reserved for transmission in the key slot. Due to this requirement, a maximum number of 127 message buffers can be assigned to the FIFO. Nevertheless, a non-protocol-conforming configuration without a transmission slot in the static segment would still be operational.
 5. The payload length and the length of the data section need to be configured identically for all message buffers belonging to the FIFO via FLXAnFRWRHS2.PLC and FLXAnFRWRHS3.DP. When the CC is not in DEFAULT_CONFIG or CONFIG state, reconfiguration of message buffers belonging to the FIFO is locked.
-

Value after reset: 0180 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SPLM	SEC[1:0]		LCB[7:0]							
Value after reset	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FFB[7:0]							FDB[7:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.58 FLXAnFRMRC Register Contents

Bit Position	Bit Name	Function
31 to 27	—	Reserved These bits are always read as 0. When writing, always write 0.
26	SPLM	Sync Frame Payload Multiplex 0: Writing to message buffer 0 is prohibited. 1: Writing to message buffer 0 or 1 is prohibited.
25, 24	SEC[1:0]	Secure Buffers 00: Writing is enabled for all buffers. 01: Writing is prohibited to static buffer and FIFO, limited transmission 10: Writing is prohibited for all buffers. 11: Writing is prohibited for all buffers, limited transmission
23 to 16	LCB[7:0]	Last Configured Buffer 0 to 127: Number of message buffers is FLXAnFRMRF.LCB + 1 128: No message buffer configured
15 to 8	FFB[7:0]	First Buffer of FIFO 0: All message buffers assigned to the FIFO 1 to 127: Message buffers from FLXAnFRMRC.FFB to FLXAnFRMRC.LCB assigned to the FIFO 128: No message buffer configured
7 to 0	FDB[7:0]	First Dynamic Buffer 0: No group of message buffers exclusively for the static segment configured 1 to 127: Message buffers 0 to FLXAnFRMRC.FDB - 1 reserved for static segment 128: No dynamic message buffers configured

(1) FLXAnFRMRC.SPLM

Sync Frame Payload Multiplex

The user can only write to this bit when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

This bit is only enabled if the node is configured as a sync node (FLXAnFRSUCC1.TXSY = '1') or for single slot mode operation (FLXAnFRSUCC1.TSM = '1').

When this bit is set to '1', message buffers 0 and 1 are dedicated for sync frame transmission with separate payload data on channels A and B.

When this bit is set to '0', sync frames are transmitted from message buffer 0 with the same payload data on all channels configured. Note that the channel filter configuration for message buffer 0 and message buffer 1 are selected according to this SPLM bit setting.

(2) FLXAnFRMRC.SEC

Secure Buffer

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

These bits are disabled when the CC is in DEFAULT_CONFIG or CONFIG state.

For temporary unlocking, see **Section 21.4.13.4, Host Handling of Access Errors**.

00_B = Writing is enabled for all buffers.

Reconfiguration of message buffers enabled with numbers < FLXAnFRMRC.FFB enabled

Exception: In nodes configured for sync frame transmission or for single slot mode operation, writing to message buffer 0 (and if FLXAnFRMRC.SPLM = '1', also message buffer 1) is prohibited.

01_B = static buffers locked, FIFO locked, limited transmission

Reconfiguration of message buffers with numbers < FLXAnFRMRC.FDB and with numbers ≥ FLXAnFRMRC.FFB locked and transmission of message buffers for static segment with numbers ≥ FLXAnFRMRC.FDB disabled

10_B = all buffers locked

Writing to any message buffer is prohibited.

11_B = Writing to any buffers is prohibited.

Reconfiguration of all message buffer locked and transmission of message buffers for static segment with numbers ≥ FLXAnFRMRC.FDB disabled

(3) FLXAnFRMRC.LCB

Last Buffer Setting

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

When a Static and Dynamic Buffer area is configured (FLXAnFRMRC.FDB < 128), the user should configure FLXAnFRMRC.LCB ≥ FLXAnFRMRC.FDB.

When a FIFO area is configured (FLXAnFRMRC.FFB < 128), the user should configure FLXAnFRMRC.LCB ≥ FLXAnFRMRC.FFB.

(4) FLXAnFRMRC.FFB

FIFO First Buffer Setting

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

When a Static and Dynamic Buffer area is configured (FLXAnFRMRC.FDB < 128), the user should configure FLXAnFRMRC.FFB > FLXAnFRMRC.FDB.

(5) FLXAnFRMRC.FDB

First Dynamic Buffer Setting

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

21.3.8.2 FLXAnFRFRF — FlexRay FIFO Rejection Filter Register

The FIFO Rejection Filter defines the user specified sequence of bits to which channel, frame ID, and cycle count of the incoming frames are compared. Together with the FIFO Rejection Filter Mask Register, this register determines whether a message is rejected by the FIFO.

Value after reset: 0180 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	RNF	RSS	CYF[6:0]						
Value after reset	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FID[10:0]										CH[1:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.59 FLXAnFRFRF Register Contents

Bit Position	Bit Name	Function
31 to 25	—	Reserved These bits are always read as 0. When writing, always write 0.
24	RNF	Reject Null Frames 0: Null frames are stored in the FIFO 1: Reject all null frames
23	RSS	Reject in Static Segment 0: FIFO also used for static segment 1: Reject messages in static segment
22 to 16	CYF[6:0]	Cycle Counter Filter
15 to 13	—	Reserved These bits are always read as 0. When writing, always write 0.
12 to 2	FID[10:0]	Frame ID Filter 0 to 2047: Frame ID filter values
1, 0	CH[1:0]	Channel Filter 00: receive on both channels 01: receive only on channel B 10: receive only on channel A 11: Reception prohibited

(1) FLXAnFRFRF.RNF

Reject Null Frame

The user can only write to this bit when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
If this bit is set to '1', received null frames are not stored in the FIFO.

(2) FLXAnFRFRF.RSS

Reject in Static Segment

The user can only write to this bit when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
If this bit is set to '1', the FIFO is used only for the dynamic segment.

(3) FLXAnFRFRF.CYF

Cycle Counter Filter

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles that are not specified by FLXAnFRFRF.CYF, all frames are rejected. For details about the configuration of the cycle counter filter, see **Section 21.4.8.2, Cycle Counter Filtering**.

(4) FLXAnFRFRF.FID

Frame ID Filter

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Determines the frame ID to be rejected by the FIFO. With the additional configuration of register FLXAnFRFRFM, the corresponding frame ID filter bits are ignored. When FLXAnFRFRFM.MFID is zero, a frame ID filter value of zero means that there is no frame ID to be rejected.

(5) FLXAnFRFRF.CH

Channel Filter

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

If reception on both channels is configured, also in the static segment, both frames (from channels A and B) are always stored in the FIFO, even if they are identical.

21.3.8.3 FLXAnFRFRFM — FlexRay FIFO Rejection Filter Mask Register

The FlexRay FIFO Rejection Filter Mask Register specifies the frame ID filter bits relevant to rejection filtering. If a bit is set to '1', it indicates that the corresponding bit in the FLXAnFRFRF register will not be considered for rejection filtering.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	MFID[10:0]										—	—	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 21.60 FLXAnFRFRFM Register Contents

Bit Position	Bit Name	Function
31 to 13	—	Reserved These bits are always read as 0. When writing, always write 0.
12 to 2	MFID[10:0]	Mask Frame ID Filter 0: Corresponding frame ID filter bit is used for rejection filtering 1: Ignore corresponding frame ID filter bit.
1, 0	—	Reserved These bits are always read as 0. When writing, always write 0.

(1) FLXAnFRFRFM.MFID

Mask Frame ID Filter

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

21.3.8.4 FLXAnFRFCL — FlexRay FIFO Critical Level Register

Value after reset: 0000 0080_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CL[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.61 FLXAnFRFCL Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are always read as 0. When writing, always write 0.
7 to 0	CL[7:0]	Critical Level Setting Critical level setting

(1) FLXAnFRFCL.CL

Critical Level Setting

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

When the receive FIFO fill level FLXAnFRFSR.RFFL is equal to or greater than the critical level configured by FLXAnFRFCL.CL, the receive FIFO critical level flag FLXAnFRFSR.RFCL is set to 1.

If FLXAnFRFCL.CL is programmed to values > 128, bit FLXAnFRFSR.RFCL is never set to 1.

21.3.9 Message Buffer Status Registers

21.3.9.1 FLXAnFRMHDS — FlexRay Message Handler Status Register

Do not rewrite this register using bit manipulation instructions.

Value after reset: 0000 0080_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	MBU[6:0]						—	MBT[6:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	FMB[6:0]						CRAM	MFMB	FMBD	ATBF2	ATBF1	AMR	—	—	
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R

Table 21.62 FLXAnFRMHDS Register Contents

Bit Position	Bit Name	Function
31	—	Reserved This bit is always read as 0. When writing, always write 0.
30 to 24	MBU[6:0]	Message Buffer Updated Flags
23	—	Reserved This bit is always read as 0. When writing, always write 0.
22 to 16	MBT[6:0]	Message Buffer Transmitted Flags
15	—	Reserved This bit is always read as 0. When writing, always write 0.
14 to 8	FMB[6:0]	Faulty Message Buffer Number Flags
7	CRAM	Internal RAM Clear Flag 0: No execution of the CHI command CLEAR_RAMs 1: Execution of the CHI command CLEAR_RAMs ongoing
6	MFMB	Multiple Faulty Message Buffer Detection Flag 0: No additional faulty message buffer 1: Additional faulty message buffer was detected while the FMBD flag is set to 1.
5	FMBD	Faulty Message Buffer Detection Flag 0: No faulty message buffer 1: Message buffer referenced by FLXAnFRMHDS.FMB holds faulty data with a parity error.
4	ATBF2	Transient Buffer RAM B Access Error Flag 0: No access error 1: Access error occurred when reading the RAM B.
3	ATBF1	Transient Buffer RAM A Access Error Flag 0: No access error 1: Access error occurred when reading the RAM A.
2	AMR	Message RAM Access Error Flag 0: No access error 1: Access error occurred when reading the message RAM.
1, 0	—	Reserved These bits are always read as 0. When writing, always write 0

(1) FLXAnFRMHDS.MBU

Message Buffer Updated Flags

These flags indicate the number of the message buffer that was updated last by the CC. For this message buffer, the respective ND and/or MBC flag in the FLXAnFRNDAT1/2/3/4 registers and the FLXAnFRMBSC1/2/3/4 registers are also set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(2) FLXAnFRMHDS.MBT

Message Buffer Transmitted Flags

These flags indicate the number of the last successfully transmitted message buffer.

If the message buffer is configured for single-shot mode, the respective TXR flags in the FLXAnFRTRXQ1/2/3/4 registers are reset to '0'.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(3) FLXAnFRMHDS.FMB

Faulty Message Buffer Number Flags

These flags indicate that an access error occurred when reading from the message buffer referenced by FLXAnFRMHDS.FMB.

The value of this flag is only valid whenever one of AMR flag, ATBF1 flag, ATBF2 flag, or FMBD flag in the FLXAnFRMHDS register is set to 1.

This flag is not updated while the FLXAnFRMHDS.FMBD flag is 1.

This flag is cleared by the CHI command CLEAR_RAMs.

(4) FLXAnFRMHDS.CRAMP

Internal RAM Clear Flag

This flag indicates that the CHI command CLEAR_RAMs is ongoing (all bits of the message RAM, input buffer, output buffer and Transient Buffer RAM are written to '0').

This flag is set by the CHI command CLEAR_RAMs.

(5) FLXAnFRMHDS.MFMB

Multiple Faulty Message Buffer Detection Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This bit indicates that an additional faulty message buffer was detected while the FMBD flag is set.

This bit is cleared by the CHI command CLEAR_RAMs.

(6) FLXAnFRMHDS.FMBD

Faulty Message Buffer Detection Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This bit indicates that the message buffer referenced by FLXAnFRMHDS.FMB holds faulty data due to an access error.

This bit is cleared by the CHI command CLEAR_RAMs.

(7) FLXAnFRMHDS.ATBF2

Transient Buffer RAM B Access Error Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag indicates that an access error occurred when reading the Transient Buffer RAM B.

CAUTION

When this flag changes from '0' to '1', the AERR bit in the FLXAnFREIR register is set to '1'. This flag can be reset by the CHI command CLEAR_RAMs.

(8) FLXAnFRMHDS.ATBF1

Transient Buffer RAM A Access Error Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag indicates that an access error occurred when reading the Transient Buffer RAM A.

CAUTION

When this flag changes from '0' to '1', the AERR bit in the FLXAnFREIR register is set to '1'. This flag can be reset by the CHI command CLEAR_RAMs.

(9) FLXAnFRMHDS.AMR

Message RAM Access Error Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag indicates that an access error occurred when reading the message RAM.

CAUTION

When this flag changes from '0' to '1', the AERR bit in the FLXAnFREIR register is set to '1'. This flag can be reset by the CHI command CLEAR_RAMs.

21.3.9.2 FLXAnFRLDTS — FlexRay Last Dynamic Transmit Slot Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	LDTB[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	LDTA[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.63 FLXAnFRLDTS Register Contents

Bit Position	Bit Name	Function
31 to 27	—	Reserved These bits are always read as 0. When writing, always write 0.
26 to 16	LDTB[10:0]	Last Dynamic Transmission Channel B Flags
15 to 11	—	Reserved These bits are always read as 0. When writing, always write 0.
10 to 0	LDTA[10:0]	Last Dynamic Transmission Channel A Flags

(1) FLXAnFRLDTS.LDTB

Last Dynamic Transmission Channel B Flags

Store the value of vSlotCounter at the time of the last frame transmission on channel B in the dynamic segment.

Updated at the end of the dynamic segment and reset to zero if no frame was transmitted during the dynamic segment.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(2) FLXAnFRLDTS.LDTA

Last Dynamic Transmission Channel A Flags

Store the value of vSlotCounter at the time of the last frame transmission on channel A in the dynamic segment.

Updated at the end of the dynamic segment and reset to zero if no frame was transmitted during the dynamic segment.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

21.3.9.3 FLXAnFRFSR — FlexRay FIFO Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFFL[7:0]							—	—	—	—	—	RFO	RFCL	RFNE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.64 FLXAnFRFSR Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved These bits are always read as 0. When writing, always write 0.
15 to 8	RFFL[7:0]	Receive FIFO Fill Level Flags
7 to 3	—	Reserved These bits are always read as 0. When writing, always write 0.
2	RFO	Receive FIFO Overrun Flag 0: No receive FIFO overrun is detected 1: A receive FIFO overrun has been detected
1	RFCL	Receive FIFO Critical Level Flag 0: Receive FIFO is below critical level 1: Receive FIFO critical level is reached
0	RFNE	Receive FIFO Not Empty Flag 0: Receive FIFO is empty 1: Receive FIFO is not empty

(1) FLXAnFRFSR.RFFL

Receive FIFO Fill Level Flags

Indicate the number of FIFO buffers filled with new data not yet read by the host. Maximum value is 128.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(2) FLXAnFRFSR.RFO

Receive FIFO Overrun Flag

The flag is set to '1' by the CC when a receive FIFO overrun is detected.

When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. In addition, interrupt flag FLXAnFREIR.RFO is set to '1'.

The flag is cleared when the FIFO is read.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(3) FLXAnFRFSR.RFCL

Receive FIFO Critical Level Flag

This flag is set to '1' when the receive FIFO fill level FLXAnFRFSR.RFFL is equal to or greater than the critical level as configured by FLXAnFRFCL.CL.

When FLXAnFRFSR.RFCL changes from '0' to '1', bit FLXAnFRSIR.RFCL is set to '1', and if enabled, an interrupt is generated.

The flag is cleared by the CC as soon as FLXAnFRFSR.RFFL drops below FLXAnFRFCL.CL.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(4) FLXAnFRFSR.RFNE

Receive FIFO Not Empty Flag

This flag is set to '1' by the CC when a received valid frame (data or null frame depending on rejection mask) is transferred to the FIFO. In addition, interrupt flag FLXAnFRSIR.RFNE is set to '1'.

The bit is reset to '0' after the host has read all messages from the FIFO.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

21.3.9.4 FLXAnFRMHDF — FlexRay Message Handler Constraints Flags Register

Do not rewrite this register using bit manipulation instructions.

Some constraints exist for the message handler regarding bus clock frequency, message RAM configuration, and FlexRay bus traffic. To simplify software development, constraints violations are reported by setting flags in FLXAnFRMHDF.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	WAHP	TNSB	TNSA	TBFB	TBFA	FNFB	FNFA	SNUB	SNUA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.65 FLXAnFRMHDF Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 9	—	Reserved These bits are always read as 0. When writing, always write 0.
8	WAHP	Write Attempt to Header Partition Flag 0: No write attempt to header partition 1: A write attempt to header partition
7	TNSB	Transmission Not Started Channel B Flag 0: No transmission not started on channel B 1: Transmission not started on channel B
6	TNSA	Transmission Not Started Channel A Flag 0: No transmission not started on channel A 1: Transmission not started on channel A
5	TBFB	Temporary Buffer Access Failure B Flag 0: No TBF B access failure 1: TBF B access failure
4	TBFA	Temporary Buffer Access Failure A Flag 0: No TBF A access failure 1: TBF A access failure
3	FNFB	Find Sequence Not Finished Channel B Flag 0: No find sequence not finished for channel B 1: Find sequence not finished for channel B
2	FNFA	Find Sequence Not Finished Channel A Flag 0: No find sequence not finished for channel A 1: Find sequence not finished for channel A

Table 21.65 FLXAnFRMHDF Register Contents (2/2)

Bit Position	Bit Name	Function
1	SNUB	Status Not Updated Channel B Flag 0: No overload condition occurred when updating MBS for channel B 1: Message buffer status (FLXAnFRMBS) for channel B not updated
0	SNUA	Status Not Updated Channel A Flag 0: No overload condition occurred when updating MBS for channel A 1: Message buffer status (FLXAnFRMBS) for channel A not updated

(1) FLXAnFRMHDF.WAHP

Write Attempt to Header Partition Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

Outside DEFAULT_CONFIG and CONFIG state, this flag is set to '1' when the message handler tries to write message data into the header partition of the message RAM due to faulty configuration of a message buffer. The write attempt is not executed, to protect the header partition from unintended write accesses.

When this flag changes from '0' to '1', in addition interrupt flag FLXAnFREIR.MHF is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(2) FLXAnFRMHDF.TNSB Flag

Transmission Not Started Channel B

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set to '1' when the message handler was not ready to start a scheduled transmission on channel B at the action point of the configured slot.

When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.MHF is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(3) FLXAnFRMHDF.TNSA

Transmission Not Started Channel A Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set when the message handler was not ready to start a scheduled transmission on channel A at the action point of the configured slot.

When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.MHF is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(4) FLXAnFRMHDF.TBFB

Temporary Buffer Access Failure B Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set to '1' when a read or write access to TBF B requested by PRT (protocol controller) B could not complete within the specified time.

When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.MHF is set to '1'.
Reset when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMs.

(5) FLXAnFRMHDF.TBFA

Temporary Buffer Access Failure A Flag
Writing '0' has no effect on the bit value.
This bit is cleared by writing '1' to it.
This flag is set to '1' by the CC when a read or write access to TBF A requested by PRT A could not complete within the specified time.
When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.MHF is set to '1'.
Reset when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMs.

(6) FLXAnFRMHDF.FNFB

Find Sequence Not Finished Channel B Flag
Writing '0' has no effect on the bit value.
This bit is cleared by writing '1' to it.
This flag is set to '1' when the message handler, due to overload condition, was not able to finish a find sequence (scan of message RAM for matching message buffer).
When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.MHF is set to '1'.
Reset when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMs.

(7) FLXAnFRMHDF.FNFA

Find Sequence Not Finished Channel A Flag
Writing '0' has no effect on the bit value.
This bit is cleared by writing '1' to it.
This flag is set to '1' when the message handler, due to overload condition, was not able to finish a find sequence (scan of message RAM for matching message buffer).
When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.MHF is set to '1'.
Reset when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMs.

(8) FLXAnFRMHDF.SNUB

Status Not Updated Channel B Flag
Writing '0' has no effect on the bit value.
This bit is cleared by writing '1' to it.
This flag is set to '1' when the message handler, due to overload condition, was not able to update a message buffer's status MBS.
When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.MHF is set to '1'.
Reset when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMs.

(9) FLXAnFRMHDF.SNUA

Status Not Updated Channel A Flag

Writing '0' has no effect on the bit value.

This bit is cleared by writing '1' to it.

This flag is set to '1' when the message handler, due to overload condition, was not able to update a message buffer's status MBS.

When this flag changes from '0' to '1', interrupt flag FLXAnFREIR.MHF is set to '1'.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

21.3.9.5 FLXAnFRTXRQi — FlexRay Transmission Request Register i (i = 1 to 4)

The four registers reflect the state of the TXR flags of all configured message buffers. The flags are evaluated for transmit buffers only. If the number of configured message buffers is less than 128, the remaining TXR flags have no meaning.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TXRo[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXRo[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.66 FLXAnFRTXRQi Register Contents

Bit Position	Bit Name	Function
31 to 0	TXRo[31:0]	Transmission Request Flag o

(1) FLXAnFRTXRQi.TXRo (o = (i-1) × 32 to (i × 32-1))

Transmission Request Flag o

If the flag is set to '1', the respective message buffer is ready for transmission or transmission of this message buffer is in progress.

In single-shot mode, the flags are reset to '0' after transmission has completed.

This bit is cleared by the CHI command CLEAR_RAMs.

21.3.9.6 FLXAnFRNDATi — FlexRay New Data Register i (i = 1 to 4)

The four registers reflect the state of the ND flags of all configured message buffers. ND flags belonging to transmit buffers have no meaning. If the number of configured message buffers is less than 128, the remaining ND flags have no meaning.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NDm[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NDm[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.67 FLXAnFRNDATi Register Contents

Bit Position	Bit Name	Function
31 to 0	NDm[31:0]	New Data Flag m

(1) FLXAnFRNDATi.NDm (m = (i-1) × 32 to (i × 32-1))

New Data Flag m

The flags are set to '1' when a valid received data frame matches the message buffer's filter configuration, independent of the payload length received or the payload length configured for that message buffer.

The flags are not set to '1' after reception of null frames except for message buffers belonging to the receive FIFO.

An ND flag is reset to '0' when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the output buffer.

Reset when leaving CONFIG state or when entering STARTUP state.

This bit is cleared by the CHI command CLEAR_RAMs.

21.3.9.7 FLXAnFRMBSCi — FlexRay Message Buffer Status Changed Register i (i = 1 to 4)

The four registers reflect the state of the MBC flags of all configured message buffers. If the number of configured message buffers is less than 128, the remaining MBC flags have no meaning.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MBCm[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MBCm[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.68 FLXAnFRMBSCi (i = 1 to 4) Register Contents

Bit Position	Bit Name	Function
31 to 0	MBCm[31:0]	Message Buffer Status Changed Flag m

(1) FLXAnFRMBSCi.MBCm (m = (i-1) × 32 to (i × 32-1))

Message Buffer Status Changed Flag m

Indicates whether the message handler has changed one of the status flags VFRA, VFRB, SEOA, SEOB, CEOA, CEOB, SVOA, SVOB, TCIA, TCIB, ESA, ESB, MLST, FTA, FTB in the header section (see **Section 21.3.11.5, FLXAnFRMBS — FlexRay Message Buffer Status Register** and **Section 21.4.13.1, Header Partition**) of the respective message buffer.

An MBC flag is reset to '0' when the header section of the corresponding message buffer is reconfigured or when it has been transferred to the output buffer.

Reset when leaving CONFIG state or when entering STARTUP state.

This bit is cleared by the CHI command CLEAR_RAMs.

21.3.10 Input Buffer

The input buffer (IBF) has a double structure of IBF host and IBF shadow. While the host can write to IBF host, the transfer to the message RAM is done from IBF shadow. The input buffer holds the header and data sections to be transferred to the selected message buffer in the message RAM. It is used to configure the message buffers in the message RAM and to update the data sections of transmit buffers.

When updating the header section of a message buffer in the message RAM from the input buffer, the message buffer status as described in **Section 21.3.11.5, FLXAnFRMBS — FlexRay Message Buffer Status Register** is automatically reset to 0.

The header sections of message buffers belonging to the receive FIFO can only be configured or reconfigured when the CC is in DEFAULT_CONFIG or CONFIG state. For those message buffers, only the payload length and the data pointer can be configured via FLXAnFRWRHS2.PLC and FLXAnFRWRHS3.DP. All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask registers.

The data transfer between input buffer (IBF) and message RAM is described in detail in **Section 21.4.12.2, Host Access to Message RAM**.

These registers cannot be written when the input data transfer function shown **Section 21.4.16.1, Input Data Transfer** is used and the FLXAnFRITS.ITS bit is 1.

21.3.10.1 FLXAnFRWRDSx — FlexRay Write Data Section Register x (x = 1 to 64)

This register holds the data words to be transferred to the data section of the specified message buffer. The number of data words (DW_x) written to the message RAM is defined by the payload length configured in the FLXAnFRWRHS2.PLC bit.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MD[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.69 FLXAnFRWRDSx Register Contents

Bit Position	Bit Name	Function
31 to 0	MD[31:0]	Message Data

(1) FLXAnFRWRDSx.MD

Message Data Bits

For information about the byte alignment of the message data in this register, see **Section 21.4.17, Byte Alignment**.

CAUTIONS

1. In case FLXAnFRWRHS2.PLC specifies an odd payload length, the remaining message data bytes are unused.
 2. When writing to FLXAnFRWRDSx, each 32-bit word has to be filled up by access in a single 32-bit unit, in two consecutive 16-bit units, or in four consecutive 8-bit units before transfer from the input buffer to the message RAM will start. If not all bytes of a 32-bit word have been written by the host (8- or 16-bit access only), FLXAnFRWRDSx holds partly undefined data.
Reset by the CHI command CLEAR_RAMs.
-

21.3.10.2 FLXAnFRWRHS1 — FlexRay Write Header Section Register 1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	MBI	TXM	PPIT	CFG	CH[1:0]		—	CYC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	FID[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.70 FLXAnFRWRHS1 Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved These bits are always read as 0. When writing, always write 0.
29	MBI	Message Buffer Interrupt 0: The corresponding message buffer interrupt is disabled 1: The corresponding message buffer interrupt is enabled
28	TXM	Transmission Mode 0: Continuous mode 1: Single-shot mode
27	PPIT	Payload Preamble Indicator Transmit 0: Payload Preamble Indicator is set to '0' 1: Payload Preamble Indicator is set to '1'
26	CFG	Message Buffer Direction Configuration 0: The corresponding buffer is configured as receive buffer 1: The corresponding buffer is configured as transmit buffer
25,24	CH[1:0]	Channel Filter Control
23	—	Reserved This bit is always read as 0. When writing, always write 0.
22 to 16	CYC[6:0]	Cycle Code
15 to 11	—	Reserved These bits are always read as 0. When writing, always write 0.
10 to 0	FID[10:0]	Frame ID

(1) FLXAnFRWRHS1.MBI

Message Buffer Interrupt Enable

This bit enables the receive/transmit interrupt for the corresponding message buffer.

After a dedicated receive buffer has been updated by the message handler, flag FLXAnFRSIR.RXI and/or FLXAnFRSIR.MBSI are set to '1'. After a transmission has completed, flag FLXAnFRSIR.TXI is set to '1'.

(2) FLXAnFRWRHS1.TXM

Transmission Mode Setting

This bit selects transmit mode of the corresponding message buffer. For transmit mode, see **Section 21.4.9.3, Transmit Buffers**.

(3) FLXAnFRWRHS1.PPIT

Payload Preamble Indicator Transmit

This bit is used to control the state of the Payload Preamble Indicator in transmit frames of the corresponding message buffer.

If the bit is set to '1' in a static message buffer, the respective message buffer holds network management information. If the bit is set to '1' in a dynamic message buffer, the first two bytes of the payload segment may be used for message ID filtering by the receiver. Message ID filtering of received FlexRay frames is not supported by the FlexRay module, but can be done by the host.

(4) FLXAnFRWRHS1.CFG

Message Buffer Direction Configuration

This bit is used to configure the corresponding buffer as transmit buffer or as receive buffer. For message buffers belonging to the receive FIFO, the bit is disabled.

If an unused area of at least 32 bits is not allocated at the start of the data partition, set this bit to 1 to select allocation of the data section of the message buffer as a transmission buffer immediately following (after the last buffer of) the header partition.

(5) FLXAnFRWRHS1.CH

Channel Filter Control

The 2-bit channel filtering field associated with each buffer serves as a filter for receive buffers, and as a control field for transmit buffers.

CH[1:0]	Transmit Buffer transmit frame on	Receive Buffer store frame received from
00	Transmission prohibited	ignore frame
01	channel A	channel A
10	channel B	channel B
11	both channels (static segment only)	channel A or B (store first semantically valid frame; static segment only)

CAUTION

If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to '1', no transmission nor reception of frames is performed. (same function as CH = "00_B")

(6) FLXAnFRWRHS1.CYC

Cycle Code

The 7-bit cycle code determines the cycle set used for cycle counter filtering.

For details about the configuration of the cycle code, see **Section 21.4.8.2, Cycle Counter Filtering**.

(7) FLXAnFRWRHS1.FID

Frame ID

Frame ID of the selected message buffer. The frame ID defines the slot number for transmission/reception of the respective message.

Message buffers with frame ID = '0' are considered as invalid.

21.3.10.3 FLXAnFRWRHS2 — FlexRay Write Header Section Register 2

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	PLC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	CRC[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.71 FLXAnFRWRHS2 Register Contents

Bit Position	Bit Name	Function
31 to 23	—	Reserved These bits are always read as 0. When writing, always write 0.
22 to 16	PLC[6:0]	Payload Length Configured
15 to 11	—	Reserved These bits are always read as 0. When writing, always write 0.
10 to 0	CRC[10:0]	Header CRC (vRF!Header!HeaderCRC) Receive Buffer: Configuration is not required Transmit Buffer: Header CRC is configured

(1) FLXAnFRWRHS2.PLC

Payload Length Configured

Length of data section (number of 2-byte words) as configured by the host.

During static segment, the static frame payload length as configured by FLXAnFRMHDC.SFDL defines the payload length for all static frames. If the payload length configured by FLXAnFRWRHS2.PLC is shorter than this value, padding bytes are inserted to ensure that frames have proper physical length. The padding pattern is “0000_H” (see **Section 21.4.9.3, Transmit Buffers**).

(2) FLXAnFRWRHS2.CRC

Header CRC (vRF!Header!HeaderCRC)

Setting of the receive buffer is not required.

Transmitting of the message buffer needs the header CRC calculation and setting.

For calculation of the header CRC, the payload length of the frame to be sent has to be considered. In static segment, the payload length of all frames is configured by FLXAnFRMHDC.SFDL.

21.3.10.4 FLXAnFRWRHS3 — FlexRay Write Header Section Register 3

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DP[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.72 FLXAnFRWRHS3 Register Contents

Bit Position	Bit Name	Function
31 to 11	—	Reserved These bits are always read as 0. When writing, always write 0.
10 to 0	DP[10:0]	Data Pointer

(1) FLXAnFRWRHS3.DP

Data Pointer

Configures the pointer to the first 32-bit word of the data section of the addressed message buffer in the message RAM.

21.3.10.5 FLXAnFRIBCM — FlexRay Input Buffer Command Mask Register

Configures how the message buffer in the message RAM selected by register FLXAnFRIBCR is updated. When IBF host and IBF shadow are swapped, also mask bits FLXAnFRIBCM.LHSH, FLXAnFRIBCM.LDSH, and FLXAnFRIBCM.STXRH are swapped with bits FLXAnFRIBCM.LHSS, FLXAnFRIBCM.LDSS, and FLXAnFRIBCM.STXRS.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	STXRS	LDSS	LHSS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	STXRH	LDSH	LHSH
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 21.73 FLXAnFRIBCM Register Contents

Bit Position	Bit Name	Function
31 to 19	—	Reserved These bits are always read as 0. When writing, always write 0.
18	STXRS	Set Transmission Request Shadow Flag 0: Reset TXR flag 1: Set TXR flag, transmit buffer released for transmission (operation ongoing or finished)
17	LDSS	Load Data Section Shadow Flag 0: Data section is not updated 1: Data section is transferred (transfer ongoing or finished)
16	LHSS	Load Header Section Shadow Flag 0: Header section is not updated 1: Header is transferred (transfer ongoing or finished)
15 to 3	—	Reserved These bits are always read as 0. When writing, always write 0.
2	STXRH	Set Transmission Request Host Bit 0: Reset TXR flag 1: Set TXR flag, transmit buffer released for transmission
1	LDSH	Load Data Section Host Bit 0: Data section is not updated 1: Data section is transferred
0	LHSH	Load Header Section Host Bit 0: Header section is not updated 1: Header is transferred

(1) FLXAnFRIBCM.STXRS

Set Transmission Request Shadow Flag

(2) FLXAnFRIBCM.LDSS

Load Data Section Shadow Flag

(3) FLXAnFRIBCM.LHSS

Load Header Section Shadow Flag

(4) FLXAnFRIBCM.STXRH

Set Transmission Request Host Bit

If this bit is set to '1', the TXR flag for the selected message buffer is set in the FLXAnFRTXRQ1/2/3/4 registers to release the message buffer for transmission. In single-shot mode, the flag is cleared after transmission has completed. TXR is enabled for transmit buffers only.

(5) FLXAnFRIBCM.LDSH

Set Load Data Section Host Bit

(6) FLXAnFRIBCM.LHSH

Set Load Header Section Host Bit

21.3.10.6 FLXAnFRIBCR — FlexRay Input Buffer Command Request Register

When the host writes the number of the target message buffer in the message RAM to FLXAnFRIBCR.IBRH, IBF host and IBF shadow are switched. In addition the message buffer numbers stored under FLXAnFRIBCR.IBRH and FLXAnFRIBCR.IBRS are also switched (see **Section 21.4.12.2(1), Data Transfer from Input Buffer to Message RAM**).

With this write operation, the FLXAnFRIBCR.IBSYS is set to '1'. The message handler then starts to transfer the contents of IBF shadow to the message buffer in the message RAM selected by FLXAnFRIBCR.IBRS.

After the transfer between IBF shadow and the message RAM has completed, FLXAnFRIBCR.IBSYS is set back to '0' and the next transfer to the message RAM may be started by the host by writing the respective target message buffer number to FLXAnFRIBCR.IBRH.

If a write access to FLXAnFRIBCR.IBRH occurs while FLXAnFRIBCR.IBSYS is '1', FLXAnFRIBCR.IBSYH is set to '1'. After completion of the ongoing data transfer from IBF Shadow to the message RAM, IBF host and IBF shadow are switched, FLXAnFRIBCR.IBSYH is reset to '0'. FLXAnFRIBCR.IBSYS remains set to '1', and the next transfer to the message RAM is started. In addition, the message buffer numbers stored under FLXAnFRIBCR.IBRH and FLXAnFRIBCR.IBRS are also switched.

Any write access to an input buffer register while both FLXAnFRIBCR.IBSYS and FLXAnFRIBCR.IBSYH are set to '1' will cause the error flag FLXAnFREIR.IIBA to be set to '1'.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IBSYS	—	—	—	—	—	—	—	—	IBRS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IBSYH	—	—	—	—	—	—	—	—	IBRH[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.74 FLXAnFRIBCR Register Contents

Bit Position	Bit Name	Function
31	IBSYS	Input Buffer Busy Shadow Flag 0: Transfer between IBF shadow and message RAM completed 1: Transfer between IBF shadow and message RAM in progress
30 to 23	—	Reserved These bits are always read as 0. When writing, always write 0.
22 to 16	IBRS[6:0]	Input Buffer Request ShadowFlags
15	IBSYH	Input Buffer Busy Host Flag 0: No request pending 1: Request while transfer between IBF shadow and message RAM in progress
14 to 7	—	Reserved These bits are always read as 0. When writing, always write 0.
6 to 0	IBRH[6:0]	Input Buffer Request Host

(1) FLXAnFRIBCR.IBSYS

Input Buffer Busy Shadow Flag

Set to '1' after writing FLXAnFRIBCR.IBRH.

This bit indicates transmitting between the IBF shadow and the message RAM is ongoing.

When the transfer between IBF shadow and the message RAM has completed, FLXAnFRIBCR.IBSYS is set back to '0'.

(2) FLXAnFRIBCR.IBRS

Input Buffer Request Shadow Flags

Number of the target message buffer actually updated or lately updated.

(3) FLXAnFRIBCR.IBSYH

Input Buffer Busy Host Flag

Set to '1' by writing FLXAnFRIBCR.IBRH while FLXAnFRIBCR.IBSYS is still '1'.

This bit indicates transmitting between the IBF shadow and the message RAM is ongoing.

After the ongoing transfer between IBF shadow and the message RAM has completed, the FLXAnFRIBCR.IBSYH is set back to '0'.

(4) FLXAnFRIBCR.IBRH

Input Buffer Request Host

Selects the target message buffer in the message RAM for data transfer from Input Buffer.

21.3.11 Output Buffer

The output buffer has a double structure of output buffer host and output buffer shadow and is used to read out message buffers from the message RAM. While the host can read from OBF host, the message handler transfers the selected message buffer from message RAM to output buffer shadow. The data transfer between message RAM and OBF is described in **Section 21.4.12.2(2), Data Transfer from Message RAM to Output Buffer**.

These registers cannot be written when the output data transfer function shown in **Section 21.4.16.2, Output Data Transfer**, in output data transfer is used and the FLXAnFROTS.OTS bit is 1.

21.3.11.1 FLXAnFRRDDSx — FlexRay Read Data Section Register x (x = 1 to 64)

These registers hold the data words read from the data section of the specified message buffer. The number of data words read from the message RAM is defined by the payload length configured in the FLXAnFRRDHS2.PLC bit.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MD[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MD[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.75 FLXAnFRRDDSx Register Contents

Bit Position	Bit Name	Function
31 to 0	MD[31:0]	Message Data Flags

(1) FLXAnFRRDDSx.MD

Message Data Flags

For information about the byte alignment of the data words in this register, see **Section 21.4.17, Byte Alignment**.

CAUTION

In case FLXAnFRWRHS2.PLC specifies an odd payload length, the remaining message data bytes are unused.

Reset by the CHI command CLEAR_RAMs.

21.3.11.2 FLXAnFRRDHS1 — FlexRay Read Header Section Register 1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	MBI	TXM	PPIT	CFG	CH[1:0]	—	CYC[6:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	FID[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.76 FLXAnFRRDHS1 Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved These bits are always read as 0. When writing, always write 0.
29	MBI	Message Buffer Interrupt Enable Flag
28	TXM	Transmission Mode Flag
27	PPIT	Payload Preamble Indicator Transmit Flag
26	CFG	Message Buffer Direction Configuration Bit Flag
25, 24	CH[1:0]	Channel Filter Control Flag
23	—	Reserved This bit is always read as 0. When writing, always write 0.
22 to 16	CYC[6:0]	Cycle Code
15 to 11	—	Reserved These bits are always read as 0. When writing, always write 0.
10 to 0	FID[10:0]	Frame ID

(1) FLXAnFRRDHS1.MBI

Message Buffer Interrupt Enable Flag

Values as configured by the host via FLXAnFRWRHS1.MBI.

In case that the message buffer read from the message RAM belongs to the receive FIFO, this bit is set to '0'.

(2) FLXAnFRRDHS1.TXM

Transmission Mode Flag

Values as configured by the host via FLXAnFRWRHS1.TXM.

In case that the message buffer read from the message RAM belongs to the receive FIFO, this bit is set to '0'.

(3) FLXAnFRRDHS1.PPIT

Payload Preamble Indicator Transmit Flag

Values as configured by the host via FLXAnFRWRHS1.PPIT.

In case that the message buffer read from the message RAM belongs to the receive FIFO, this bit is set to '0'.

(4) FLXAnFRRDHS1.CFG

Message Buffer Direction Configuration Flag

Values as configured by the host via FLXAnFRWRHS1.CFG.

In case that the message buffer read from the message RAM belongs to the receive FIFO, this bit is set to '0'.

(5) FLXAnFRRDHS1.CH

Channel Filter Control Flag

Values as configured by the host via FLXAnFRWRHS1.CH.

In case that the message buffer read from the message RAM belongs to the receive FIFO, these bits are set to '0'.

(6) FLXAnFRRDHS1.CYC

Cycle Code

Values as configured by the host via FLXAnFRWRHS1.CYC.

In case that the message buffer read from the message RAM belongs to the receive FIFO, these bits are set to '0'.

(7) FLXAnFRRDHS1.FID

Frame ID

Values as configured by the host via FLXAnFRWRHS1.FID

In case that the message buffer read from the message RAM belongs to the receive FIFO, these bits are holding the received frame ID.

21.3.11.3 FLXAnFRRDHS2 — FlexRay Read Header Section Register 2

Value after reset: 0000 0000_H

For message buffers belonging to the Static Buffer area or Static and Dynamic Buffer area, FLXAnFRWRHS2 is updated from data frames only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	PLR[6:0]						—	PLC[6:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	CRC[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.77 FLXAnFRRDHS2 Register Contents

Bit Position	Bit Name	Function
31	—	Reserved This bit is always read as 0. When writing, always write 0.
30 to 24	PLR[6:0]	Receive Frame Payload Length Flags (vRF!Header!Length)
23	—	Reserved This bit is always read as 0. When writing, always write 0.
22 to 16	PLC[6:0]	Configured Payload Length Flags
15 to 11	—	Reserved These bits are always read as 0. When writing, always write 0.
10 to 0	CRC[10:0]	Header CRC Flags (vRF!Header!HeaderCRC)

(1) FLXAnFRRDHS2.PLR

Receive Frame Payload Length Flags (vRF!Header!Length)

Payload length (vRF!Header!Length) value updated from received data frames (exception: To be updated even through null frames are received in the message buffer specified to the receive FIFO).

(2) FLXAnFRRDHS2.PLC

Configured Payload Length Flags

Length of data section (number of 2-byte words) as configured by the host.

(3) FLXAnFRRDHS2.CRC

Header CRC Flags (vRF!Header!HeaderCRC)

Receive Buffer: Header CRC (vRF!Header!HeaderCRC) updated from received data frames

Transmit Buffer: Header CRC configured by the host

(4) Data storage

When a message is stored into a message buffer, the following processes with respect to payload length received and payload length configured are implemented:

$FLXAnFRRDHS2.PLR > FLXAnFRRDHS2.PLC$:

The payload data stored in the message buffer is truncated to the payload length configured if $FLXAnFRRDHS2.PLC$ even or else truncated to $FLXAnFRRDHS2.PLC + 1$.

$FLXAnFRRDHS2.PLR \leq FLXAnFRRDHS2.PLC$:

The received payload data is stored into the message buffers data section. The remaining data bytes of the data section as configured by $FLXAnFRRDHS2.PLC$ are filled with undefined value.

$FLXAnFRRDHS2.PLR = \text{zero}$:

The message buffer's data section is filled with undefined value.

$FLXAnFRRDHS2.PLC = \text{zero}$:

No data section is configured in the message buffer. No data is stored into the message buffer's data section.

CAUTION

1. The message RAM is organized in 4-byte words. When received data is stored into a message buffer's data section, the number of 2-byte data words written into the message buffer is $FLXAnFRRDHS2.PLC$ rounded to the next even value.
2. $FLXAnFRRDHS2.PLC$ should be configured identical for all message buffers belonging to the receive FIFO. For message buffers belonging to the Static Buffer area or Static and Dynamic Buffer area, $FLXAnFRWRHS2$ is updated from data frames only.

21.3.11.4 FLXAnFRRDHS3 — FlexRay Read Header Section Register 3

For message buffers belonging to the Static Buffer area or Static and Dynamic Buffer area, FLXAnFRWRHS3 is updated from data frames only.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	RES	PPI	NFI	SYN	SFI	RCI	—	—	RCC[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DP[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.78 FLXAnFRRDHS3 Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved These bits are always read as 0. When writing, always write 0.
29	RES	Reserved Bit Indicator Flag (vRF!Header!Reserved)
28	PPI	Payload Preamble Indicator Flag (vRF!Header!PPIndicator)
27	NFI	Null Frame Indicator Flag (vRF!Header!NFIndicator) 0: Up to now, no data frame has been stored into the respective message buffer 1: At least one data frame has been stored into the respective message buffer
26	SYN	Sync Frame Indicator Flag (vRF!Header!SyFIndicator) 0: The received frame is not a sync frame 1: The received frame is a sync frame
25	SFI	Startup Frame Indicator Flag (vRF!Header!SuFIndicator) 0: The received frame is not a startup frame 1: The received frame is a startup frame
24	RCI	Received on Channel Indicator Flag (vSS!Channel) 0: Frame received on channel B 1: Frame received on channel A
23, 22	—	Reserved These bits are always read as 0. When writing, always write 0.
21 to 16	RCC[5:0]	Receive Cycle Count (vRF!Header!CycleCount)
15 to 11	—	Reserved These bits are always read as 0. When writing, always write 0.
10 to 0	DP[10:0]	Data Pointer Flags

(1) FLXAnFRRDHS3.RES

Reserved Bit Indicator Flag (vRF!Header!Reserved)

Reflects the value of the received reserved bit. The reserved bit is transmitted as '0'.

(2) FLXAnFRRDHS3.PPI

Payload Preamble Indicator Flag (vRF!Header!PPIIndicator)

The payload preamble indicator defines whether a network management vector or message ID is contained in the payload segment of the received frame.

0 = The payload segment of the received frame does not contain a network management vector nor a message ID

1 = Static segment: contains network management vector is included in the first part of the payload

Dynamic segment: contains message ID in the first part of the payload

(3) FLXAnFRRDHS3.NFI

Null Frame Indicator Flag (vRF!Header!NFIndicator)

Is set to '1' after storage of the first received data frame.

(4) FLXAnFRRDHS3.SYN

Sync Frame Indicator Flag (vRF!Header!SyFIndicator)

A sync frame is indicated by the sync frame indicator.

(5) FLXAnFRRDHS3.SFI

Startup Frame Indicator Flag (vRF!Header!SuFIndicator)

A startup frame is indicated by the startup frame indicator.

(6) FLXAnFRRDHS3.RCI

Received on Channel Indicator Flag (vSS!Channel)

Indicates the channel from which the received data frame was taken to update the respective receive buffer.

(7) FLXAnFRRDHS3.RCC

Receive Cycle Count (vRF!Header!CycleCount)

Cycle counter value updated from received data frame is read.

(8) FLXAnFRRDHS3.DP

Data Pointer Flags

This flag indicates the position of the first 32-bit word of the data section of the addressed message buffer in the message RAM.

The bit value is the same as that set in the FLXAnFRWRHS3.DP bit.

21.3.11.5 FLXAnFRMBS — FlexRay Message Buffer Status Register

The message buffer status is updated by the CC with respect to the assigned channel(s) latest at the end of the slot following the slot assigned to the message buffer.

The flags are updated only when the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state.

If only one channel (A or B) is assigned to a message buffer, the channel-specific status flags of the other channel are written to zero. If both channels are assigned to a message buffer, the channel-specific status flags of both channels are updated.

The message buffer status is updated only when the slot counter reached the configured frame ID and when the cycle counter filter matched. When the host updates a message buffer via input buffer, all FLXAnFRMBS flags are reset to zero independent of which FLXAnFRIBCM bits are set or not.

For details about receive/transmit filtering, see **Section 21.4.8, Filtering and Masking**, **Section 21.4.9, Transmit Process** and **Section 21.4.10, Receive Process**.

Whenever the message handler changes one of the flags FLXAnFRMBS.VFRA, FLXAnFRMBS.VFRB, FLXAnFRMBS.SEOA, FLXAnFRMBS.SEOB, FLXAnFRMBS.CEOA, FLXAnFRMBS.CEOB, FLXAnFRMBS.SVOA, FLXAnFRMBS.SVOB, FLXAnFRMBS.TCIA, FLXAnFRMBS.TCIB, FLXAnFRMBS.ESA, FLXAnFRMBS.ESB, FLXAnFRMBS.MLST, FLXAnFRMBS.FTA, FLXAnFRMBS.FTB, the respective message buffer's MBC flag in registers FLXAnFRMBS1/2/3/4 is set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	RESS	PPIS	NFIS	SYNS	SFIS	RCIS	—	—	CCS[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTB	FTA	—	MLST	ESB	ESA	TCIB	TCIA	SVOB	SVOA	CEOB	CEOA	SEOB	SEOA	VFRB	VFRA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.79 FLXAnFRMBS Register Contents (1/3)

Bit Position	Bit Name	Function
31, 30	—	Reserved These bits are always read as 0. When writing, always write 0.
29	RESS	Reserved Bit Status Flag (vRF!Header!Reserved)
28	PPIS	Payload Preamble Indicator Status Flag (vRF!Header!PPIndicator) 0: PPI indicator set to '0' 1: PPI indicator set to '1'
27	NFIS	Null Frame Indicator Status Flag (vRF!Header!NFIndicator) 0: Received frame is a null frame 1: Received frame is not a null frame

Table 21.79 FLXAnFRMBS Register Contents (2/3)

Bit Position	Bit Name	Function
26	SYNS	Sync Frame Indicator Status Flag (vRF!Header!SyFIndicator) 0: No sync frame received 1: The received frame is a sync frame
25	SFIS	Startup Frame Indicator Status Flag (vRF!Header!SuFIndicator) 0: No startup frame received 1: The received frame is a startup frame
24	RCIS	Received Channel Indicator Status Flag (vSS!Channel) 0: Frame received on channel B 1: Frame received on channel A
23, 22	—	Reserved These bits are always read as 0. When writing, always write 0.
21 to 16	CCS[5:0]	Cycle Count Status Flags
15	FTB	Frame Transmitted on Channel B Flag 0: No data frame transmitted on channel B 1: Data frame transmitted on channel B
14	FTA	Frame Transmitted on Channel A Flag 0: No data frame transmitted on channel A 1: Data frame transmitted on channel A
13	—	Reserved This bit is always read as 0. When writing, always write 0.
12	MLST	Message Lost Flag 0: No message lost 1: Unprocessed message was overwritten
11	ESB	Empty Slot Channel B Flag 0: Bus activity detected in the assigned slot on channel B 1: No bus activity detected in the assigned slot on channel B
10	ESA	Empty Slot Channel A Flag 0: Bus activity detected in the assigned slot on channel A 1: No bus activity detected in the assigned slot on channel A
9	TCIB	Transmission Conflict Indication Channel B Flag (vSS!TxConflictB) 0: No transmission conflict occurred on channel B 1: Transmission conflict occurred on channel B
8	TCIA	Transmission Conflict Indication Channel A Flag (vSS!TxConflictA) 0: No transmission conflict occurred on channel A 1: Transmission conflict occurred on channel A
7	SVOB	Slot Boundary Violation Observed on Channel B Flag (vSS!BViolationB) 0: No slot boundary violation observed on channel B 1: Slot boundary violation observed on channel B
6	SVOA	Slot Boundary Violation Observed on Channel A Flag (vSS!BViolationA) 0: No slot boundary violation observed on channel A 1: Slot boundary violation observed on channel A
5	CEOB	Content Error Observed on Channel B Flag (vSS!ContentErrorB) 0: No content error observed on channel B 1: Content error observed on channel B
4	CEOA	Content Error Observed on Channel A Flag (vSS!ContentErrorA) 0: No content error observed on channel A 1: Content error observed on channel A

Table 21.79 FLXAnFRMBS Register Contents (3/3)

Bit Position	Bit Name	Function
3	SEOB	Syntax Error Observed on Channel B Flag (vSS!SyntaxErrorB) 0: No syntax error observed on channel B 1: Syntax error observed on channel B
2	SEOA	Syntax Error Observed on Channel A Flag (vSS!SyntaxErrorA) 0: No syntax error observed on channel A 1: Syntax error observed on channel A
1	VFRB	Valid Frame Received on Channel B Flag (vSS!ValidFrameB) 0: No valid frame received on channel B 1: Valid frame received on channel B
0	VFRA	Valid Frame Received on Channel A Flag (vSS!ValidFrameA) 0: No valid frame received on channel A 1: Valid frame received on channel A

(1) FLXAnFRMBS.RESS

Reserved Bit Status Flag (vRF!Header!Reserved)

Reflects the state of the received reserved bit. The reserved bit is transmitted as '0'.

For receive buffers (FLXAnFRWRHS1.CFG = '0'), this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers, the flag has no meaning and should be ignored.

(2) FLXAnFRMBS.PPIS

Payload Preamble Indicator Status Flag (vRF!Header!PPIndicator)

The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame.

For receive buffers (FLXAnFRWRHS1.CFG = '0'), this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers, the flag has no meaning and should be ignored.

0 = PPI indicator set to '0'

The payload segment of the received frame does not contain a network management vector or a message ID

1 = PPI indicator set to '1'

Static segment: contains network management vector at the beginning of the payload

Dynamic segment: contains message ID at the beginning of the payload

(3) FLXAnFRMBS.NFIS

Null Frame Indicator Status Flag (vRF!Header!NFIndicator)

If set to '0' the payload segment of the received frame contains no usable data.

For receive buffers (FLXAnFRWRHS1.CFG = '0'), this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers, the flag has no meaning and should be ignored.

(4) FLXAnFRMBS.SYNS

Sync Frame Indicator Status Flag (vRF!Header!SyFIndicator)

A sync frame is indicated by the sync frame indicator.

For receive buffers (FLXAnFRWRHS1.CFG = '0'), this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers, the flag has no meaning and should be ignored.

(5) FLXAnFRMBS.SFIS

Startup Frame Indicator Status Flag (vRF!Header!SuFIndicator)

The startup frame indicator specifies a startup frame.

For receive buffers (FLXAnFRWRHS1.CFG = '0'), this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers, the flag has no meaning and should be ignored.

(6) FLXAnFRMBS.RCIS

Received Channel Indicator Status Flag (vSS!Channel)

Indicates the channel on which the frame was received.

For receive buffers (FLXAnFRWRHS1.CFG = '0'), this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers, the flag has no meaning and should be ignored.

(7) FLXAnFRMBS.CCS

Cycle Count Status Flag

Actual cycle count at a status update is read.

(8) FLXAnFRMBS.FTB

Frame Transmitted on Channel B Flag

Indicates that this node has transmitted a data frame in the configured slot on channel B.

CAUTION

The FlexRay protocol specification requires that FLXAnFRMBS.FTB can only be reset by the host. Therefore the Cycle Count Status FLXAnFRMBS.CCS for this bit is only valid for the cycle where the bit is set to '1'.

(9) FLXAnFRMBS.FTA

Frame Transmitted on Channel A Flag

Indicates that this node has transmitted a data frame in the configured slot on channel A.

CAUTION

The FlexRay protocol specification requires that FLXAnFRMBS.FTA can only be reset by the host. Therefore the Cycle Count Status FLXAnFRMBS.CCS for this bit is only valid for the cycle where this bit is set to '1'.

(10) FLXAnFRMBS.MLST

Message Lost Flag

The flag is set in case the host did not read the message before the message buffer was updated from a received data frame.

Not affected by reception of null frames except for message buffers belonging to the receive FIFO. The flag is reset to '0' by a host write to the message buffer via IBF or when a new message is stored into the message buffer after the message buffers ND flag was reset to '0' by reading out the message buffer via OBF.

(11) FLXAnFRMBS.ESB

Empty Slot Channel B Flag

In an empty slot, the bus is in idle state. This means that any frame transmission is not detected. This state can be checked in static and dynamic slots.

(12) FLXAnFRMBS.ESA

Empty Slot Channel A Flag

In an empty slot, the bus is in idle state. This means that any frame transmission is not detected. This state can be checked in static and dynamic slots.

(13) FLXAnFRMBS.TCIB

Transmission Conflict Indication Channel B Flag (vSS!TxConflictB)

A transmission conflict indication is set to '1' if a transmission conflict has occurred on channel B.

(14) FLXAnFRMBS.TCIA

Transmission Conflict Indication Channel A Flag (vSS!TxConflictA)

A transmission conflict indication is set if a transmission conflict has occurred on channel A.

(15) FLXAnFRMBS.SVOB

Slot Boundary Violation Observed on Channel B Flag (vSS!BViolationB)

A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on channel B.

(16) FLXAnFRMBS.SVOA

Slot Boundary Violation Observed on Channel A Flag (vSS!BViolationA)

A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on channel A.

(17) FLXAnFRMBS.CEOB

Content Error Observed on Channel B Flag (vSS!ContentErrorB)

A content error was observed in the assigned slot on channel B.

(18) FLXAnFRMBS.CEOA

Content Error Observed on Channel A Flag (vSS!ContentErrorA)

A content error was observed in the assigned slot on channel A.

(19) FLXAnFRMBS.SEOB

Syntax Error Observed on Channel B Flag (vSS!SyntaxErrorB)

A syntax error was observed in the assigned slot on channel B.

(20) FLXAnFRMBS.SEOA

Syntax Error Observed on Channel A Flag (vSS!SyntaxErrorA)

A syntax error was observed in the assigned slot on channel A.

(21) FLXAnFRMBS.VFRB

Valid Frame Received on Channel B Flag (vSS!ValidFrameB)

This flag is set to “1” when a valid frame was received on channel B.

(22) FLXAnFRMBS.VFRA

Valid Frame Received on Channel A Flag (vSS!ValidFrameA)

This flag is set to “1” when a valid frame was received on channel A.

21.3.11.6 FLXAnFROBCM — FlexRay Output Buffer Command Mask Register

This register configures how the output buffer is updated from the message buffer in the message RAM selected by FLXAnFROBCR.OBRS.

Mask bits FLXAnFROBCM.RDSS and FLXAnFROBCM.RHSS are copied to the register internal storage when a message RAM transfer is requested by FLXAnFROBCR.REQ.

When OBF host and OBF shadow are switched, mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are switched with the register internal storage to keep them attached to the respective output buffer transfer.

The data transfer between output buffer and message RAM is described in detail in **Section 21.4.12.2(2), Data Transfer from Message RAM to Output Buffer**.

CAUTION

After the transfer of the header section from the message RAM to OBF shadow has completed, the message buffer status changed flag MBC of the selected message buffer in the FLXAnFRMBSC1/2/3/4 registers is cleared. After the transfer of the data section from the message RAM to OBF shadow has completed, the new data flag ND of the selected message buffer in the FLXAnFRNDAT1/2/3/4 registers is cleared.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDSH	RHSH
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDSS	RHSS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 21.80 FLXAnFROBCM Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 18	—	Reserved These bits are always read as 0. When writing, always write 0.
17	RDSH	Read Data Section Host Flag 0: Data section is not read 1: Data section selected for transfer from message RAM to Output Buffer
16	RHSH	Read Header Section Host Flag 0: Header section is not read 1: Header section selected for transfer from message RAM to Output Buffer
15 to 2	—	Reserved These bits are always read as 0. When writing, always write 0.
1	RDSS	Read Data Section Shadow Bit 0: Data section is not read 1: Data section selected for transfer from message RAM to Output Buffer

Table 21.80 FLXAnFROBCM Register Contents (2/2)

Bit Position	Bit Name	Function
0	RHSS	Read Header Section Shadow Bit 0: Header section is not read 1: Header section selected for transfer from message RAM to Output Buffer

(1) FLXAnFROBCM.RDSH

Read Data Section Host Flag

(2) FLXAnFROBCM.RHSH

Read Header Section Host Flag

(3) FLXAnFROBCM.RDSS

Read Data Section Shadow Bit

(4) FLXAnFROBCM.RHSS

Read Header Section Shadow Bit

21.3.11.7 FLXAnFROBCR — FlexRay Output Buffer Command Request Register

After setting bit FLXAnFROBCR.REQ to '1' while FLXAnFROBCR.OBSYS is '0', FLXAnFROBCR.OBSYS is automatically set to '1', FLXAnFROBCR.OBRS is copied to the register internal storage, mask bits FLXAnFROBCM.RDSS and FLXAnFROBCM.RHSS are copied to register FLXAnFROBCM internal storage, and the transfer of the message buffer selected by FLXAnFROBCR.OBRS from the message RAM to OBF shadow is started. When the transfer between the message RAM and OBF shadow has completed, this is signaled by setting FLXAnFROBCM.OBSYS back to '0'.

By setting bit FLXAnFROBCR.VIEW to '1' while FLXAnFROBCR.OBSYS is '0', OBF host and OBF shadow are switched. Additionally mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are switched with the register FLXAnFROBCM internal storage to keep them attached to the respective output buffer transfer. FLXAnFROBCR.OBRH signals the number of the message buffer currently accessible by the host.

If bits FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW are set to '1' with the same write access while FLXAnFROBCR.OBSYS is '0', FLXAnFROBCR.OBSYS is automatically set to '1' and OBF shadow and OBF host are switched. Additionally mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are switched with the registers internal storage to keep them attached to the respective output buffer transfer. Afterwards FLXAnFROBCR.OBRS is copied to the register internal storage, and the transfer of the selected message buffer from the message RAM to OBF shadow is started. While the transfer is ongoing the host can read the message buffer transferred by the previous transfer from OBF host. When the current transfer between message RAM and OBF shadow has completed, this is signaled by setting FLXAnFROBCR.OBSYS back to '0'.

Any write access to FLXAnFROBCR[15:8] while FLXAnFROBCR.OBSYS is set to '1' will cause the error flag FLXAnFREIR.IOBA to be set to '1'. In this case, this write access has no effect and the output buffer will not be changed.

The data transfer between output buffer and message RAM is described in detail in **Section 21.4.12.2(2), Data Transfer from Message RAM to Output Buffer**.

Value after reset:		0000 0000 _H																				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16						
	—	—	—	—	—	—	—	—	—	OBRH[6:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
	OBSYS	—	—	—	—	—	REQ	VIEW	—	OBRH[6:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W					

Table 21.81 FLXAnFROBCR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 23	—	Reserved These bits are always read as 0. When writing, always write 0.
22 to 16	OBRH[6:0]	Output Buffer Request Host Flag
15	OBSYS	Output Buffer Busy Shadow Flag 0: No transfer in progress 1: Transfer between message RAM and OBF shadow in progress

Table 21.81 FLXAnFROBCR Register Contents (2/2)

Bit Position	Bit Name	Function
14 to 10	—	Reserved These bits are always read as 0. When writing, always write 0.
9	REQ	Request Message RAM Transfer Bit 0: No request 1: Transfer to OBF shadow requested
8	VIEW	View Shadow Buffer Bit 0: No action 1: Switch OBF shadow and OBF host
7	—	Reserved This bit is always read as 0. When writing, always write 0.
6 to 0	OBR[6:0]	Output Buffer Request Shadow Bit

(1) FLXAnFROBCR.OBRH

Output Buffer Request Host Flag

Indicate number of message buffer currently accessible by the host via FLXAnFRRDHS1 to FLXAnFRRDHS3, FLXAnFRMBS, and FLXAnFRRDDS1 to FLXAnFRRDDS64.

By writing FLXAnFROBCR.VIEW to '1', OBF host is switched and the transferred message buffer is accessible by the host.

(2) FLXAnFROBCR.OBSYS

Output Buffer Busy Shadow Flag

Set to '1' after setting bit FLXAnFROBCR.REQ. When the transfer between the message RAM and OBF shadow has completed, FLXAnFROBCR.OBSYS is set back to '0'.

(3) FLXAnFROBCR.REQ

Request Message RAM Transfer Bit

Only writeable while FLXAnFROBCR.OBSYS = '0'.

Requests transfer of message buffer addressed by FLXAnFROBCR.OBR[6:0] from message RAM to OBF shadow.

(4) FLXAnFROBCR.VIEW

View Shadow Buffer Bit

Only writeable while FLXAnFROBCR.OBSYS = '0'.

Toggles between OBF shadow and OBF host.

(5) FLXAnFROBCR.OBR[6:0]

Output Buffer Request Shadow Bit

Only writeable while FLXAnFROBCR.OBSYS = '0'.

Indicate number of source message buffer to be transferred from the message RAM to OBF shadow.

If the number of the first message buffer of the receive FIFO is written to this register, the message buffer addressed by the GET Index (GIDX, see **Section 21.4.11, FIFO Function**) is transferred to OBF shadow.

21.3.12 Data Transfer Control Register

21.3.12.1 FLXAnFRITC — FlexRay Input Transfer Configuration Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	ITM[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	IQEIE	IQFIE	—	—	—	—	—	—	IQHR	ITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 21.82 FLXAnFRITC Register Contents

Bit Position	Bit Name	Function
31 to 23	—	Reserved These bits are always read as 0. When writing, always write 0.
22 to 16	ITM[6:0]	Input Queue Table Max Bit These bits configure the number of entries in the input pointer table the input buffer handler can maintain in the input queue.
15 to 10	—	Reserved These bits are always read as 0. When writing, always write 0.
9	IQEIE	Input Queue Empty Interrupt Enable Bit 0: Disabled 1: Enabled
8	IQFIE	Input Queue Full Interrupt Enable Bit 0: Disabled 1: Enabled
7 to 2	—	Reserved These bits are always read as 0. When writing, always write 0.
1	IQHR	Input Queue Halt Request Bit 0: Input queue run request 1: Input queue halt request
0	ITE	Input Transfer Enable Bit 0: Operation Disable request 1: Operation Enable request

(1) FLXAnFRITC.ITM

Input Queue Table Max Bit

The user can only write to this bit when FLXAnFRITC.ITE is '0'.

These bits configure the number of entries in the input pointer table the input buffer handler can maintain in the input queue.

Valid values are 00_H (1 queue entry) to 7F_H (128 queue entries).

Note that each entry requires two long words in the input pointer table.

(2) FLXAnFRITC.IQEIE

Input Queue Empty Interrupt Enable Bit

This bit controls the input queue empty interrupt.

0: Disabled

No interrupt will be requested and the input queue empty interrupt will be not generated.

1: Enabled

Input queue empty interrupt will be generated when FLXAnFRITS.IQEIS is '1'.

(3) FLXAnFRITC.IQFIE

Input Queue Full Interrupt Enable Bit

This bit controls the input queue full interrupt.

0: Disabled

No interrupt will be requested and the input queue full interrupt will be not generated.

1: Enabled

Input queue full interrupt will be generated when FLXAnFRITS.IQFIS is '1'.

(4) FLXAnFRITC.IQHR

Input Queue Halt Request Bit

The IQHR bit should not be set to '1' when FLXAnFRITS.ITS is '0'.

This bit requests a halt of the input queue.

The status of the halt request is shown in the FLXAnFRITS.IQH register.

Refer to **Section 21.4.16.1(5), Halting the Input Queue** about usage of this bit.

0: Input queue run request

The input queue resumes their operation.

1: Input queue halt request

The input queue gets halted. An active input transfer will be completed but no further transfer request will start.

(5) FLXAnFRITC.ITE

Input Transfer Enable Bit

The user should only set this bit to '1' when FLXAnFRIBCR.IBSYS is '0'.

The user should only set this bit to '0' when FLXAnFRITC.IQHR '0'. Otherwise committed input transfers get lost.

This bit controls the operation mode of the input transfer queue.

The operation status of the input transfer queue function is shown in FLXAnFRITS.ITS.

Refer to **Section 21.4.16.1(1), Activation and Deactivation** about usage of this bit.

0: Operation Disable request

The input transfer queue gets disabled when it becomes empty.

1: Operation Enable request

The input transfer queue gets enabled. Input data structures are transferred to the FlexRay internal message RAM.

21.3.12.2 FLXAnFROTC — FlexRay Output Transfer Configuration Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	FTM[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	FWIE	OWIE	FIE	OIE	—	—	—	—	—	—	OTCS	OIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 21.83 FLXAnFROTC Register Contents

Bit Position	Bit Name	Function
31 to 21	—	Reserved These bits are always read as 0. When writing, always write 0.
20 to 16	FTM[4:0]	FIFO Table Max Bit Configure the number of FIFO entries the output transfer handler can maintain in the Local RAM/Cluster RAM.
15 to 12	—	Reserved These bits are always read as 0. When writing, always write 0.
11	FWIE	FIFO Transfer Warning Interrupt Enable Bit 0: Disabled 1: Enabled
10	OWIE	Output Transfer Warning Interrupt Enable Bit 0: Disabled 1: Enabled
9	FIE	FIFO Transfer Interrupt Enable Bit 0: Disabled 1: Enabled
8	OIE	Output Transfer Interrupt Enable Bit 0: Disabled 1: Enabled
7 to 2	—	Reserved These bits are always read as 0. When writing, always write 0.
1	OTCS	Output Transfer Condition Select Bit 0: New data only mode 1: New data and status changed mode
0	OIE	Output Transfer Enable Bit 0: Operation Disable request 1: Operation Enable request

(1) FLXAnFROTC.FTM

FIFO Table Max Bit

The user can only write to these bits when FLXAnFROTS.OTS is '0'.

These bits configure the number of FIFO entries the output transfer handler can maintain in the Local RAM/Cluster RAM.

Valid values are 00_H (1 FIFO entry) to 1F_H (32 FIFO entries).

(2) FLXAnFROTC.FWIE

FIFO transfer Warning Interrupt Enable Bit

This bit controls the FIFO transfer warning interrupt.

0: Disabled

No interrupt will be requested and the FIFO transfer warning interrupt will be not generated.

1: Enabled

FIFO transfer warning interrupt will be generated when FLXAnFROTS.FWIS is '1'.

(3) FLXAnFROTC.OWIE

Output Transfer Warning Interrupt Enable Bit

This bit controls the output transfer warning interrupt.

0: Disabled

No interrupt will be requested and the output transfer warning interrupt will be not generated.

1: Enabled

Output transfer warning interrupt will be generated when FLXAnFROTS.OWIS is '1'.

(4) FLXAnFROTC.FIE

FIFO Transfer Interrupt Enable Bit

This bit controls the FIFO transfer interrupt.

0: Disabled

No interrupt will be requested and the FIFO transfer interrupt will be not generated.

1: Enabled

FIFO transfer interrupt will be generated when FLXAnFROTS.FIS is '1'.

(5) FLXAnFROTC.OIE

Output Transfer Interrupt Enable Bit

This bit controls the output transfer interrupt.

0: Disabled

No interrupt will be requested and the output transfer interrupt will be not generated.

1: Enabled

Output transfer interrupt will be generated when FLXAnFROTS.OTIS is '1'.

(6) FLXAnFROTC.OTCS

Output Transfer Condition Select Bit

The user can only write to this bit when FLXAnFROTS.OTS is '0'.

This bit controls the output transfer condition.

0: New data only mode

The ND bits in the FLXAnFRNDATi registers are used to detect a transfer condition for dedicated receive buffer

1: New data and status changed mode

The ND bits in the FLXAnFRNDATi registers and the MBC bits in the FLXAnFRMBSCi register are used to detect a transfer condition for dedicated transmit and receive buffer

(7) FLXAnFROTC.OTE

Output Transfer Enable Bit

The user should only set this bit to '1' when FLXAnFROBCR.OBSYS is '0'.

This bit controls the operation mode of the output transfer function.

The operation status of the output buffer transfer function is shown in FLXAnFROTS.OTS.

Refer to **Section 21.4.16.2(1), Activation and Deactivation** about usage of this bit.

0: Operation Disable request

The output buffer transfer gets disabled.

An active message buffer transfer will be completed but no further transfer will start.

1: Operation Enable request

The output buffer transfer gets enabled. Message buffers are transferred from the FlexRay internal message RAM to output data structures.

The user should not change the E-Ray message RAM configuration by writing to the FLXAnFRMRC register.

21.3.12.3 FLXAnFRIBA — FlexRay Input Pointer Table Base Address Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ITA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ITA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 21.84 FLXAnFRIBA Register Contents

Bit Position	Bit Name	Function
31 to 0	ITA[31:0]	Input Table Base Address Bit These bits configure the base address of the input pointer table.

(1) FLXAnFRIBA.ITA

Input Table Base Address Bit

The user can only write to these bits when FLXAnFRITS.ITS is '0'.

The address should be 32 bit aligned, thus the bits FLXAnFRIBA.ITA[1:0] are always '0'.

These bits configure the base address of the input pointer table.

The table is used for the input transfer queue transferring message buffers from the Local RAM/Cluster RAM into the FlexRay internal message RAM.

The size of the input queue is configured in FLXAnFRITC.ITM.

Note that each entry requires two long words in the input pointer table.

21.3.12.4 FLXAnFRFBA — FlexRay FIFO Pointer Table Base Address Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 21.85 FLXAnFRFBA Register Contents

Bit Position	Bit Name	Function
31 to 0	FTA[31:0]	FIFO Pointer Table Base Address Bit These bits configure the base address of the FIFO pointer table.

(1) FLXAnFRFBA.FTA

FIFO Pointer Table Base Address Bit

The user can only write to these bits when FLXAnFROTS.OTS is '0'.

The address should be 32 bit aligned, thus the bits FLXAnFRFBA.FTA[1:0] are always '0'.

These bits configure the base address of the FIFO pointer table.

The table is used for message buffers transferred from the FlexRay internal FIFO to the Local RAM/Cluster RAM.

The size of the FIFO is configured in FLXAnFROTC.FTM.

21.3.12.5 FLXAnFROBA — FlexRay Output Pointer Table Base Address Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 21.86 FLXAnFROBA Register Contents

Bit Position	Bit Name	Function
31 to 0	OTA[31:0]	Output Pointer Table Base Address Bit These bits configure the base address of the output pointer table.

(1) FLXAnFROBA.OTA

Output Pointer Table Base Address Bit

The user can only write to these bits when FLXAnFROTS.OTS is '0'.

The address should be 32 bit aligned, thus the bits FLXAnFROBA.OTA[1:0] are always '0'.

These bits configure the base address of the output pointer table.

The table is used for message buffers transferred from the FlexRay internal message RAM to the Local RAM/Cluster RAM.

The size of the table depends on the utilization of the FlexRay internal message RAM and can have up to 128 entries.

21.3.12.6 FLXAnFRIQC — FlexRay Input Queue Control Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	IMBNR[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W

Table 21.87 FLXAnFRIQC Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved These bits are always read as 0. When writing, always write 0.
6 to 0	IMBNR[6:0]	Input Message Buffer Number Bit Message buffer number added to the input queue

(1) FLXAnFRIQC.IMBNR

Input Message Buffer Number Bit

The user can only write to these bits when FLXAnFRITS.IQFP is '0'.

The user should not write to this register when FLXAnFRITS.ITS is '0' or when FLXAnFRITC.ITE is '0'.

These bits are read as "0".

This value specifies the message buffer added to the input queue.

The number has to be identical to FLXAnFRWRHS4.IMBNR (see **Section 21.4.16.1(3), Input Pointer Table**) of the input pointer table.

The address to the input data structure has to be provided in the input pointer table at the put index (FLXAnFRITS.IPIDX) before writing to this register.

Writing to this register increments the input put index (FLXAnFRITS.IPIDX).

21.3.12.7 FLXAnFRUIR — FlexRay User Input Transfer Request Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	UIDX[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.88 FLXAnFRUIR Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are always read as 0. When writing, always write 0.
7 to 0	UIDX[7:0]	User Requested Input Index Bit Input pointer table index requested for input transfer

(1) FLXAnFRUIR.UIDX

User Requested Input Index Bit

The user can only write to these bits when FLXAnFRITS.UIRP is '0'.

The user should not write to this register when FLXAnFRITS.ITS is '0'.

The user should not write to this register when FLXAnFRITS.UIRP is '1'.

The user should not write to this register when FLXAnFRITS.IQH is '1'.

The user should only write FLXAnFRITC.ITM + 1 to this register.

This value configures the user input pointer table index requested for input transfer.

The address to the input data structure has to be provided in the input pointer table at the index UIDX before writing to this register.

When writing to this register, the requested input data structure will be transferred from input data structure position to the FlexRay internal message RAM.

In opposite to queued input transfers, the related DA flag in the FLXAnFRDA register is not affected by the user input transfer.

21.3.12.8 FLXAnFRUOR — FlexRay User Output Transfer Request Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	URDS	—	—	UMBNR[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.89 FLXAnFRUOR Register Contents

Bit Position	Bit Name	Function
31 to 10	—	Reserved These bits are always read as 0. When writing, always write 0.
9	URDS	User Request Read Data Section Bit 0: Data section is not transferred 1: Data section is transferred
8, 7	—	Reserved These bits are always read as 0. When writing, always write 0.
6 to 0	UMBNR[6:0]	User Requested Output Message Buffer Number Bit Message buffer number requested for output transfer

(1) FLXAnFRUOR.URDS

User Request Read Data Section Bit

The user can only write to this bit when FLXAnFROTS.UORP is '0'.

The user should not write to this register when FLXAnFROTS.OTS is '0'.

The user should not write to this register when FLXAnFROTS.UORP is '1'.

0: Data section is not transferred

The data section of the message buffer selected by the bits UMBNR is not requested

1: Data section is transferred

The data section of the message buffer selected by the bits UMBNR is requested

(2) FLXAnFRUOR.UMBNR

User Requested output Message Buffer Number Bit

The user can only write to these bits when FLXAnFROTS.UORP is '0'.

The user should not write to this register when FLXAnFROTS.OTS is '0'.

The user should not write to this register when FLXAnFROTS.UORP is '1'.

The user should restrict this bit to dedicated receive and transmit buffers when the FlexRay module is not in the CONFIG state.

When writing to this register, the header sections and the optional data section (configurable by URDS) of the requested message buffer will be transferred from the FlexRay internal message RAM to the output data structure position defined by the output structure data pointer in the output pointer table.

21.3.13 Data Transfer Status Register

21.3.13.1 FLXAnFRITS — FlexRay Input Transfer Status Register

Do not rewrite this register using bit manipulation instructions.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	IGIDX[6:0]						—	IPIDX[6:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	IQFP	—	—	IQEIS	IQFIS	—	—	—	—	—	UIRP	IQH	ITS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Table 21.90 FLXAnFRITS Register Contents (1/2)

Bit Position	Bit Name	Function
31	—	Reserved This bit is always read as 0. When writing, always write 0.
30 to 24	IGIDX[6:0]	Input Queue Get Index Bit Represents the get index of the input pointer table
23	—	Reserved This bit is always read as 0. When writing, always write 0.
22 to 16	IPIDX[6:0]	Input Queue Put Index Bit Represents the put index of the input pointer table
15 to 13	—	Reserved These bits are always read as 0. When writing, always write 0.
12	IQFP	Input Queue Full Condition Pending Bit 0: Entries in the input queue are available 1: All entries in the input queue are occupied
11, 10	—	Reserved These bits are always read as 0. When writing, always write 0.
9	IQEIS	Input Queue Empty Interrupt Status Bit 0: No input queue empty condition detected 1: Input queue empty condition detected
8	IQFIS	Input Queue Full Interrupt Status Bit 0: No input queue full condition detected 1: Input queue full condition detected
7 to 3	—	Reserved These bits are always read as 0. When writing, always write 0.
2	UIRP	User Input Transfer Request Pending Bit 0: No user input transfer request pending 1: User input transfer request pending

Table 21.90 FLXAnFRITS Register Contents (2/2)

Bit Position	Bit Name	Function
1	IQH	Input Queue Halted Bit 0: Input queue not halted 1: Input queue halted
0	ITS	Input Transfer Status Bit 0: Disabled 1: Enabled

(1) FLXAnFRITS.IGIDX

Input Queue Get Index Bit

These bits are only valid when FLXAnFRITS.IQH is '1'

These bits represent the input pointer index the input queue handler will transfer next.

Valid values are 00_H to FLXAnFRITC.ITM.

The get index is incremented when the input data structure has been transferred from the Local RAM/Cluster RAM and the related DA flag in the FLXAnFRDA register is cleared.

The index is set to 00_H when FLXAnFRITS.ITS changes from '0' to '1'.

(2) FLXAnFRITS.IPIDX

Input Queue Put Index Bit

These bits represent the index where the next input data structure pointer in the input pointer table should be stored.

Valid values are 00_H to FLXAnFRITC.ITM.

After reaching the maximum value, the put index continues from 00_H.

The index is incremented when writing to FLXAnFRIQC.IMBNR.

The index is set to 00_H when FLXAnFRITS.ITS changes from '0' to '1'.

(3) FLXAnFRITS.IQFP

Input Queue Full Condition Pending Bit

This bit represents that the input queue is full.

There should be no further input transfer requests, by writing to FLXAnFRIQC.IMBNR, as long as FLXAnFRITS.IQFP is '1'.

[Clearing condition]

This bit is cleared by there is one free entry in the input queue.

[Setting condition]

This bit is set when all entries in the input queue are occupied.

(4) FLXAnFRITS.IQEIS

Input Queue Empty Interrupt Status Bit

Writing '0' has no effect on the bit value.

If enabled in FLXAnFRITC.IQEIE the input queue empty interrupt is generated when FLXAnFRITS.IQEIS is '1'.

[Clearing condition]

This bit is cleared by writing '1' to FLXAnFRITS.IQEIS.

This bit is cleared when FLXAnFRITS.ITS changes from '0' to '1'.

[Setting condition]

This bit is set when all pending input transfers have been processed and consequently the input queue becomes empty.

(5) FLXAnFRITS.IQFIS

Input Queue Full Interrupt Status Bit

Writing '0' has no effect on the bit value.

If enabled in FLXAnFRITC.IQFIE, the input queue full interrupt is generated when FLXAnFRITS.IQFIS is '1'.

This flag is intended as an interrupt status flag. It does not represent the current input queue status; for this status, refer to FLXAnFRITS.IQFP.

[Clearing condition]

This bit is cleared by writing '1' to FLXAnFRITS.IQFIS.

This bit is cleared when FLXAnFRITS.ITS changes from '0' to '1'.

[Setting condition]

This bit is set when all entries in the input queue are occupied.

(6) FLXAnFRITS.UIRP

User Input Transfer Request Pending Bit

This bit represents that the user input transfer is still pending.

There should be no further write access to FLXAnFRUIR.UIDX when this bit is '1'.

[Clearing condition]

This bit is cleared when the user input transfer request is processed by the input transfer handler.

[Setting condition]

This bit is set by writing to FLXAnFRUIR.UIDX.

(7) FLXAnFRITS.IQH

Input Queue Halted Bit

This bit represents the status of the input queue.

There should be no further write access to FLXAnFRUIR.UIDX when this bit is '1'.

[Clearing condition]

This bit is cleared when FLXAnFRITC.IQHR is set to '0'.

[Setting condition]

This bit is set immediately when FLXAnFRITC.IQHR is set to '1' and there is no ongoing input transfer.

This bit is set only after an ongoing input transfer has been completed and FLXAnFRITC.IQHR is set to '1'.

(8) FLXAnFRITS.ITS

Input Transfer Status Bit

This bit represents the status of the input queue handler.

While this bit is '1', there should be no read or write access to the address area <FLXAn_base> + 0400_H to <FLXAn_base> + 05FF_H and there should be no CLEAR_RAM command applied to FLXAnFRSUCC1.CMD.

The input transfer queue indexes and related status flags are set to '0' when FLXAnFRITS.ITS changes from '0' to '1'.

[Clearing condition]

This bit is cleared immediately when FLXAnFRITC.ITE is set to '0' and there are no pending input transfers.

This bit is cleared after all pending requests have been processed and FLXAnFRITC.ITE is '0'.

[Setting condition]

This bit is set when FLXAnFRITC.ITE is set to '1'.

21.3.13.2 FLXAnFROTS — FlexRay Output Transfer Status Register

Do not rewrite this register using bit manipulation instructions.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	FFL[5:0]					—	—	—	FGIDX[4:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FWP	OWP	FDA	—	FWS	OWS	FIS	OTIS	—	—	—	—	—	UORP	—	OTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 21.91 FLXAnFROTS Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	—	Reserved These bits are always read as 0. When writing, always write 0.
29 to 24	FFL[5:0]	FIFO Fill Level Bit Represent the number of unprocessed output FIFO structures
23 to 21	—	Reserved These bits are always read as 0. When writing, always write 0.
20 to 16	FGIDX[4:0]	FIFO Get Index Bit Represent the get index in the FIFO pointer table
15	FWP	FIFO Transfer Warning Condition Pending Bit 0: No FIFO transfer warning condition pending 1: FIFO transfer warning condition pending
14	OWP	Output Transfer Warning Condition Pending Bit 0: No output transfer warning condition pending 1: Output transfer warning condition pending
13	FDA	FIFO Data Available Bit 0: No available FIFO structures 1: FIFO structures available at current FLXAnFROTS.FGIDX index
12	—	Reserved This bit is always read as 0. When writing, always write 0.
11	FWIS	FIFO Transfer Warning Interrupt Status Bit 0: No FIFO transfer warning condition detected 1: FIFO transfer warning condition detected
10	OWIS	Output Transfer Warning Interrupt Status Bit 0: No output transfer warning condition detected 1: Output transfer warning condition detected
9	FIS	FIFO Transfer Interrupt Status Bit 0: No FIFO structure updated in Local RAM/Cluster RAM 1: FIFO structure updated in Local RAM/Cluster RAM

Table 21.91 FLXAnFROTS Register Contents (2/2)

Bit Position	Bit Name	Function
8	OTIS	Output Transfer Interrupt Status Bit 0: No output structure updated in Local RAM/Cluster RAM 1: Output structure updated in Local RAM/Cluster RAM
7 to 3	—	Reserved These bits are always read as 0. When writing, always write 0.
2	UORP	User Output Transfer Request Pending Bit 0: No user output transfer request pending 1: User output transfer request pending
1	—	Reserved This bit is always read as 0. When writing, always write 0.
0	OTS	Output Transfer Status Bit 0: Disabled 1: Enabled

(1) FLXAnFROTS.FFL

FIFO Fill Level Bit

These bits represent the number of available output FIFO structures in the Local RAM/Cluster RAM.

Valid values are 00_H to FLXAnFROTC.FTM + 1.

The value 00_H represents that the FIFO is empty.

The value FLXAnFROTC.FTM + 1 represents that the FIFO is full and no further FIFO transfers will be done.

These bits are incremented when a FIFO data structure has been transferred from the FlexRay internal FIFO into the Local RAM/Cluster RAM.

The FIFO fill level is decremented when the user releases a FIFO data structure in the Local RAM/Cluster RAM by writing '1' to FLXAnFROTS.FDA.

The FIFO fill level is set to 00_H when bit FLXAnFROTS.OTS changes from '0' to '1'.

(2) FLXAnFROTS.FGIDX

FIFO Get Index Bit

These bits represent the index where the current output data structure pointer in the FIFO pointer table is available for reading.

Valid values are 00_H to FLXAnFROTC.FTM.

After reaching the maximum value, the get index continues from 00_H.

The index is incremented when a FIFO data structure is released by writing '1' to FLXAnFROTS.FDA.

The index is set to 00_H when FLXAnFROTS.OTS changes from '0' to '1'.

(3) FLXAnFROTS.FWP

FIFO Transfer Warning Condition Pending Bit

This bit represents the FIFO transfer warning condition.

[Clearing condition]

This bit is cleared when there are free output data structures ($FLXAnFROTS.FFL \leq FLXAnFROTC.FTM$).

This bit is cleared when FLXAnFROTS.OTS changes from '0' to '1'.

[Setting condition]

This bit is set when the output transfer handler detects a transfer condition for FIFO message buffers but there are no free output data structures ($FLXAnFROTS.FFL = FLXAnFROTC.FTM + 1$).

(4) FLXAnFROTS.OWP

Output Transfer Warning Condition Pending Bit

This bit represents the output transfer warning condition.

[Clearing condition]

This bit is cleared when all output structure pointers that have a pending output handler transfer condition detected, are released (for dedicated transmit and receive message buffers or the user output transfer request).

[Setting condition]

This bit is set when the output transfer handler detects a transfer condition (for dedicated transmit and receive message buffers or the user output transfer request) but the related output structure pointer was not yet released by the application (data available flag is still set to '1').

This bit is set when the output transfer handler detects a transfer condition for dedicated transmit and receive message buffers but there is a pending input transfer for the same message buffer (data available flag is set to '1' due to the input transfer request).

(5) FLXAnFROTS.FDA

FIFO Data Available Bit

Writing '0' has no effect on the bit value.

When this bit is '1', the next valid output data structure is available.

The related data structure pointer is in the FIFO pointer table at FLXAnFROTS.FGIDX.

Writing '1' to FLXAnFROTS.FDA

- Increments FLXAnFROTS.FGIDX and
- Decrements the FIFO fill level (FLXAnFROTS.FFL)

If there are still unprocessed data structures, FLXAnFROTS.FDA remains '1'.

[Clearing condition]

This bit is cleared by writing '1' to FLXAnFROTS.FDA and the FIFO fill level becomes 00_H.

This bit is cleared when the FLXAnFROTS.OTS changes from '0' to '1'.

[Setting condition]

This bit is set when there is at least one FIFO data structure available in the Local RAM/Cluster RAM.

(6) FLXAnFROTS.FWIS

FIFO Transfer Warning Interrupt Status Bit

Writing '0' has no effect on the bit value.

If the FLXAnFROTC.FWIE is enabled, the FIFO transfer warning interrupt is generated when FLXAnFROTS.FWIS is '1'.

[Clearing condition]

This bit is cleared by writing '1' to FLXAnFROTS.FWIS.

This bit is cleared when bit FLXAnFROTS.OTS changes from '0' to '1'.

[Setting condition]

This bit is set when the output transfer handler detects a transfer condition for FIFO message buffers but there are no free output data structures (FLXAnFROTS.FFL = FLXAnFROTC.FTM + 1).

(7) FLXAnFROTS.OWIS

Output Transfer Warning Interrupt Status Bit

Writing '0' has no effect on the bit value.

If enabled in FLXAnFROTC.OWIE, the FIFO transfer warning interrupt is generated when FLXAnFROTS.OWIS is '1'.

[Clearing condition]

This bit is cleared by writing '1' to FLXAnFROTS.OWIS.

This bit is cleared when bit FLXAnFROTS.OTS changes from '0' to '1'.

[Setting condition]

This bit is set when the output transfer handler detects a transfer condition (for dedicated transmit and receive message buffers or the user output transfer request) but the related output structure pointer was not yet released by the application (data available flag is still set to '1').

This bit is set when the output transfer handler detects a transfer condition for dedicated transmit and receive message buffers but there is a pending input transfer for the same message buffer (data available flag is set to '1' due to the input transfer request).

(8) FLXAnFROTS.FIS

FIFO Transfer Interrupt Status Bit

Writing '0' has no effect on the bit value.

If enabled in FLXAnFROTC.FIE the FIFO transfer interrupt is generated when FLXAnFROTS.FIS is '1'.

[Clearing condition]

This bit is cleared by writing '1' to FLXAnFROTS.FIS.

This bit is cleared when bit FLXAnFROTS.OTS changes from '0' to '1'.

[Setting condition]

This bit is set when a FIFO data structure is updated by the transfer handler or the FFL bit changes from 00_H to 01_H.

(9) FLXAnFROTS.OTIS

Output Transfer Interrupt Status Bit

Writing '0' has no effect on the bit value.

If enabled in FLXAnFROTC.OIE, the output transfer interrupt is generated when FLXAnFROTS.OTIS is '1'.

[Clearing condition]

This bit is cleared by writing '1' to FLXAnFROTS.OTIS.

This bit is cleared when bit FLXAnFROTS.OTS changes from '0' to '1'.

[Setting condition]

This bit is set when an output data structure is updated by the transfer handler (from a dedicated transmit or receive message buffer or by the user output transfer request).

(10) FLXAnFROTS.UORP

User Output Transfer Request Pending Bit

This bit represents that the user output transfer is still pending.

There should be no further write access to FLXAnFRUOR.UMBNR when this bit is '1'.

[Clearing condition]

This bit is cleared when the user output transfer request is processed by the output transfer handler.

This bit is cleared when bit FLXAnFROTS.OTS changes from '0' to '1'.

[Setting condition]

This bit is set by writing to FLXAnFRUOR.UMBNR.

(11) FLXAnFROTS.OTS

Output Transfer Status Bit

This bit represents the status of the output transfer handler.

While this bit is '1', no read or write access to the address area <FLXAn_base> + 0600_H to <FLXAn_base> + 07FF_H cannot be made and no CLEAR_RAM command to the FLXAnFRSUCC1.CMD cannot be provided.

While this bit is '1', the user cannot change the E-Ray message RAM configuration by writing to the FLXAnFRMRC register.

The output handler transfer indexes and related status flags are set to '0' when FLXAnFROTS.OTS changes from '0' to '1'.

[Clearing condition]

This bit is cleared immediately when FLXAnFROTC.OTE is set to '0' and there are no ongoing output transfers.

This bit is cleared after an ongoing transfer has been completed and FLXAnFROTC.OTE is '0'.

[Setting condition]

This bit is set when FLXAnFROTC.OTE is set to '1'.

21.3.13.3 FLXAnFRAES — FlexRay Access Error Status Register

Do not rewrite this register using bit manipulation instructions.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MAE	FAE	OAE	IAE	EIDX[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 21.92 FLXAnFRAES Register Contents

Bit Position	Bit Name	Function
31 to 12	—	Reserved These bits are always read as 0. When writing, always write 0.
11	MAE	Multiple Access Errors Bit 0: No multiple access errors occurred 1: Multiple access errors occurred
10	FAE	FIFO Transfer Access Error Bit 0: No access error occurred during FIFO transfer 1: Access error occurred during FIFO transfer
9	OAE	Output Transfer Access Error Bit 0: No access error occurred during output transfer 1: Access error occurred during output transfer
8	IAE	Input Transfer Access Error Bit 0: No access error occurred during input transfer 1: Access error occurred during input transfer
7 to 0	EIDX[7:0]	Error Index Bit Data structure pointer index number

(1) FLXAnFRAES.MAE

Multiple Access Errors Bit

Writing '0' has no effect on the bit value.

This bit represents that there were multiple access errors during a data transfer.

[Clearing condition]

This bit is cleared by writing '1' to FLXAnFRAES.MAE.

[Setting condition]

This bit is set when one of bits FLXAnFRAES.FAE, FLXAnFRAES.OAE or FLXAnFRAES.IAE is set and

- an access to an protected address occurred during a FIFO data transfer or
- an access to an protected address occurred during an output data transfer or
- an access to an protected address occurred during an input data transfer

(2) FLXAnFRAES.FAE

FIFO Transfer Access Error Bit

Writing '0' has no effect on the bit value.

This bit indicates that there was an access error during a FIFO data transfer.

[Clearing condition]

This bit is cleared by writing '1' to FLXAnFRAES.FAE.

[Setting condition]

This bit is set when a Local RAM/Cluster RAM access error was detected during a FIFO transfer and the bits FLXAnFRAES.OAE, FLXAnFRAES.IAE and FLXAnFRAES.MAE are '0'.

(3) FLXAnFRAES.OAE

Output Transfer Access Error Bit

Writing '0' has no effect on the bit value.

This bit indicates that there was an access error during an output data transfer.

[Clearing condition]

This bit is cleared by writing '1' to FLXAnFRAES.OAE.

[Setting condition]

This bit is set when a Local RAM/Cluster RAM access error was detected during an output transfer and the bits FLXAnFRAES.FAE, FLXAnFRAES.IAE and FLXAnFRAES.MAE are '0'.

(4) FLXAnFRAES.IAE

Input Transfer Access Error Bit

Writing '0' has no effect on the bit value.

This bit indicates that there was an access error during an input data transfer.

[Clearing condition]

This bit is cleared by writing '1' to FLXAnFRAES.IAE.

[Setting condition]

This bit is set when a Local RAM/Cluster RAM access error was detected during an output transfer and the bits FLXAnFRAES.OAE, FLXAnFRAES.FAE and FLXAnFRAES.MAE are '0'.

(5) FLXAnFRAES.EIDX

Error Index Bit

This value is only valid when one of bits FLXAnFRAES.FAE, FLXAnFRAES.OAE or FLXAnFRAES.IAE is '1'.

When bit FLXAnFRAES.FAE is '1', FLXAnFRAES.EIDX holds the used FIFO put index when an access error has occurred.

When bit FLXAnFRAES.OAE is '1', FLXAnFRAES.EIDX holds the used output table entry (which is related to message buffer number) when an access error has occurred.

When bit FLXAnFRAES.IAE is '1', FLXAnFRAES.EIDX holds the used input pointer table get index when an access error has occurred during the input transfer or user requested input transfer.

These bits are updated when one of bits FLXAnFRAES.FAE, FLXAnFRAES.OAE or FLXAnFRAES.IAE changes from '0' to '1'.

21.3.13.4 FLXAnFRAEA — FlexRay Access Error Address Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AEA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.93 FLXAnFRAEA Register Contents

Bit Position	Bit Name	Function
31 to 0	AEA[31:0]	Access Error Address Bit Address in the Local RAM/Cluster RAM when an access error has occurred

(1) FLXAnFRAEA.AEA

Access Error Address Bit

This value is only valid when one of bits FLXAnFRAES.FAE, FLXAnFRAES.OAE or FLXAnFRAES.IAE is '1'.

These bits represent the address of the access error indicated in the FLXAnFRAES register.

These bits are updated when one of bits FLXAnFRAES.FAE, FLXAnFRAES.OAE or FLXAnFRAES.IAE changes from '0' to '1'.

21.3.13.5 FLXAnFRDAi — FlexRay Message Data Available Register i (i = 0 to 3)

Do not rewrite this register using bit manipulation instructions.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DAb[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DAb[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.94 FLXAnFRDAi Register Contents

Bit Position	Bit Name	Function
31 to 0	Dab[31:0]	Data Available b Bit 0: No data available for destination 1: Data available for destination

(1) FLXAnFRDAi.DAb (b = i × 32 to ((i + 1) × 32)–1)

Data Available b Bit

The user should not write '1' to bits that are '0'.

To maintain the status of input transfers, the user should not clear bits related to input transfers.

This register is used for input and output transfers.

Each flag corresponds to a FlexRay message buffer.

[Clearing condition]

- Input transfer:

This bit is cleared when the input data structure has been transferred from the Local RAM/Cluster RAM. The data structure and the data structure pointer can be changed when the related flag is '0'.

- Output transfer:

This bit is cleared by writing '1' to it.

[Setting condition]

- Input transfer:

This bit is set when the corresponding message buffer number has been written to FLXAnFRIQC.IMBNR.

As long as this bit is '1', the input data structure and the data structure pointer corresponding to this input transfer request should not be changed.

- Output transfer:

This bit is set when the output data structure corresponding to this message buffer has been updated.

As long as this bit is '1', the data structure is stable; no further update of the data structure will be done by the output handler. While this bit is '1', the application can change the output data structure pointer in the output pointer table for this message buffer number.

21.4 Operation

This chapter describes the FlexRay implementation together with the related FlexRay protocol features. More information about the FlexRay protocol itself can be found in the FlexRay protocol specification.

21.4.1 FlexRay Module Operation Control

21.4.1.1 FlexRay Module Enable

After hardware reset or after the FlexRay module has been disabled (following **Section 21.4.1.2, FlexRay Module Disable**) the FlexRay module is in the reset state (FLXAnFROS.OS is '0') and the clocks of the FlexRay core module are disabled.

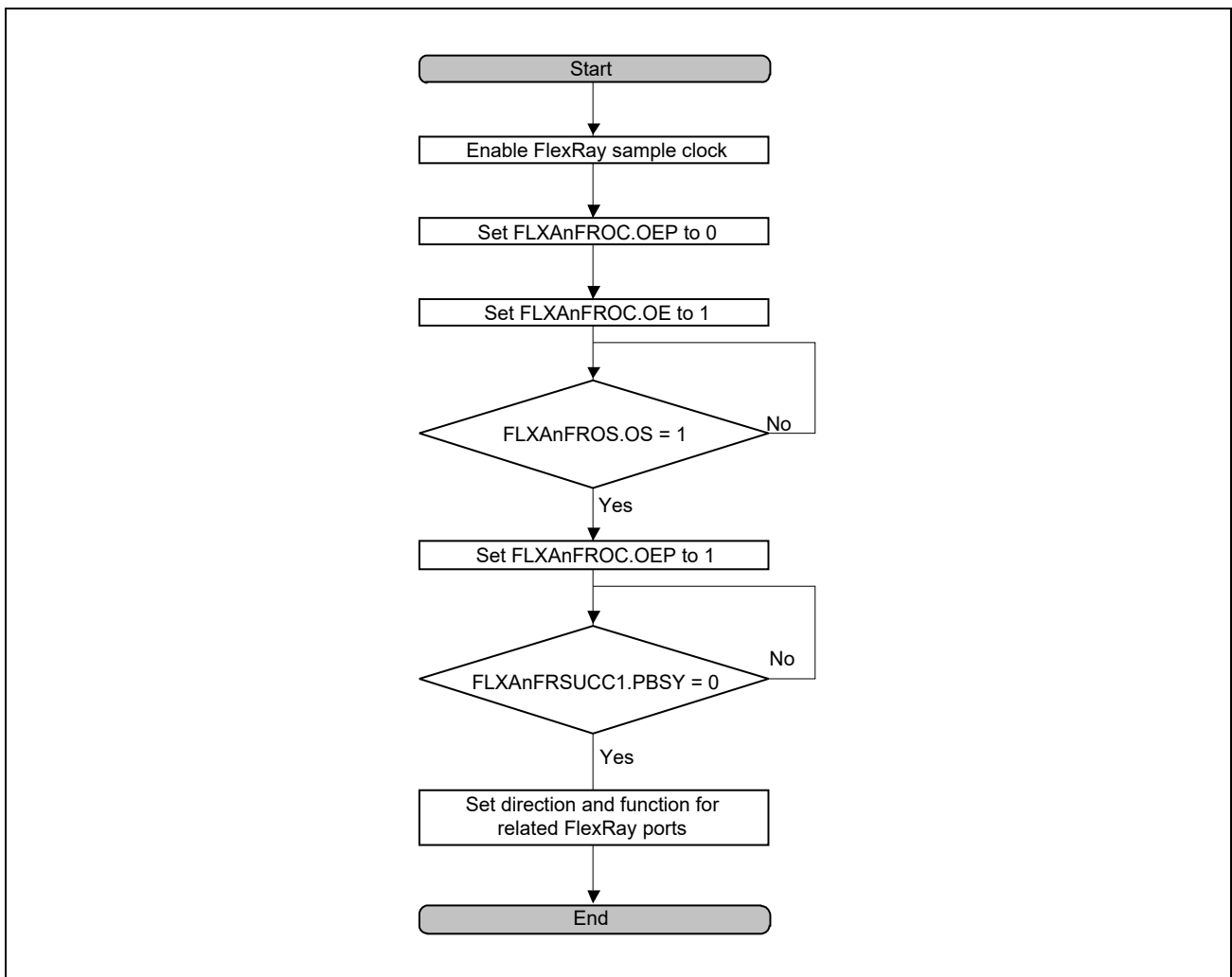


Figure 21.3 FlexRay Enable Flow

21.4.1.2 FlexRay Module Disable

The FlexRay module can be disabled at any time. However, it is recommended to disable the FlexRay module using the FLXAnFROC.OE register only when the FlexRay module is in HALT, CONFIG or DEFAULT_CONFIG state. Resetting the FlexRay module in any other state will terminate any ongoing FlexRay communication.

If the data transfer function is used, it is also required to disable this function before disabling the FlexRay module (see **Section 21.4.16.1(1), Activation and Deactivation** for suspending input transfer function and **Section 21.4.16.2(2), Activation and Deactivation** for suspending output transfer).

The following flow should be executed to disable the FlexRay module.

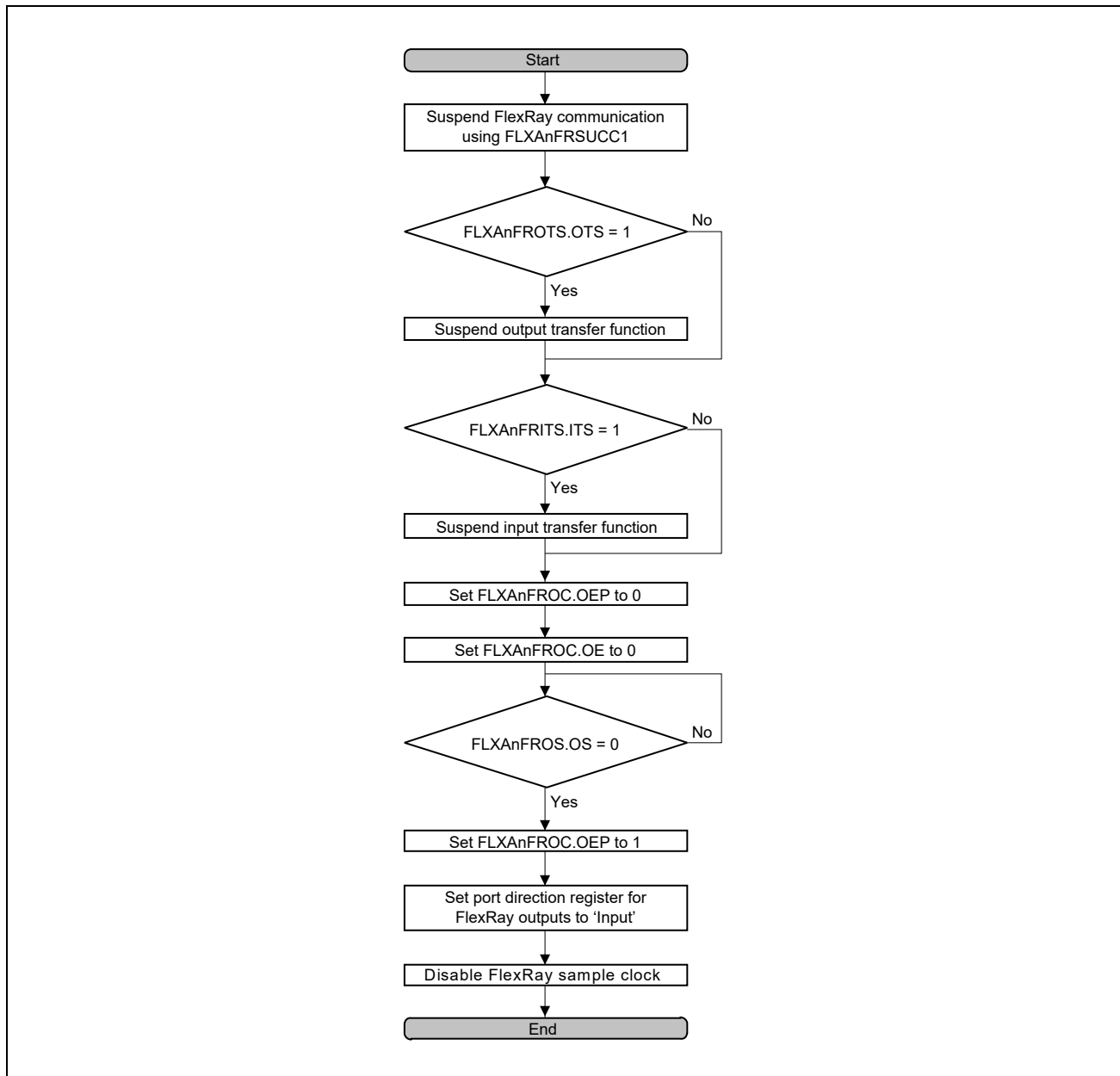


Figure 21.4 Flexray Disable Flow

21.4.2 Communication Cycle

Communication on FlexRay networks is based on frames and symbols. The wakeup symbol (WUS) and the collision avoidance symbol (CAS) are transmitted outside the communication cycle to set up the time schedule. Frames and media access test symbols (MTS) are transmitted inside the communication cycle.

A FlexRay communication cycle consists of the following four elements:

- Static Segment
- Dynamic Segment (optional)
- Symbol Window (optional)
- Network Idle Time (NIT)

Static segment, dynamic segment, and symbol window form the Network Communication Time (NCT). For each communication channel, the slot counter starts at 1 and counts up until the end of the dynamic segment is reached. Both channels share the same arbitration grid, which means that they use the same synchronized macrotick.

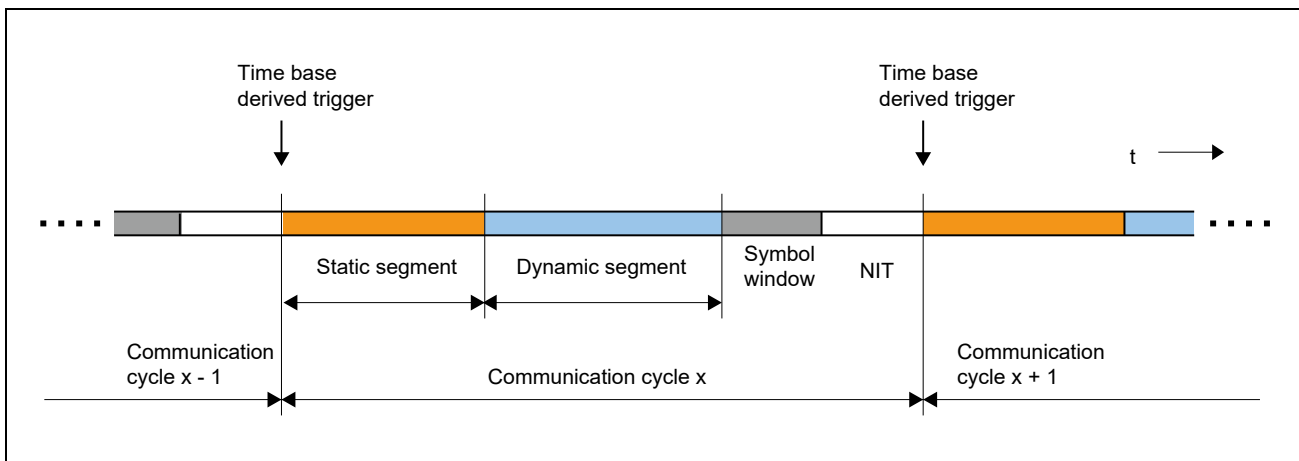


Figure 21.5 Structure of Communication Cycle

(1) Static Segment

The Static Segment is characterized by the following features:

- Time slots of fixed length (optionally protected by bus guardian)
- Start of frame transmission at action point of the respective static slot
- Payload length same for all frames on both channels

<Parameters>

Number of Static Slots (FLXAnFRGTUC7.NSS)

Static Slot Length (FLXAnFRGTUC7.SSL)

Payload Length Static (FLXAnFRMHDC.SFDL)

Action Point Offset (FLXAnFRGTUC9.APO)

(2) Dynamic Segment

The Dynamic Segment is characterized by the following features:

- All controllers have bus access (no bus guardian protection possible)
- Variable payload length and duration of slots, different for both channels
- Start of transmission at minislot action point

<Parameters>

Number of Minislots (FLXAnFRGTUC8.NMS)

Minislot Length (FLXAnFRGTUC8.MSL)

Minislot Action Point Offset (FLXAnFRGTUC9.MAPO)

Start of Latest Transmit (last minislot) (FLXAnFRMHDC.SLT)

(3) Symbol Window

During the symbol window only one media access test symbol (MTS) may be transmitted per channel. MTS symbols are sent in NORMAL_ACTIVE state to test the bus guardian.

The symbol window is characterized by the following features:

- Send single symbol
- Transmission of the MTS symbol starts at the symbol windows action point

<Parameters>

Symbol Window Action Point Offset (FLXAnFRGTUC9.APO) (same as for static slots)

Network Idle Time Start (FLXAnFRGTUC4.NIT)

(4) Network Idle Time (NIT)

During network idle time, the CC has to perform the following tasks:

- Calculate clock correction terms (offset and rate)
- Distribute offset correction over multiple macroticks after offset correction start
- Perform cluster cycle related tasks

<Parameters>

Network Idle Time Start Position Setting (FLXAnFRGTUC4.NIT)

Offset Correction Start Position Setting (FLXAnFRGTUC4.OCS)

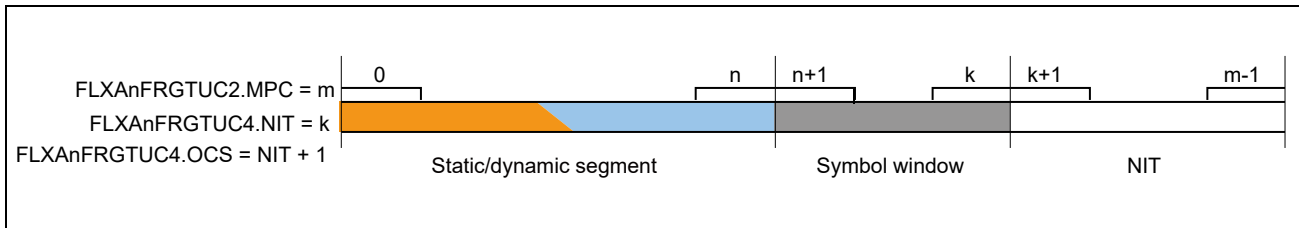
(5) Configuration of NIT Start and Offset Correction Start

Figure 21.6 Configuration of NIT Start and Offset Correction Start

The number of macroticks per cycle $gMacroPerCycle$ is assumed to be m . It is configured by programming $FLXAnFRGTUC2.MPC = m$.

The static/dynamic segment starts with macrotick 0 and ends with macrotick n :

$$n = \text{static segment length} + \text{dynamic segment offset} + \text{dynamic segment length} - 1\text{MT}$$

$$n = gNumberOfStaticSlots \cdot gdStaticSlot + \text{dynamic segment offset} + gNumberOfMinislots \cdot gdMinislot - 1\text{MT}$$

The static segment length is configured by $FLXAnFRGTUC7.SSL$ and $FLXAnFRGTUC7.NSS$.

The dynamic segment length is configured by $FLXAnFRGTUC8.MSL$ and $FLXAnFRGTUC8.NMS$.

The dynamic segment offset is:

If $gdActionPointOffset \leq gdMinislotActionPointOffset$:

dynamic segment offset = 0 MT

Else if $gdActionPointOffset > gdMinislotActionPointOffset$:

dynamic segment offset = $gdActionPointOffset - gdMinislotActionPointOffset$

The NIT starts with macrotick $k+1$ and ends with the last macrotick of cycle $m-1$. It has to be configured by setting $FLXAnFRGTUC4.NIT = k$.

For the FlexRay module, the offset correction start is required to be $FLXAnFRGTUC4.OCS \geq FLXAnFRGTUC4.NIT + 1 = k+1$.

The length of the symbol window results from the number of macroticks between the end of the static/dynamic segment and the beginning of the NIT. It can be calculated by the number of macroticks $(k - n)$.

21.4.3 Communication Modes

The FlexRay Protocol Specification defines the Time-Triggered Distributed (TT-D) mode.

21.4.3.1 Time-triggered Distributed (TT-D)

In TT-D mode, the following configurations are possible:

- Pure static: Minimum 2 or more static slots + symbol window (optional)
- Mixed static/dynamic: Minimum 2 or more static slots + dynamic segment + symbol window (optional)

A minimum of two coldstart nodes needs to be configured for distributed time-triggered operation. Two fault-free coldstart nodes are necessary for the cluster startup. Each startup frame must be a sync frame, therefore all coldstart nodes are sync nodes.

21.4.4 Clock Synchronization

In TT-D mode, a distributed clock synchronization is used. Each node individually synchronizes itself to the cluster by observing the timing of received sync frames from other nodes.

21.4.4.1 Global Time

Activities in a FlexRay node, including communication, are based on the concept of a global time. It is the clock synchronization mechanism that differentiates the FlexRay cluster from other node collections with independent clock mechanisms. The global time is a vector of two values; the cycle (cycle counter) and the cycle time (macrotick counter).

Cluster specific:

- Macrotime (MT) = basic unit of time measurement in a FlexRay network, a macrotime consists of an integer number of microtimes (μT)
- Cycle length = duration of a communication cycle in units of macrotimes (MT)

21.4.4.2 Local Time

Internally, nodes time their behavior with microtick resolution. Microticks are time units derived from the oscillator clock tick of the specific node. Therefore, microticks are controller-specific units. They may have different durations in different controllers. The precision of a node's local time difference measurements is a microtick (μT).

Node specific:

- Oscillator clock \rightarrow prescaler \rightarrow microtick (μT)
- μT = basic unit of time measurement in a CC, clock correction is done in units of μTs
- Cycle counter + macrotime counter = node's local view of the global time

21.4.4.3 Synchronization Process

Clock synchronization is performed by means of sync frames. Only preconfigured nodes (sync nodes) are allowed to send sync frames. In a two-channel cluster, a sync node has to send its sync frame on both channels.

For synchronization in FlexRay, the following constraints have to be considered:

- Max. of one sync frame per node in one communication cycle
- Max. of 15 sync frames per cluster in one communication cycle
- Every node has to use a preconfigured number of sync frames (FLXAnFRGTUC2.SNM) for clock synchronization
- Minimum of two sync nodes required for clock synchronization and startup

For clock synchronization, the time difference between expected and observed arrival time of sync frames received during the static segment is measured. In a two channel cluster, the sync node has to be configured to send sync frames on both channels. The calculation of correction terms is done during NIT (offset: every cycle, rate: every odd cycle) by using an FTM algorithm. For details, see FlexRay protocol specification, section 8.

(1) Offset (phase) Correction

- Only deviation values measured and stored in the current cycle used
- For a two channel node, the smaller value will be taken
- Calculation during NIT of every communication cycle
- Offset correction value calculated in even cycles used for error checking only
- Checked against limit values
- Correction value is a signed integer number of μ Ts
- Correction done in odd numbered cycles, distributed over the macroticks beginning at offset correction startup to cycle end (end of NIT) to shift nodes next start of cycle (MTs lengthened or shortened)

(2) Rate (frequency) Correction

- Pairs of deviation values measured and stored in even/odd cycle pair used
- For a two channel node, the average of the differences from the two channels is used
- Calculated during NIT of odd numbered cycles
- Cluster drift damping is performed using global damping value
- Checked against limit values
- Correction value is a signed integer number of μ Ts
- Distributed over macroticks comprising the next even/odd cycle pair (MTs lengthened or shortened)

(3) Sync Frame Transmission

Sync frame transmission is only possible from buffers 0 and 1. Message buffer 1 may be used for sync frame transmission in case that sync frames should have different payloads on the two channels. In this case, bit FLXAnFRMRC.SPLM has to be programmed to '1'.

Message buffers used for sync frame transmission have to be configured with the key slot ID and can be (re)configured in DEFAULT_CONFIG or CONFIG state only. For nodes transmitting sync frames FLXAnFRSUCC1.TXSY must be set to '1'.

(4) External Clock Synchronization

During normal operation, independent clusters can drift significantly. If synchronous operation across independent clusters is desired, external synchronization is necessary, even though the nodes within each cluster are synchronized. This can be accomplished with synchronous application of host-deduced rate and offset correction terms to the clusters.

- External offset/rate correction value is a signed integer
- External offset/rate correction value is added to calculated offset/rate correction value
- Aggregated offset/rate correction term (external + internal) is not checked against configured limits

21.4.5 Error Handling

The implemented error handling concept is intended to ensure that, in case of a lower layer protocol error in a single node, communication between non-affected nodes can be maintained. In some cases, higher layer program activity is required for the CC to resume normal operation. A change of the error handling state will set FLXAnFREIR.PEMC to '1' and may trigger an interrupt to the host if enabled. The actual error mode is signaled by FLXAnFRCCEV.ERRM.

Table 21.95 Error Modes of the POC (Degradation Model)

Error Mode	Activity
ACTIVE	Full operation, State: NORMAL_ACTIVE The CC is fully synchronized and supports the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers FLXAnFREIR and FLXAnFRSIR.
PASSIVE	Reduced operation, State: NORMAL_PASSIVE, CC self rescue allowed The CC stops transmitting frames and symbols, but received frames are still processed. Clock synchronization mechanisms are continued based on received frames. No active contribution to the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers FLXAnFREIR and FLXAnFRSIR.
COMM_HALT	Operation halted, State: HALT, CC self rescue not allowed The CC stops frame and symbol processing, clock synchronization processing, and the macrotick generation. The host has still access to error and status information by reading the error and status interrupt flags from registers FLXAnFREIR and FLXAnFRSIR. The bus drivers are disabled.

21.4.5.1 Clock Correction Failed Counter

When the Clock Correction Failed Counter reaches the “maximum without clock correction passive” limit defined by FLXAnFRSUCC3.WCP, the POC transits from NORMAL_ACTIVE to NORMAL_PASSIVE state. When it reaches the “maximum without clock correction fatal” limit defined by FLXAnFRSUCC3.WCF, it transits from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state.

The Clock Correction Failed Counter FLXAnFRCCEV.CCFC allows the host to monitor the duration of the inability of a node to compute clock correction terms after the CC passes the protocol startup phase. The counter will be incremented by one at the end of any odd communication cycle during which either the missing offset correction FLXAnFRSFS.MOCS or the missing rate correction FLXAnFRSFS.MRCS flag is set to '1'.

The Clock Correction Failed Counter is reset to zero at the end of an odd communication cycle if neither the missing offset correction FLXAnFRSFS.MOCS nor the missing rate correction FLXAnFRSFS.MRCS flag is set to '1'.

The Clock Correction Failed Counter stops incrementing when the “maximum without clock correction fatal” value FLXAnFRSUCC3.WCF is reached (i.e. incrementing the counter at its maximum value will not cause it to wrap around back to zero). The Clock Correction Failed Counter is initialized to zero when the CC enters READY state or when NORMAL_ACTIVE state is entered.

CAUTION

The transition to HALT state is prevented if FLXAnFRSUCC1.HCSE is not set to '1'.

21.4.5.2 Passive to Active Counter

The passive to active counter controls the transition of the POC from NORMAL_PASSIVE to NORMAL_ACTIVE state. FLXAnFRSUCC1.PTA defines the number of consecutive even/odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state. If FLXAnFRSUCC1.PTA is set to zero, the CC is not allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state.

21.4.5.3 HALT Command

In case the host wants to stop FlexRay communication of the local node, it can bring the CC into HALT state by asserting the HALT command. This can be done by writing FLXAnFRSUCC1.CMD = "0110". In order to shut down communication on an entire FlexRay network, a higher layer protocol is required to assure that all nodes apply the HALT command at the same time.

The POC state from which the transition to HALT state took place can be read from FLXAnFRCCSV.PSL.

When called in NORMAL_ACTIVE or NORMAL_PASSIVE state, the POC transits to HALT state at the end of the current cycle. When called in any other state, FLXAnFRSUCC1.CMD will be reset to "0000" = command_not_accepted and bit FLXAnFREIR.CNA is set to '1'. If enabled, an interrupt to the host is generated.

21.4.5.4 FREEZE Command

In case the host detects a severe error condition, it can bring the CC into HALT state by asserting the FREEZE command. This can be done by writing FLXAnFRSUCC1.CMD = "0111". The FREEZE command triggers the entry of the HALT state immediately regardless of the actual POC state.

The POC state from which the transition to HALT state took place can be read from FLXAnFRCCSV.PSL.

CAUTION

When a terminal restarts transfer as a leading coldstart node after transfer was stopped by using the FREEZE or READY command, the startup frame may not be transmitted in cycle 0. This depends on the internal state of the FlexRay module. This phenomenon occurs when the startup frame is set in a slot with a number from 1 to 7.

This phenomenon does not occur in a coldstart after a hardware reset.

Even if this phenomenon does occur, a second try of the coldstart will succeed.

While the coldstart time will be prolonged, this phenomenon does not otherwise affect a coldstart of the FlexRay system.

To avoid this phenomenon, allocate the Startup/Sync frame to static slot 8 or a slot with a higher number.

21.4.6 Communication Controller States

21.4.6.1 Communication Controller State Diagram

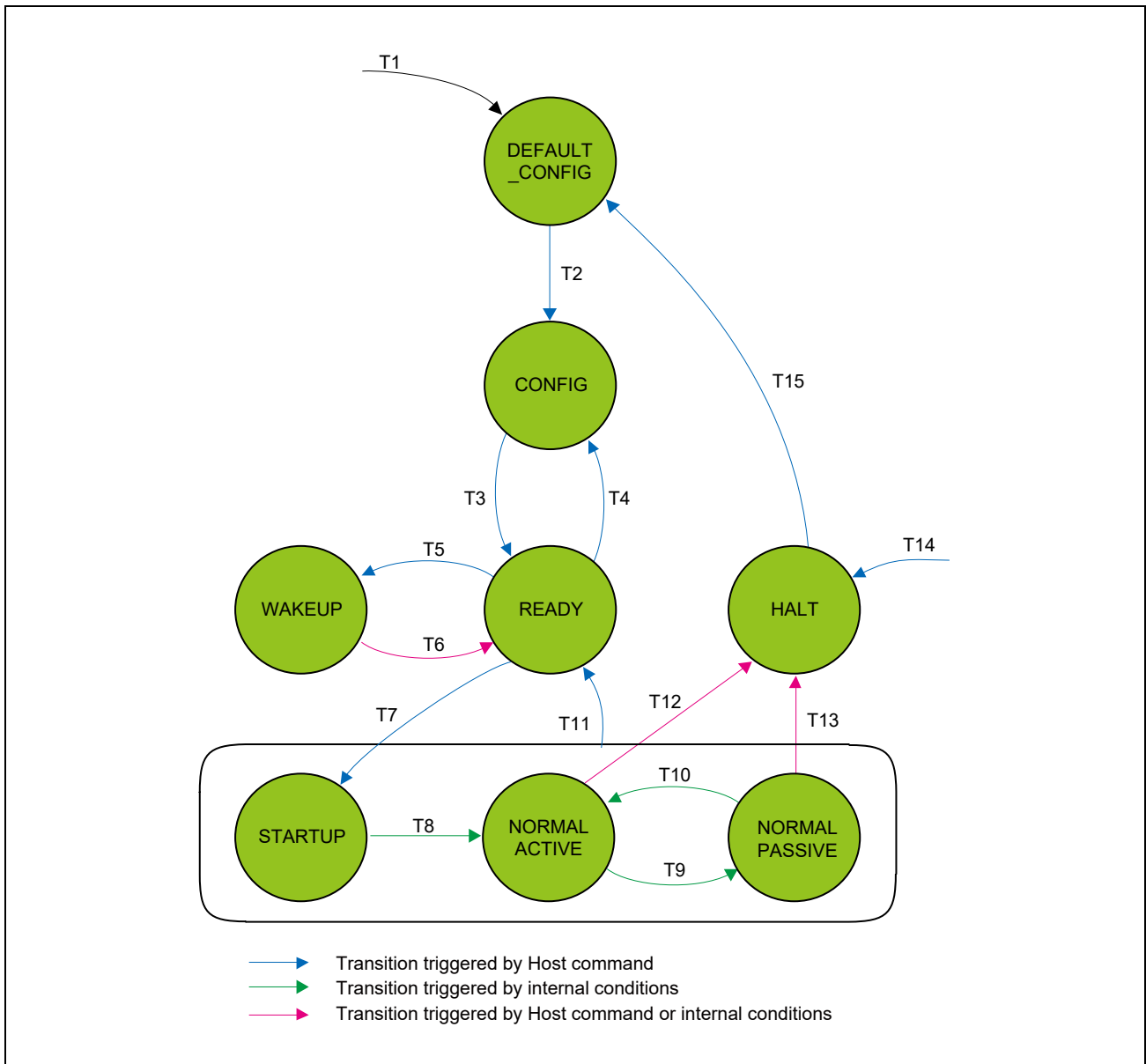


Figure 21.7 Overall State Diagram of FlexRay Communication Controller

State transitions are controlled by reset, FLXAnFR0RXDA, FLXAnFR0RXDB, by the POC state machine, and by the CHI command vector FLXAnFRSUCC1.CMD.

The CC transits from all states to HALT state after application of the FREEZE command (FLXAnFRSUCC1.CMD = “0111_B”).

Table 21.96 State Transitions of FlexRay Overall State Machine

T#	Condition	From	To
1	Reset	All States	DEFAULT_CONFIG
2	Command CONFIG, FLXAnFRSUCC1.CMD = "0001 _B "	DEFAULT_CONFIG	CONFIG
3	Unlock sequence followed by command READY, FLXAnFRSUCC1.CMD = "0010 _B "	CONFIG	READY
4	Command CONFIG, FLXAnFRSUCC1.CMD = "0001 _B "	READY	CONFIG
5	Command WAKEUP, FLXAnFRSUCC1.CMD = "0011 _B "	READY	WAKEUP
6	Complete transmission of wakeup pattern or received WUP or received frame header or wakeup collision or command READY, FLXAnFRSUCC1.CMD = "0010 _B "	WAKEUP	READY
7	Command RUN, FLXAnFRSUCC1.CMD = "0100 _B "	READY	STARTUP
8	Successful STARTUP	STARTUP	NORMAL_ACTIVE
9	Clock Correction Failed counter reached Maximum Without Clock Correction Passive limit configured by FLXAnFRSUCC3.WCP	NORMAL_ACTIVE	NORMAL_PASSIVE
10	Number of valid correction terms reached the Passive to Active limit configured by FLXAnFRSUCC1.PTA	NORMAL_PASSIVE	NORMAL_ACTIVE
11	Command READY, FLXAnFRSUCC1.CMD = "0010 _B "	STARTUP, NORMAL_ACTIVE, NORMAL_PASSIVE	READY
12	Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by FLXAnFRSUCC3.WCF when bit FLXAnFRSUCC1.HCSE set to '1' or command HALT, FLXAnFRSUCC1.CMD = "0110 _B "	NORMAL_ACTIVE	HALT
13	Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by FLXAnFRSUCC3.WCF when bit FLXAnFRSUCC1.HCSE set to '1' or command HALT, FLXAnFRSUCC1.CMD = "0110 _B "	NORMAL_PASSIVE	HALT
14	Command FREEZE, FLXAnFRSUCC1.CMD = "0111 _B "	All States	HALT
15	Command CONFIG, FLXAnFRSUCC1.CMD = "0001 _B "	HALT	DEFAULT_CONFIG

21.4.6.2 DEFAULT_CONFIG State

In DEFAULT_CONFIG state, the CC is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state.

The CC enters this state

- When the reset is applied (HW reset or SW reset)
- When exiting from HALT state

To leave DEFAULT_CONFIG state, the host has to write FLXAnFRSUCC1.CMD = “0001_B”. The CC then transits to CONFIG state.

21.4.6.3 CONFIG State

In CONFIG state, the CC is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state. This state is used to initialize the CC configuration.

The CC enters this state

- When exiting from DEFAULT_CONFIG state
- When exiting from READY state

When the state has been entered via HALT and DEFAULT_CONFIG state, the host can analyze status information and configuration. Before leaving CONFIG state, the host has to ensure that the configuration is fault-free.

To leave CONFIG state, the host has to perform the unlock sequence as described in **Section 21.3.3.1, FLXAnFRLCK — FlexRay Lock Register**. Directly after unlocking the CONFIG state, the host has to write FLXAnFRSUCC1.CMD to enter the next state.

CAUTION

Status bits FLXAnFRMHDS[14:0], registers FLXAnFRTXRQ1/2/3/4, and status data stored in the message RAM are not affected by the transition of the POC from CONFIG to READY state.

When the CC is in CONFIG state, it is also possible to bring the CC into a power saving mode by halting the module clocks (bus clock and sample clock). To do this, the host has to ensure that all message RAM transfers have finished before turning off the clocks.

21.4.6.4 READY State

After unlocking CONFIG state and writing `FLXAnFRSUCC1.CMD = "0010B"`, the CC enters READY state. From this state, the CC can transit to WAKEUP state and perform a cluster wake-up or to STARTUP state to perform a coldstart or to integrate into a running cluster.

The CC enters this state

- When exiting from CONFIG, WAKEUP, STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE state by writing `FLXAnFRSUCC1.CMD = "0010B"` (READY command).

The CC exits from this state

- To CONFIG state by writing `FLXAnFRSUCC1.CMD = "0001B"` (CONFIG command)
- To WAKEUP state by writing `FLXAnFRSUCC1.CMD = "0011B"` (WAKEUP command)
- To STARTUP state by writing `FLXAnFRSUCC1.CMD = "0100B"` (RUN command)

Internal counters and the CC status flags are reset when the CC enters STARTUP state.

CAUTION

Status bits `FLXAnFRMHDS[14:0]`, registers `FLXAnFRTXRQ1/2/3/4`, and status data stored in the message RAM are not affected by the transition of the POC from READY to STARTUP state.

21.4.6.5 WAKEUP State

The description below is intended to help configure wake-up for the FlexRay IP module. A detailed description of the wake-up procedure together with the respective SDL diagrams can be found in *FlexRay protocol specification v2.1, Section 7.1*.

The CC enters this state

- When exiting from READY state by writing FLXAnFRSUCC1.CMD = “0011_B” (WAKEUP command).

The CC exits from this state to READY state

- After complete non-aborted transmission of wakeup pattern (WUP)
- After WUP reception
- After detecting a WUP collision
- After reception of a frame header
- By writing FLXAnFRSUCC1.CMD = “0010_B” (READY command)

The cluster wakeup must precede the communication startup in order to ensure that all nodes in a cluster are awake. The minimum requirement for a cluster wakeup is that all bus drivers are supplied with power. A bus driver has the ability to wake up the other components of its node when it receives a wakeup pattern on its channel. At least one node in the cluster needs an external wakeup source.

The host completely controls the wakeup procedure. It is informed about the state of the cluster by the bus driver and the CC and configures bus guardian (if available) and the CC to perform the cluster wakeup. The CC provides to the host the ability to transmit a special wakeup pattern on each of its available channels separately. The CC needs to recognize the wakeup pattern only during WAKEUP state.

Wakeup can be performed on only one channel at a time. The host has to configure the wakeup channel while the CC is in CONFIG state by writing FLXAnFRSUCC1.WUCS. The CC ensures that ongoing communication on this channel is not disturbed. The CC cannot guarantee that all nodes connected to the configured channel awake upon the transmission of the wakeup pattern, since these nodes cannot give feedback until the startup phase. The wakeup procedure enables single-channel devices in a two-channel system to trigger the wakeup, by only transmitting the wakeup pattern on the single channel to which they are connected. Any coldstart node that deems a system startup necessary will then wake the remaining channel before initiating communication startup.

The wakeup procedure tolerates any number of nodes simultaneously trying to wakeup a single channel and resolves this situation such that only one node transmits the pattern. Additionally the wakeup pattern is collision resilient, so even in the presence of a fault causing two nodes to simultaneously transmit a wakeup pattern, the resulting collided signal can still wake the other nodes.

After wakeup, the CC returns to READY state and signals the change of the wakeup status to the host by setting flag FLXAnFRSIR.WST. The wakeup status vector can be read from FLXAnFRCCSV.WSV. If a valid wakeup pattern was received also, either flag FLXAnFRSIR.WUPA or flag FLXAnFRSIR.WUPB is set to ‘1’.

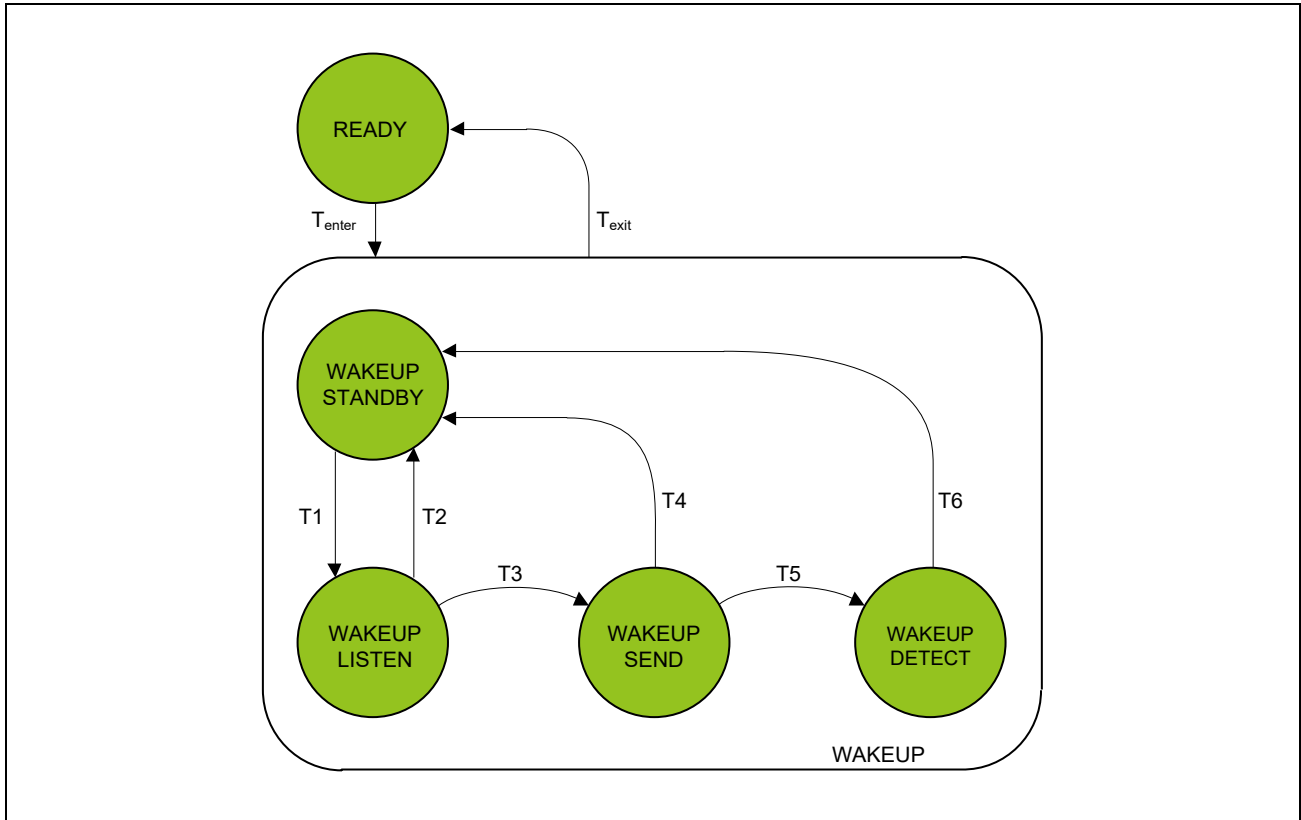


Figure 21.8 Structure of POC State WAKEUP

Table 21.97 State Transitions WAKEUP

T#	Condition	From	To
enter	Host commands change to WAKEUP state by writing FLXAnFRSUCC1.CMD = "0011 _B " (WAKEUP command)	READY	WAKEUP
1	CHI command WAKEUP triggers wakeup FSM to transit to WAKEUP_LISTEN state	WAKEUP_STANDBY	WAKEUP_LISTEN
2	Received WUP on wakeup channel selected by bit FLXAnFRSUCC1.WUCS or frame header on either available channel	WAKEUP_LISTEN	WAKEUP_STANDBY
3	Timer event	WAKEUP_LISTEN	WAKEUP_SEND
4	Complete, non-aborted transmission of wakeup pattern	WAKEUP_SEND	WAKEUP_STANDBY
5	Collision detected	WAKEUP_SEND	WAKEUP_DETECT
6	Wakeup timer expired or WUP detected on wakeup channel selected by bit FLXAnFRSUCC1.WUCS or frame header received on either available	WAKEUP_DETECT	WAKEUP_STANDBY
exit	Wakeup completed (after T2 or T4 or T6) or host commands change to READY state by writing FLXAnFRSUCC1.CMD = "0010 _B " (READY command). This command also resets the wakeup FSM to WAKEUP_STANDBY state	WAKEUP	READY

The WAKEUP_LISTEN state is controlled by the wakeup timer and the wakeup noise timer. The two timers are controlled by the parameters Listen Timeout FLXAnFRSUCC2.LT and Listen Timeout Noise FLXAnFRSUCC2.LTN. Listen Timeout enables a fast cluster wakeup in case of a noise free environment, while Listen Timeout Noise enables wakeup under more difficult conditions regarding noise interference.

In WAKEUP_SEND state, the CC transmits the wakeup pattern on the configured channel and checks for collisions. After return from wakeup, the host has to bring the CC into STARTUP state by CHI command RUN.

In WAKEUP_DETECT state, the CC attempts to identify the reason for the wakeup collision detected in WAKEUP_SEND state. The monitoring is limited by the expiration of listen timeout as configured by FLXAnFRSUCC2.LT. Either the detection of a wakeup pattern indicating a wakeup attempt by another node or the reception of a frame header indicating ongoing communication, causes the direct transition to READY state. Otherwise WAKEUP_DETECT is left after expiration of listen timeout; in this case, the reason for wakeup collision is unknown.

The host has to be aware of possible failures of the wakeup and act accordingly. It is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal time it takes another coldstart node to become awake and to be configured.

The FlexRay Protocol Specification recommends that two different CCs shall awake the two channels.

(1) Host Activities

The host must coordinate the wakeup of the two channels and must decide whether or not to wake a specific channel. The sending of the wakeup pattern is initiated by the host. The wakeup pattern is detected by the remote BDs and signaled to their respective local hosts.

Wakeup procedure controlled by host (single-channel wakeup):

- Configure the CC in CONFIG state
 - Select wakeup channel by programming bit FLXAnFRSUCC1.WUCS
- Check local BDs for whether a WUP was received
- Activate BD of selected wakeup channel
- Command CC to enter READY state
- Command CC to start wakeup on the configured channel by writing FLXAnFRSUCC1.CMD = “0011”
 - CC enters WAKEUP
 - CC returns to READY state and signals status of wakeup attempt to the host
- Wait predefined time to allow the other nodes to wakeup and configure themselves
- Coldstart node:
 - In a dual channel cluster, wait for WUP on the other channel
 - Reset coldstart inhibit flag FLXAnFRCCSV.CSI by writing FLXAnFRSUCC1.CMD = “1001” (ALLOW_COLDSTART command)
- Command CC to enter startup by writing FLXAnFRSUCC1.CMD = “0100” (RUN command)

Wakeup procedure triggered by BD:

- Wakeup recognized by BD
- BD triggers power-up of host (if required)
- BD signals wakeup event to host
- Host configures its local CC
- If necessary, host commands wakeup of second channel and waits predefined time to allow the other nodes to wake up and configure themselves
- Host commands CC to enter STARTUP state by writing FLXAnFRSUCC1.CMD = “0100” (RUN command)

(2) Wakeup Pattern (WUP)

The wakeup pattern (WUP) is composed of at least two wakeup symbols (WUS). Wakeup symbol and wakeup pattern are configured by registers FLXAnFRPRTC1 and FLXAnFRPRTC2.

- Single channel wakeup, wakeup symbol may not be sent on both channels at the same time
- Wakeup symbol collision resilient for at least two sending nodes (two overlapping wakeup symbols always recognizable)
- Wakeup symbol must be configured identically in all nodes of a cluster
- Wakeup symbol transmit low time configured by FLXAnFRPRTC2.TXL
- Wakeup symbol idle time used to listen for activity on the bus, configured by FLXAnFRPRTC2.TXI
- A wakeup pattern composed of at least two Tx-wakeup symbols needed for wakeup
- Number of repetitions configurable by FLXAnFRPRTC1.RWP (2 to 63 repetitions)
- Wakeup symbol receive window length configured by FLXAnFRPRTC1.RXW
- Wakeup symbol receive low time configured by FLXAnFRPRTC2.RXL
- Wakeup symbol receive idle time configured by FLXAnFRPRTC2.RXI

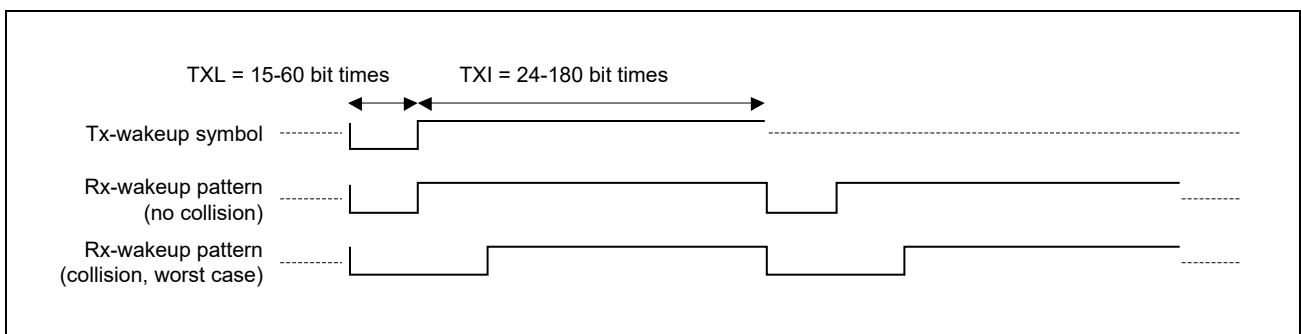


Figure 21.9 Timing of Wakeup Pattern

21.4.6.6 STARTUP State

This section describes the configuration of startup for the FlexRay module. A detailed description of the startup procedure together with the respective SDL diagrams can be found in FlexRay protocol specification v2.1, section 7.2.

Any node entering STARTUP state that has coldstart capability should ensure that both channels attached have been awakened before initiating coldstart.

The required time for complete wake-up and configuration vary for each node and star. Since at least two nodes are necessary to start up cluster communication, it is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal amount of time it takes another coldstart node to become awake, to be configured and to enter startup. It may require several hundred milliseconds (depending on the hardware used) before all nodes and stars are completely awakened and configured.

Startup is performed on all channels synchronously. During startup, a node only transmits startup frames. Startup frames are both sync frames and null frames during startup.

A fault-tolerant, distributed startup strategy is specified for initial synchronization of all nodes. In general, a node may enter NORMAL_ACTIVE state via following paths (see **Figure 21.10**):

- Coldstart path initiating the schedule synchronization (Leading coldstart node)
- Coldstart path joining other coldstart nodes (Following coldstart node)
- Integration path integrating into an existing communication schedule (all other nodes).

A coldstart attempt begins with the transmission of a collision avoidance symbol (CAS). Only a coldstart node that had transmitted the CAS transmits frames in the first four cycles after the CAS, it is then joined firstly by the other coldstart nodes and afterwards by all other nodes.

A coldstart node has bits FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY set to '1'. Message buffer 0 holds the key slot ID, which defines the slot number where the startup frame is send. In the frame header of the startup frame the startup frame indicator bit is set to '1'.

In clusters consisting of three or more nodes, at least three nodes shall be configured to be coldstart nodes. In clusters consisting of two nodes, both nodes must be coldstart nodes. At least two fault-free coldstart nodes are necessary for the cluster to start up.

Each startup frame must also be a sync frame; therefore each coldstart node will also be a sync node. The number of coldstart attempts is configured by FLXAnFRSUCC1.CSA.

A non-coldstart node requires at least two startup frames from distinct nodes for integration.

It may start integration before the coldstart nodes have finished their startup. It will not finish its startup until at least two coldstart nodes have finished their startup.

Both non-coldstart nodes and coldstart nodes start integration via the integration path as soon as they receive sync frames from which to derive the TDMA schedule information. During integration, the node has to adapt its own clock to the global clock (rate and offset) and has to make its cycle time consistent with the global schedule observable on the network. Afterwards, these settings are checked for consistency with all available network nodes. The node can only leave the integration phase and actively participate in communication when these checks are passed.

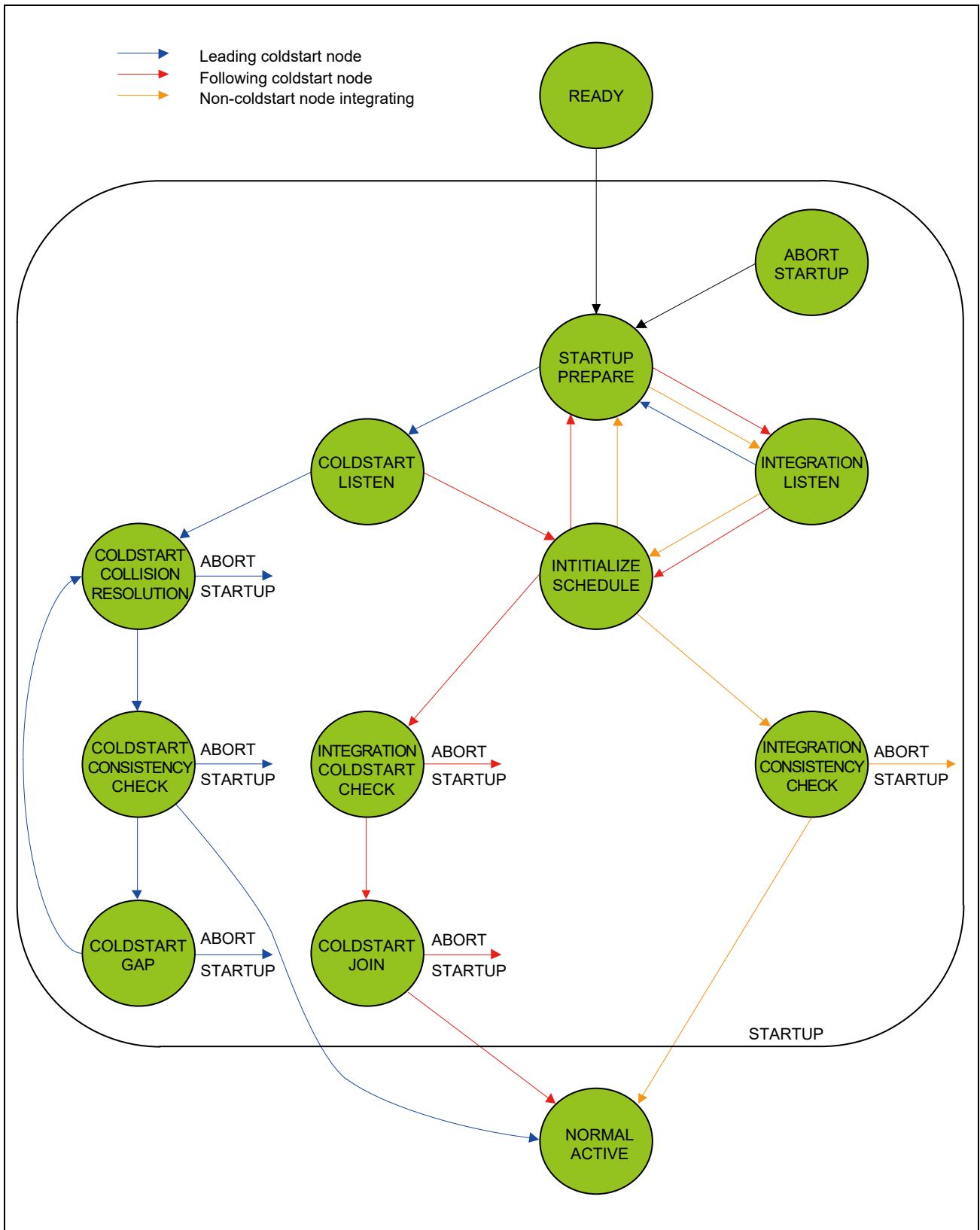


Figure 21.10 State Diagram Time-triggered Startup

(1) Coldstart Inhibit Mode

In coldstart inhibit mode, the node is prevented from initializing the TDMA communication schedule. If bit FLXAnFRCCSV.CSI is set to '1', the node is not allowed to initialize the cluster communication, i.e. entering the coldstart path is prohibited. The node is allowed to integrate to a running cluster or to transmit startup frames after another coldstart node started the initialization of the cluster communication.

The coldstart inhibit bit FLXAnFRCCSV.CSI is set to '1' whenever the POC enters READY state. The bit has to be cleared by issuing the CHI command ALLOW_COLDSTART (FLXAnFRSUCC1.CMD = "1001") by a program.

(2) Startup Timeouts

The CC supplies two different μ T timers supporting two timeout values, startup timeout and startup noise timeout. The two timers are started when the CC enters the COLDSTART_LISTEN state. The expiration of either of these timers causes the node to leave the initial sensing phase (COLDSTART_LISTEN state) with the intention of starting up communication.

CAUTION

The startup and startup noise timers are identical with the wakeup and wakeup noise timers and use the same configuration values FLXAnFRSUCC2.LT and FLXAnFRSUCC2.LTN.

(a) Startup Timeout

The startup timeout limits the monitoring time used by a node to determine if there is already communication between other nodes or at least one coldstart node actively requesting the integration of others. The startup timer is configured by programming FLXAnFRSUCC2.LT (see **Section 21.3.6.2, FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2**).

The startup timeout is:

$pdListenTimeout = FLXAnFRSUCC2.LT$

The startup timer is restarted upon:

- Entering the COLDSTART_LISTEN state
- Both channels reaching idle state while in COLDSTART_LISTEN state

The startup timer is stopped:

- If communication channel activity is detected on one of the configured channels while the node is in the COLDSTART_LISTEN state.
- When the COLDSTART_LISTEN state is left

Once the startup timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The timer status is kept for further processing by the startup state machine.

(b) Startup Noise Timeout

At the same time that the startup timer is started for the first time (transition from STARTUP_PREPARE state to COLDSTART_LISTEN state), the startup noise timer is started. This additional timeout is used to improve reliability of the startup procedure in the presence of noise. The startup noise timeout is configured by programming FLXAnFRSUCC2.LTN (see **Section 21.3.6.2, FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2**).

The startup noise timeout is:

$$\text{pdListenTimeout} * \text{gListenNoise} = \text{FLXAnFRSUCC2.LT} * (\text{FLXAnFRSUCC2.LTN} + 1)$$

The startup noise timer is restarted upon:

- Entering the COLDSTART_LISTEN state
- Reception of correctly decoded headers or CAS symbols while the node is in COLDSTART_LISTEN state

The startup noise timer is stopped when the COLDSTART_LISTEN state is left.

Once the startup noise timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The status is kept for further processing by the startup state machine. Since the startup noise timer will not be restarted when random channel activity is detected, this timeout defines the fall-back solution that guarantees that a node will try to start up the communication cluster even in the presence of noise.

(3) Path of Leading Coldstart Node (initiating coldstart)

When a coldstart node enters COLDSTART_LISTEN, it listens to its attached channels.

If no communication is detected, the node enters the COLDSTART_COLLISION_RESOLUTION state and commences a coldstart attempt. The initial transmission of a CAS symbol is succeeded by the first regular cycle. This cycle has the number zero.

From cycle zero on, the node transmits its startup frame. Since each coldstart node may perform a coldstart attempt, it may occur that several nodes simultaneously transmit the CAS symbol and enter the coldstart path. This situation is resolved during the first four cycles after CAS transmission.

As soon as a node that initiates a coldstart attempt receives a CAS symbol or a frame header during these four cycles, it re-enters the COLDSTART_LISTEN state. Thereby, only one node remains in this path. In cycle four, other coldstart nodes begin to transmit their startup frames.

After four cycles in COLDSTART_COLLISION_RESOLUTION state, the node that initiated the coldstart enters the COLDSTART_CONSISTENCY_CHECK state. It collects all startup frames from cycle four and five and performs the clock correction. If the clock correction does not deliver any errors and it has received at least one valid startup frame pair, the node leaves COLDSTART_CONSISTENCY_CHECK and enters NORMAL_ACTIVE state.

The number of coldstart attempts that a node is allowed to perform is configured by FLXAnFRSUCC1.CSA. The number of remaining coldstart attempts can be read from FLXAnFRCCSV.RCA. The number of remaining coldstart attempts is reduced by one for each attempted coldstart. A node may enter the COLDSTART_LISTEN state only if this value is 2 or larger and it may enter the COLDSTART_COLLISION_RESOLUTION state only if this value is 1 or larger. If the number of coldstart attempts is 1, coldstart is inhibited but integration is still possible.

(4) Path of Following Coldstart Node (responding to Leading Coldstart Node)

When a coldstart node enters the COLDSTART_LISTEN state, it tries to receive a valid pair of startup frames to derive its schedule and clock correction from the leading coldstart node.

As soon as a valid startup frame has been received, the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_COLDSTART_CHECK state is entered.

In INTEGRATION_COLDSTART_CHECK state, it is assured that the clock correction can be performed correctly and that the coldstart node from which this node has initialized its schedule is still valid. The node collects all sync frames and performs clock correction in the following double-cycle. If clock correction does not signal any errors and if the node continues to receive sufficient frames from the same node it has integrated on, the COLDSTART_JOIN state is entered.

In COLDSTART_JOIN state, following coldstart nodes begin to transmit their own startup frames and continue to do so in subsequent cycles. Thereby, the leading coldstart node and the nodes joining it can check if their schedules match with each other. If the clock correction signals any error, the node aborts the integration attempt. If a node in this state sees at least one valid startup frame during all even cycles in this state and at least one valid startup frame pair during all double cycles in this state, the node leaves COLDSTART_JOIN state and enters NORMAL_ACTIVE state. Thereby it leaves STARTUP at least one cycle after the node that initiated the coldstart.

(5) Path of Non-Coldstart Node

When a non-coldstart node enters the INTEGRATION_LISTEN state, it listens to its attached channels.

As soon as a valid startup frame has been received, the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_CONSISTENCY_CHECK state is entered.

In INTEGRATION_CONSISTENCY_CHECK state, the node verifies that the clock correction can be performed correctly and that enough coldstart nodes (at least 2) are sending startup frames that agree with the node's own schedule. Clock correction is activated, and if any errors are signaled, the integration attempt is aborted.

During the first even cycle in this state, either two valid startup frames or the startup frame of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

During the first double-cycle in this state, either two valid startup frame pairs or the startup frame pair of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

If after the first double-cycle, less than two valid startup frames are received within an even cycle, or less than two valid startup frame pairs are received within a double-cycle, the startup attempt is aborted.

Nodes in this state need to detect two valid startup frame pairs for two consecutive double-cycles each to be allowed to leave STARTUP and enter NORMAL_OPERATION. Consequently, they leave startup at least one double-cycle after the node that initiated the coldstart and only at the end of a cycle with an odd cycle number.

21.4.6.7 NORMAL_ACTIVE State

As soon as the node that transmitted the first CAS symbol (resolving the potential access conflict and entering STARTUP via coldstart path) and one additional node have entered the NORMAL_ACTIVE state, the startup phase for the cluster has finished. In the NORMAL_ACTIVE state, all configured messages are scheduled for transmission. This includes all data frames as well as the sync frames. Rate and offset measurement is started in all even cycles (even/odd cycle pairs required).

In NORMAL_ACTIVE state, the CC supports regular communication functions.

- The CC performs transmissions and reception on the FlexRay bus as configured
- Clock synchronization is running
- The host interface is operational

The CC exits from that state to

- HALT state by writing FLXAnFRSUCC1.CMD = "0110_B" (HALT command, at the end of the current cycle)
- HALT state by writing FLXAnFRSUCC1.CMD = "0111_B" (FREEZE command, immediately)
- HALT state due to change of the error state from ACTIVE to COMM_HALT
- NORMAL_PASSIVE state due to change of the error state from ACTIVE to PASSIVE
- READY state by writing FLXAnFRSUCC1.CMD = "0010_B" (READY command)

21.4.6.8 NORMAL_PASSIVE State

NORMAL_PASSIVE state is entered from NORMAL_ACTIVE state when the error state changes from ACTIVE to PASSIVE.

In NORMAL_PASSIVE state, the node is able to receive all frames (node is fully synchronized and performs clock synchronization). Contrary to the NORMAL_ACTIVE state, the node does not actively participate in communication, i.e. neither symbols nor frames are transmitted.

In NORMAL_PASSIVE state

- The CC performs reception on the FlexRay bus
- The CC does not transmit any frames or symbols on the FlexRay bus
- Clock synchronization is running
- The host interface is operational

The CC exits from this state to

- HALT state by writing FLXAnFRSUCC1.CMD = "0110_B" (HALT command, at the end of the current cycle)
- HALT state by writing FLXAnFRSUCC1.CMD = "0111_B" (FREEZE command, immediately)
- HALT state due to change of the error state from PASSIVE to COMM_HALT
- NORMAL_ACTIVE state due to change of the error state from PASSIVE to ACTIVE. The transition takes place when FLXAnFRCCEV.PTAC equals FLXAnFRSUCC1.PTA - 1
- To READY state by writing FLXAnFRSUCC1.CMD = "0010_B" (READY command)

21.4.6.9 HALT State

In this state, all communication (reception and transmission) is stopped.

The CC enters this state

- By writing FLXAnFRSUCC1.CMD = “0110_B” (HALT command) while the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state
- By writing FLXAnFRSUCC1.CMD = “0111_B” (FREEZE command) from all states
- When exiting from NORMAL_ACTIVE state because the FLXAnFRSUCC1.HCSE is set to ‘1’ and the clock correction failed counter reached the “maximum without clock correction fatal” limit.
- When exiting from NORMAL_PASSIVE state because the FLXAnFRSUCC1.HCSE is set to ‘1’ and the clock correction failed counter reached the “maximum without clock correction fatal” limit.

The CC exits from this state to DEFAULT_CONFIG state

- By writing FLXAnFRSUCC1.CMD = “0001_B” (CONFIG command)

When the CC enters HALT state, all configuration and status data are maintained for analyzing purposes.

When the host writes FLXAnFRSUCC1.CMD = “0110_B” (HALT command), the CC sets bit FLXAnFRCCSV.HRQ to ‘1’ and enters HALT state at the next end of cycle.

When the host writes FLXAnFRSUCC1.CMD = “0111_B” (FREEZE command), the CC enters HALT state immediately and sets bit FLXAnFRCCSV.FSI to ‘1’.

The POC state from which the transition to HALT state took place can be read from FLXAnFRCCSV.PSL.

21.4.7 Network Management

The accrued Network Management (NM) vector can be read from registers FLXAnFRNMV1...3. The CC performs a bit-wise OR operation over all NM vectors out of all received valid NM frames with the Payload Preamble Indicator (PPI) bit set. Only static frames may be configured to hold NM information. The CC updates the NM vector at the end of each cycle. The length of the NM vector can be configured from 0 to 12 bytes by FLXAnFRNEMC.NML. The NM vector length must be configured identically in all nodes of a cluster.

To configure a transmit buffer to send FlexRay frames with the PPI bit set as '1', bit PPIT in the header section of the respective transmit buffer has to be set to '1' via FLXAnFRWRHS1.PPIT. In addition the host has to write the NM information to the data section of the respective transmit buffer.

The evaluation of the NM vector has to be done by the application running on the host.

Section 21.4.17, Byte Alignment, for byte alignment of the received NM vector in registers FLXAnFRNMV1 to FLXAnFRNMV3.

CAUTIONS

1. In case a message buffer is configured for transmission/reception of network management frames, the payload length configured in header 2 of that message buffer should be equal to or greater than the length of the NM vector configured by FLXAnFRNEMC.NML.
 2. When the CC transits to HALT state, the cycle count is not incremented and therefore the NM vector is not updated. In this case FLXAnFRNMV1...3 holds the value from the cycle before.
-

21.4.8 Filtering and Masking

Filtering is done by comparison of the configuration of assigned message buffers against actual slot and cycle counter values and channel ID (channel A, B). A message buffer is only updated/transmitted if the required matches occur.

Filtering is done on:

- Slot Counter
- Cycle Counter
- Channel ID

The following filter combinations for acceptance/transmit filtering are allowed:

- Slot Counter + Channel ID
- Slot Counter + Cycle Counter + Channel ID

All configured filters must match in order to store a received message in a message buffer.

CAUTION

For the FIFO, the acceptance filter is configured by the FIFO Rejection Filter (FLXAnFRFRF) and the FIFO Rejection Filter Mask (FLXAnFRFRFM).

A message will be transmitted in the time slot corresponding to the configured frame ID on the configured channel(s). If cycle counter filtering is enabled, the configured cycle filter value must also match.

21.4.8.1 Slot Counter Filtering

Every transmit and receive buffer contains a frame ID stored in the header section. This frame ID is compared against the actual slot counter value in order to assign receive and transmit buffers to the corresponding slot.

If two or more message buffers are configured with the same frame ID and channel ID, and if they have a matching cycle counter filter value for the same slot, then the message buffer with the lowest message buffer number is used.

21.4.8.2 Cycle Counter Filtering

Cycle counter filtering is based on the concept of a cycle set. For filtering purposes, a match is detected if any one of the elements of the cycle set is matched. The cycle set is defined by the cycle code field in header section 1 of each message buffer.

If message buffer 0 or 1 is configured to hold the startup/sync frame or the single slot frame by bits FLXAnFRSUCC1.TXST, FLXAnFRSUCC1.TXSY, and FLXAnFRSUCC1.TSM, cycle counter filtering for message buffer 0 or 1 shall be disabled.

CAUTION

Sharing of a static time slot via cycle counter filtering between different nodes of a FlexRay network is not allowed.

The set of cycle numbers belonging to a cycle set is determined as described in **Table 21.98**.

Table 21.98 Definition of Cycle Set

Cycle Code	Matching Cycle Counter Values
0b000000x	All Cycles
0b000001c	Every second Cycle at $(\text{Cycle Count}) \bmod 2 = c$
0b00001cc	Every fourth Cycle at $(\text{Cycle Count}) \bmod 4 = cc$
0b0001ccc	Every eighth Cycle at $(\text{Cycle Count}) \bmod 8 = ccc$
0b001cccc	Every sixteenth Cycle at $(\text{Cycle Count}) \bmod 16 = cccc$
0b01ccccc	Every thirty-second Cycle at $(\text{Cycle Count}) \bmod 32 = ccccc$
0b1cccccc	Every sixty-fourth Cycle at $(\text{Cycle Count}) \bmod 64 = ccccccc$

Table 21.99 below gives some examples for valid cycle sets to be used for cycle counter filtering.

Table 21.99 Examples for Valid Cycle Sets

Cycle Code	Matching Cycle Counter Values
0b0000011	1-3-5-7-.... -63
0b0000100	0-4-8-12-.... -60
0b0001110	6-14-22-30-.... -62
0b0011000	8-24-40-56
0b0100011	3-35
0b1001001	9

The received message is stored only if the cycle counter value of the cycle during which the message is received matches an element of the receive buffer's cycle set. Channel ID and frame ID must also be match.

The frame is transmitted when the cycle code set in the transmit buffer matches the current cycle counter value. Channel ID and frame ID must also be match.

21.4.8.3 Channel ID Filtering

There is a 2-bit channel filtering control field (CH) located in the header section of each message buffer in the message RAM. It serves as a filter for receive buffers, and as a control field for transmit buffers (see **Table 21.100**).

Table 21.100 Channel Filtering Configuration

CH[1:0]	Transmit Buffer Transmit Frame	Receive Bufferstore Valid Receive Frame
11 _B	on both channels (static segment only)	received on channel A or B (store first semantically valid frame, static segment only)
10 _B	on channel B	received on channel B
01 _B	on channel A	received on channel A
00 _B	Transmission prohibited	ignore frame

The contents of a transmit buffer is transmitted on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a transmit buffer may be set up for transmission on both channels (CH = “11_B”).

Valid received frames are stored if they are received on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a receive buffer may be setup for reception on both channels (CH = “11_B”).

CAUTION

If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to ‘1’, no frames are transmitted and received frames are ignored (same function as CH = “00_B”).

21.4.8.4 FIFO Filtering

For FIFO filtering, registers FLXAnFRFRF and FLXAnFRFRFM are used. The FIFO filter consists of channel filter FLXAnFRFRF.CH, frame ID filter FLXAnFRFRF.FID, and cycle counter filter FLXAnFRFRF.CYF. Registers FLXAnFRFRF and FLXAnFRFRFM can be configured in DEFAULT_CONFIG or CONFIG state only. The filter configuration in the header section of message buffers belonging to the FIFO is ignored.

The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles not belonging to the cycle set specified by FLXAnFRFRF.CYF, all frames are rejected.

A valid received frame is stored in the FIFO if channel ID, frame ID, and cycle counter are not rejected by FLXAnFRFRF and FLXAnFRFRFM register settings, and if there is no matching dedicated receive buffer.

21.4.9 Transmit Process

21.4.9.1 Static Segment

For the static segment, if there are several messages pending for transmission, the message with the frame ID corresponding to the next sending slot is selected for transmission.

The data section of transmit buffers assigned to the static segment can be updated until the end of the preceding time slot. This means that a transfer from the input buffer has to be started by writing to the Input Buffer Command Request register (FLXAnFRIBCR) latest at this time.

21.4.9.2 Dynamic Segment

In the dynamic segment, if several messages are pending, the message with the highest priority (lowest frame ID) is selected next. In the dynamic segment, different slot counter sequences on channel A and channel B are possible (concurrent sending of different frame IDs on both channels).

The data section of transmit buffers assigned to the dynamic segment can be updated until the end of the preceding slot. This means that a transfer from the input buffer has to be started by writing to the Input Buffer Command Request register (FLXAnFRIBCR) latest at this time.

The start of latest transmit configured by FLXAnFRMHDC.SLT defines the maximum minislot value allowed before inhibiting new frame transmission in the dynamic segment of the current cycle.

21.4.9.3 Transmit Buffers

FlexRay message buffers can be configured as transmit buffers by programming bit CFG in the header section of the corresponding message buffer to '1' via FLXAnFRWRHS1.

There exist the following possibilities to assign a transmit buffer to the CC channels:

- Static segment: channel A or channel B,
channel A and channel B.
- Dynamic segment: channel A or channel B.

Message buffers 0 and 1 are dedicated to hold the startup frame, the sync frame, or the designated single slot frame as configured by FLXAnFRSUCC1.TXST, FLXAnFRSUCC1.TXSY, and FLXAnFRSUCC1.TSM, respectively. In this case, they can be reconfigured in DEFAULT_CONFIG or CONFIG state only. This ensures that any node transmits at most one startup/sync frame per communication cycle. Transmission of startup/sync frames from other message buffers is not possible.

All other message buffers configured for transmission in static or dynamic segment are reconfigurable during runtime depending on the configuration of FLXAnFRMRC.SEC (see **Section 21.4.12.1, Reconfiguration of Message Buffers**). Due to the structure of the data partition in the message RAM (to be referenced by data pointer), reconfiguration of the configured payload length and the data pointer in the header section of a message buffer may lead to erroneous configurations.

If a message buffer is reconfigured (header section updated) during runtime, it may happen that this message buffer is not sent out in the corresponding communication cycle.

The CC does not have the capability to calculate the header CRC. The header CRC needs to be provided to all transmit buffers by a program. If network management is required, the host has to set the PPIT bit in the header section of the

respective message buffer to '1' and write the network management information to the data section of the message buffer (see **Section 21.4.7, Network Management**).

The payload length field configures the payload length in 2-byte words. If the configured payload length of a static transmit buffer is shorter than the payload length configured for the static segment by FLXAnFRMHDC.SFDL, the CC generates padding bytes to ensure that frames have proper physical length. The padding pattern is "0000_H".

CAUTION

In case of an odd payload length (PLC = 1,3,5,...), the application has to write zero to the last 16 bits of the message buffers data section to ensure that the padding pattern is "0000_H".

Each transmit buffer provides a transmission mode flag TXM that allows the host to configure the transmission mode for the transmit buffer. If this bit is set, the transmitter operates in single-shot mode. If this bit is cleared, the transmitter operates in continuous mode.

In single-shot mode, the CC resets the respective TXR flag to '0' after transmission has completed. Now the host may update the transmit buffer.

In continuous mode, the CC does not reset the respective transmission request flag TXR to '0' after successful transmission. In this case a frame is sent out each time the filter criteria match. The TXR flag can be reset to '0' by the host by writing the respective message buffer number to the FLXAnFRIBCR register while bit FLXAnFRIBCM.STXRH is set to '0'.

If two or more transmit buffers meet the filter criteria simultaneously, the transmit buffer with the lowest message buffer number will be transmitted in the respective slot.

21.4.9.4 Frame Transmission

The following steps are required to prepare a message buffer for transmission:

- Configure the transmit buffer in the message RAM via FLXAnFRWRHS1, FLXAnFRWRHS2, and FLXAnFRWRHS3
- Write the data section of the transmit buffer via FLXAnFRWRDSx
- Transfer the configuration and message data from input buffer to the message RAM by writing the number of the target message buffer to register FLXAnFRIBCR
- If configured in register FLXAnFRIBCM, the transmission request flag TXR for the respective message buffer will be set as soon as the transfer has completed, and the message buffer is ready for transmission.
- Check whether the message buffer has been transmitted by checking the respective TXR bit (TXR = '0') in the FLXAnFRTXRQ1/2/3/4 registers (single-shot mode only).

After transmission has completed, the respective TXR flag in the FLXAnFRTXRQ1/2/3/4 register is reset to '0' (single-shot mode), and if bit MBI in the header section of the message buffer is set to '1', flag FLXAnFRSIR.TXI is set to '1'. If enabled, an interrupt is generated.

21.4.9.5 Null Frame Transmission

If in the static segment the host does not set the transmission request flag to '1' before transmit time, the CC transmits a null frame with the null frame indication bit set to '0' and the payload data set to zero.

In the following cases, the CC transmits a null frame:

- If the message buffer with the lowest message buffer number matching the filter criteria does not have its transmission request flag set to '1' (TXR = '0').
- No transmit buffer configured for the slot has a cycle counter filter that matches the current cycle. In this case, no message buffer status FLXAnFRMBS is updated.

Null frames are not transmitted in the dynamic segment.

21.4.10 Receive Process

21.4.10.1 Dedicated Receive Buffers

Some FlexRay message buffers can be configured as dedicated receive buffers by programming bit CFG in the header section of the respective message buffer to '0' via FLXAnFRWRHS1.

The following possibilities exist to assign a receive buffer to the CC channels:

- Static segment: channel A or channel B,
channel A and channel B (the CC stores the first semantically valid frame)
- Dynamic segment: channel A or channel B

The CC transfers the payload data of valid received messages from the shift register of the FlexRay channel protocol controller (channel A or B) to the receive buffer with the matching filter configuration. A receive buffer stores all frame elements except the frame CRC.

All message buffers configured for reception in static or dynamic segment are reconfigurable during runtime depending on the configuration of FLXAnFRMRC.SEC (see **Section 21.4.12.1, Reconfiguration of Message Buffers**). If a message buffer is reconfigured (header section updated) during runtime, it may happen that in the respective communication cycle a received message is lost.

If two or more receive buffers meet the filter criteria simultaneously, the receive buffer with the lowest message buffer number is updated with the received message.

21.4.10.2 Frame Reception

The following steps are required to prepare a dedicated message buffer for reception:

- Configure the receive buffer in the message RAM via FLXAnFRWRHS1, FLXAnFRWRHS2, and FLXAnFRWRHS3
- Transfer the configuration from input buffer to the message RAM by writing the number of the target message buffer to register FLXAnFRIBCR

Once these steps are performed, the message buffer functions as an active receive buffer and participates in the internal acceptance filtering process which takes place every time the CC receives a message. The first matching receive buffer is updated from the received message.

If a valid payload segment was stored in the data section of a message buffer, the respective ND flag in the FLXAnFRNDAT1/2/3/4 registers is set to '1', and, if bit MBI in the header section of that message buffer is set to '1', flag FLXAnFRSIR.RXI is set to '1'. If enabled, an interrupt is generated.

In case that bit ND was already set to '1' when the message handler updates the message buffer, bit FLXAnFRMBS.MLST of the respective message buffer is set to '1' and the unprocessed message data is lost.

If a frame was not received, or if a null frame or a corrupted frame was received in a slot, the data section of the message buffer configured for this slot is not updated. In this case only the respective message buffer status FLXAnFRMBS is updated.

When the message handler changed the message buffer status FLXAnFRMBS in the header section of a message buffer, the respective MBC flag in the FLXAnFRMBSC1/2/3/4 registers is set to '1', and if bit MBI in the header section of that message buffer is set, flag FLXAnFRSIR.MBSI is set to '1'. If enabled an interrupt is generated.

If the payload length of a received frame PLR is longer than the value programmed by PLC in the header section of the respective message buffer, the data field stored in the message buffer is truncated to that length.

To read a receive buffer from the message RAM via the output buffer, proceed as described in **Section 21.4.12.2(2), Data Transfer from Message RAM to Output Buffer.**

CAUTION

The ND and MBC flags are automatically cleared by the message handler when the payload data and the header of a received message have been transferred to the output buffer, respectively.

21.4.10.3 Null Frame Reception

The payload segment of a received null frame is not copied into the matching dedicated receive buffer. If a null frame has been received, only the message buffer status FLXAnFRMBS of the matching message buffer is updated from the received null frame. All bits in headers 2 and 3 of the matching message buffer remain unchanged. They are updated from received data frames only.

When the message handler changes the message buffer status FLXAnFRMBS in the header section of a message buffer, the respective MBC flag in the FLXAnFRMBSC1/2/3/4 register is set to '1', and if bit MBI in the header section of that message buffer is set to '1', flag FLXAnFRSIR.MBSI is set to '1'. If enabled, an interrupt is generated.

21.4.11 FIFO Function

21.4.11.1 Description

Some message buffers can be configured as a cyclic First-In-First-Out (FIFO) buffer. The group of message buffers belonging to the FIFO is contiguous in the register map starting with the message buffer referenced by FLXAnFRMRC.FFB and ending with the message buffer referenced by FLXAnFRMRC.LCB. Up to 127 message buffers can be assigned to the FIFO.

Every valid incoming message not matching with any dedicated receive buffer but passing the programmable FIFO filter is stored into the FIFO. In this case, frame ID, payload length, receive cycle count, and the message buffer status FLXAnFRMBS of the addressed FIFO message buffer are overwritten with frame ID, payload length, receive cycle count, and the status from the received frame. It indicates that the FIFO is not empty when the RFNE flag in the FLXAnFRSIR register is set to '1', the receive FIFO fill level FLXAnFRFSR.RFFL is equal to or greater than the critical level configured by FLXAnFRSIR.RFCL bit when FLXAnFRFCL.RFCL bit is 1, and a FIFO overrun has been detected when FLXAnFREIR.RFO is 1, respectively. If enabled, interrupts are generated.

If null frames are not removed by the FIFO rejection filter, the null frames will be treated like data frames when they are stored into the FIFO.

There are two index registers associated with the FIFO. The PUT Index Register (PIDX) is an index to the next available location in the FIFO. When a new message has been received, it is written into the message buffer addressed by the PIDX register. The PIDX register is then incremented and addresses the next available message buffer. If the PIDX register is incremented past the highest numbered message buffer of the FIFO, the PIDX register is loaded with the number of the first (lowest numbered) message buffer in the FIFO chain. The GET Index Register (GIDX) is used to address the next message buffer of the FIFO to be read. The GIDX register is incremented after transfer of the contents of a message buffer belonging to the FIFO to the output buffer. The PUT Index Register and the GET Index Register are not accessible by the host.

The FIFO is completely filled when the PUT index (PIDX) reaches the value of the GET index (GIDX). When the next message is written to the FIFO before the oldest message has been read, both PUT index and GET index are incremented and the new message overwrites the oldest message in the FIFO. This will set FIFO overrun flag FLXAnFREIR.RFO to '1'.

When the PUT index (PIDX) differs from the GET index (GIDX), it indicates that the FIFO is not empty. In this case, flag FLXAnFRSIR.RFNE is set to '1'. This indicates that there is at least one received message in the FIFO. The FIFO empty, FIFO not empty, and the FIFO overrun states are explained in **Figure 21.11** for a three message buffer FIFO.

The programmable FlexRay FIFO Rejection Filter (FLXAnFRFRF) defines a filter pattern for messages to be rejected. The FIFO filter consists of channel filter, frame ID filter, and cycle counter filter. If bit FLXAnFRFRF.RSS is set to '1', all messages received in the static segment are rejected by the FIFO. If bit FLXAnFRFRF.RNF is set to '1', received null frames are not stored in the FIFO.

The FlexRay FIFO Rejection Filter Mask (FLXAnFRFRFM) specifies which bits of the frame ID filter in the FIFO Rejection Filter register are marked 'don't care' for rejection filtering.

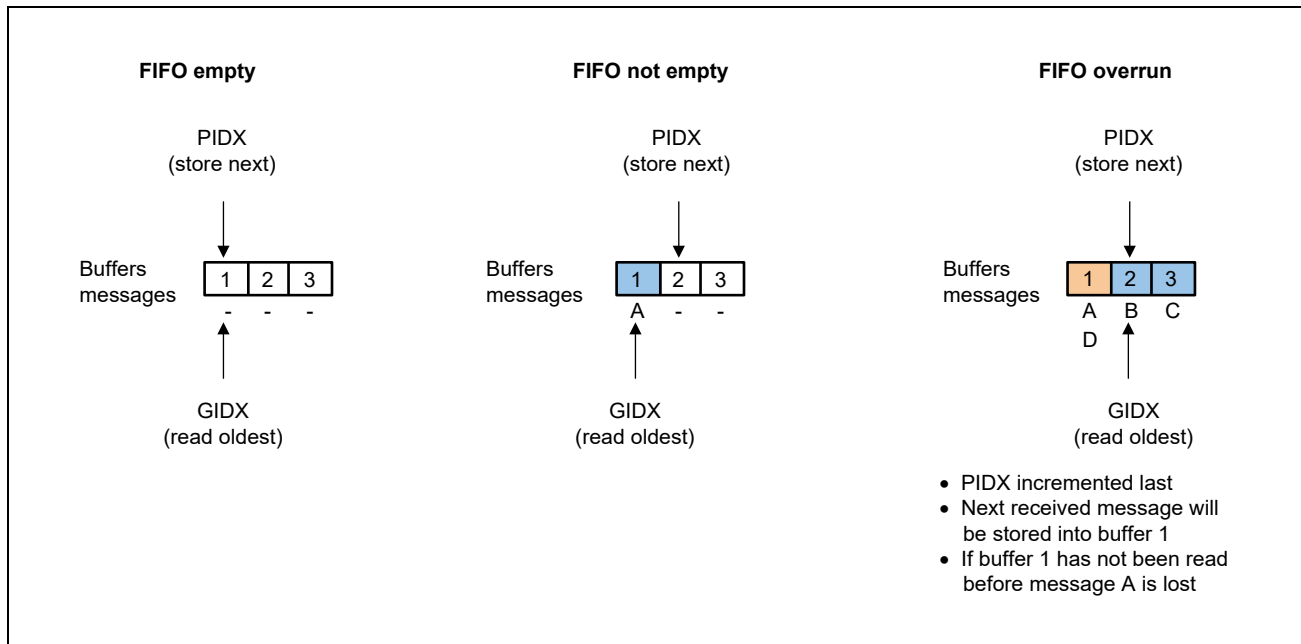


Figure 21.11 FIFO Status: Empty, Not Empty, Overrun

21.4.11.2 Configuration of the FIFO

(Re)configuration of message buffers belonging to the FIFO is only possible when the CC is in DEFAULT_CONFIG or CONFIG state. While the CC is in DEFAULT_CONFIG or CONFIG state, the FIFO function is not available.

For all message buffers belonging to the FIFO the payload length configured need to be programmed to the same value via FLXAnFRWRHS2.PLC. The data pointer to the first 32-bit word of the data section of the respective message buffer in the message RAM has to be configured via FLXAnFRWRHS3.DP.

All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask registers. The values configured in the header sections of the message buffers belonging to the FIFO, with exception of DP and PLC, have no meaning.

CAUTIONS

1. It is recommended to program the MBI bits of the message buffers belonging to the FIFO to '0' via FLXAnFRWRHS1.MBI to avoid generation of RX interrupts.
2. If the payload length of a received frame is longer than the value programmed by FLXAnFRWRHS2.PLC in the header section of the respective message buffer, the data field stored in a message buffer of the FIFO is truncated to that length.

21.4.11.3 Access to the FIFO

(1) When the output buffer is used:

For FIFO access outside DEFAULT_CONFIG and CONFIG state, the host has to trigger a transfer from the message RAM to the output buffer by writing the number of the first message buffer of the FIFO (referenced by FLXAnFRMRC.FFB) to the register FLXAnFROBCR. The message handler then transfers the message buffer addressed by the GET index register (GIDX) to the output buffer. After this transfer the GET index register (GIDX) is incremented.

(2) When the data transfer function is used:

The message received in FIFO can be transferred to the Local RAM/Cluster RAM by using the output data transfer function. For the output data transfer function, see **Section 21.4.16.2, Output Data Transfer**.

21.4.12 Message Handling

The message handler controls data transfers between the input/output buffer and the message RAM and between the message RAM and the two temporary buffers.

Access to the message buffers stored in the message RAM is done under control of the message Handler state machine. This avoids conflicts between accesses of the two FlexRay channel protocol controllers and the host to the message RAM.

Frame IDs of message buffers assigned to the static segment have to be in the range from 1 to FLXAnFRGTUC7.NSS. Frame IDs of message buffers assigned to the dynamic segment have to be in the range from FLXAnFRGTUC7.NSS + 1 to 2047.

Received messages with no matching dedicated receive buffer (static or dynamic segment) are stored in the receive FIFO (if configured) if they pass the FIFO rejection filter.

Access of the host to the message buffer contents using the input or output buffer function is described in this subsection. Access to the message buffer contents using the data transfer function is mentioned in **Section 21.4.16, Usage of Data Transfer**.

21.4.12.1 Reconfiguration of Message Buffers

In case that an application needs to handle with more than 128 different messages, static and dynamic message buffers can be reconfigured during FlexRay operation. This is done by updating the header section of the respective message buffer via input buffer registers FLXAnFRWRHS1...3.

Reconfiguration has to be enabled via control bits FLXAnFRMRC.SEC in the message RAM Configuration register.

If a message buffer has not been transmitted/updated from a received frame before reconfiguration starts, the respective message is lost.

The point in time when a reconfigured message buffer is ready for transmission/reception according to the reconfigured frame ID depends on the actual state of the slot counter when the update of the header section has completed. Therefore it may happen that a reconfigured message buffer is not transmitted/updated from a received frame in the cycle where it was reconfigured.

The message RAM is scanned according to **Table 21.101 Scan of Message RAM** below.

Table 21.101 Scan of Message RAM

Start of Scan In Slot	Scan for Slots
1	2...15, 1 (next cycle)
8	16...23, 1 (next cycle)
16	24...31, 1 (next cycle)
24	32...39, 1 (next cycle)
...	...

A message RAM scan is terminated with the start of NIT regardless of whether it has completed or not. The scan of the message RAM for slots 2 to 15 starts at the beginning of slot 1 of the actual cycle. The scan of the message RAM for slot 1 is done in the cycle before by checking in parallel to each scan of the message RAM whether there is a message buffer configured for slot 1 of the next cycle.

The number of the first dynamic message buffer is configured by FLXAnFRMRC.FDB. In case a message RAM scan starts while the CC is in dynamic segment, the scan starts with the message buffer number configured by FLXAnFRMRC.FDB.

In case a message buffer should be reconfigured to be used in slot 1 of the next cycle, the followings have to be considered:

- If the message buffer to be reconfigured for slot 1 is part of the “Static Buffers”, it will only be detected if it is reconfigured before the last message RAM scan in the static segment of the actual cycle evaluates this message buffer.
- If the message buffer to be reconfigured for slot 1 is part of the “Static + Dynamic Buffers”, it will be detected if it is reconfigured before the last message RAM scan in the actual cycle evaluates this message buffer.
- The start of NIT terminates the message RAM scan. In case the message RAM scan has not evaluated the reconfigured message buffer by this point in time, the message buffer will not be considered for the next cycle.

CAUTION

Reconfiguration of message buffers may lead to the loss of messages and therefore has to be performed very carefully. In the worst case (reconfiguration in consecutive cycles), a message buffer may never be transmitted or may not be updated from a received frame.

21.4.12.2 Host Access to Message RAM

The message transfer between input buffer and message RAM as well as between message RAM and output buffer is triggered by the host by writing the number of the target and source message buffer to be accessed to FLXAnFRIBCR or FLXAnFROBCR register.

The FLXAnFRIBCM and FLXAnFROBCM registers can be used to write/read header and data section of the selected message buffer separately.

If bit FLXAnFRIBCM.STXR is set to '1', the transmission request flag TXR of the selected message buffer is automatically set to '1' after the message buffer has been updated. If bit FLXAnFRIBCM.STXR is reset to '0', the transmission request flag TXR of the selected message buffer is reset. This can be used to stop transmission from message buffers operated in continuous mode.

Input buffer (IBF) and output buffer (OBF) are built up as a double buffer structure. One half of this double buffer structure is accessible by the host (IBF host or OBF host), while the other half (IBF shadow or OBF shadow) is accessed by the message handler for data transfers between IBF/OBF and message RAM.

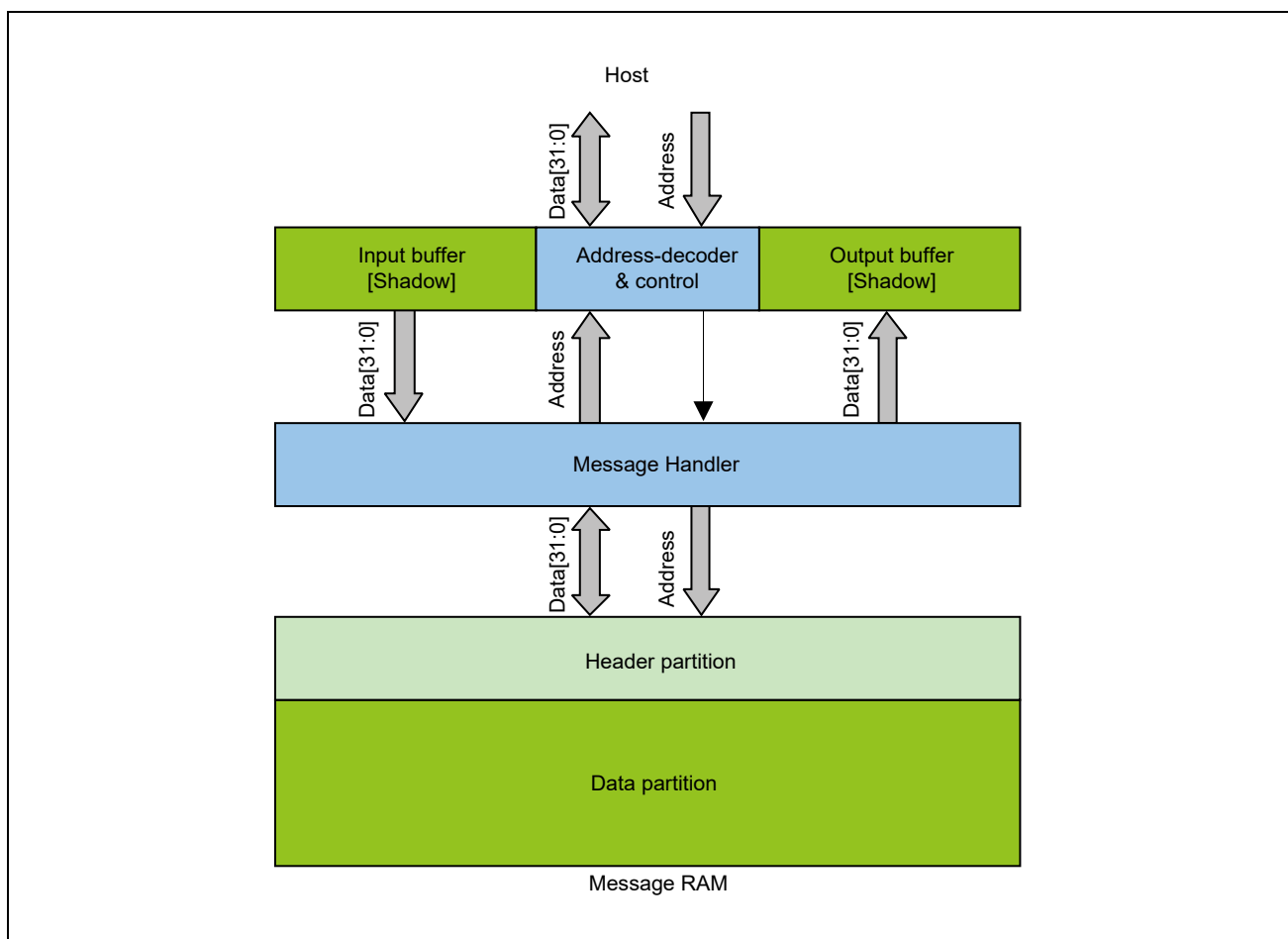


Figure 21.12 Host Access to Message RAM

(1) Data Transfer from Input Buffer to Message RAM

To configure or update a message buffer in the message RAM, the host has to write the data to FLXAnFRWRDSx and the header to FLXAnFRWRHS1...3. The specific action is selected by configuring the FlexRay input buffer command mask register FLXAnFRIBCM.

When the host writes the number of the target message buffer in the message RAM to FLXAnFRIBCR.IBRH, IBF host and IBF shadow are switched (see **Figure 21.13**).

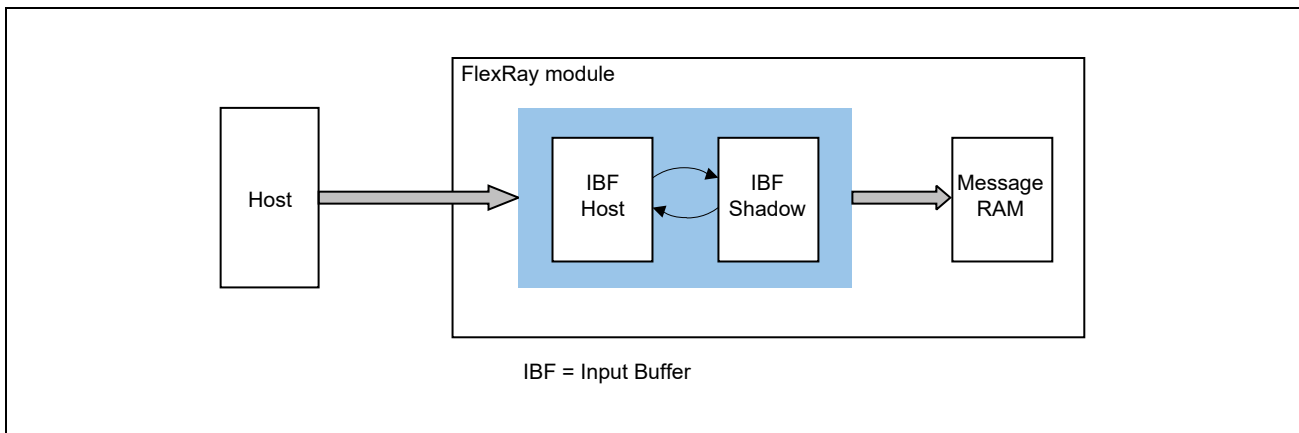


Figure 21.13 Double Buffer Structure Input Buffer

In addition the bits in the FLXAnFRIBCM and FLXAnFRIBCR registers are also switched while they hold the value in each IBF section (see **Figure 21.14**).

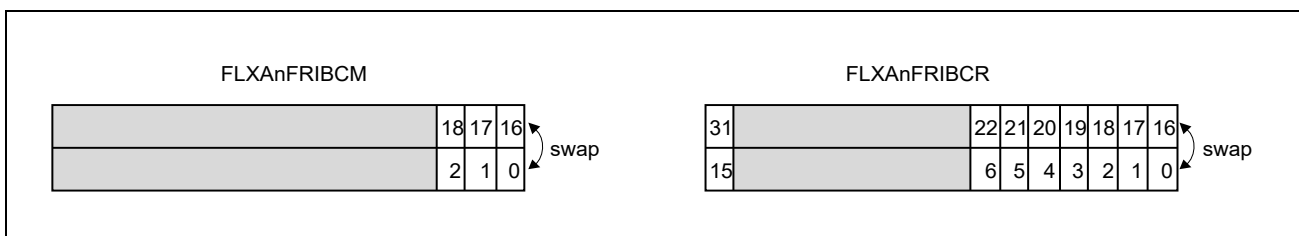


Figure 21.14 Switching of FLXAnFRIBCM and FLXAnFRIBCR Bits

With this write operation, bit FLXAnFRIBCR.IBSYS is set to '1'. The message handler then starts to transfer the contents of IBF shadow to the message buffer in the message RAM selected by FLXAnFRIBCR.IBRS.

While the message handler transfers the data from IBF shadow to the target message buffer in the message RAM, the host can write the next message to IBF host. After the transfer between IBF shadow and the message RAM has completed, bit FLXAnFRIBCR.IBSYS is set back to '0' and the next transfer to the message RAM may be started by the host by writing the respective target message buffer number to FLXAnFRIBCR.IBRH.

If a write access to FLXAnFRIBCR.IBRH is made while FLXAnFRIBCR.IBSYS is '1', FLXAnFRIBCR.IBSYH is set to '1'. After completion of the ongoing data transfer from IBF shadow to the message RAM, IBF host and IBF shadow are switched, FLXAnFRIBCR.IBSYH is reset to '0', FLXAnFRIBCR.IBSYS remains set to '1', and the next transfer to the message RAM is started. In addition the message buffer numbers stored under FLXAnFRIBCR.IBRH and FLXAnFRIBCR.IBRS and the command mask flags are also switched.

Example of an 8/16/32-bit host access sequence:

Configure or update n-th message buffer via IBF

- Wait until FLXAnFRIBCR.IBSYH is reset
- Write data section to FLXAnFRWRDSx
- Write header section to FLXAnFRWRHS1...3
- Write command mask: set FLXAnFRIBCM.STXRH, FLXAnFRIBCM.LDSH, FLXAnFRIBCM.LHSH
- Demand data transfer to target message buffer: write FLXAnFRIBCR.IBRH

Configure or update (n + 1)th message buffer via IBF

- Wait until FLXAnFRIBCR.IBSYH is reset
- Write data section to FLXAnFRWRDSx
- Write header section to FLXAnFRWRHS1...3
- Write command mask: write FLXAnFRIBCM.STXRH, FLXAnFRIBCM.LDSH, FLXAnFRIBCM.LHSH
- Demand data transfer to target message buffer: write FLXAnFRIBCR.IBRH

CAUTION

Any write access to IBF while FLXAnFRIBCR.IBSYH is '1' will set error flag FLXAnFREIR.IIBA to '1'. In this case, the write access is ignored.

Table 21.102 Assignment of FLXAnFRIBCM Bits

Pos.	Access	Bit	Function
18	r	STXRS	Set Transmission Request Shadow Flag (ongoing or finished)
17	r	LDSS	Load Data Section Shadow Flag (ongoing or finished)
16	r	LHSS	Load Header Section Shadow Flag (ongoing or finished)
2	r/w	STXRH	Set Transmission Request Host
1	r/w	LDSH	Load Data Section Host
0	r/w	LHSH	Load Header Section Host

Table 21.103 Assignment of FLXAnFRIBCR Bits

Pos.	Access	Bit	Function
31	r	IBSYS	IBF Busy Shadow Flag, signals ongoing transfer from IBF Shadow to message RAM
22...16	r	IBRS	IBF Request Shadow Flag, number of message buffer currently or lately updated
15	r	IBSYH	IBF Busy Host Flag, transfer request pending for message buffer referenced by IBRH
6...0	r/w	IBRH	IBF Request Host, number of message buffer to be updated next

(2) Data Transfer from Message RAM to Output Buffer

To read a message buffer from the message RAM, the host has to write to register FLXAnFROBCR to trigger the data transfer as configured in FLXAnFROBCM. After the transfer has completed, the host can read the transferred data from FLXAnFRRDDSx, FLXAnFRRDHS1...3, and FLXAnFRMBS.

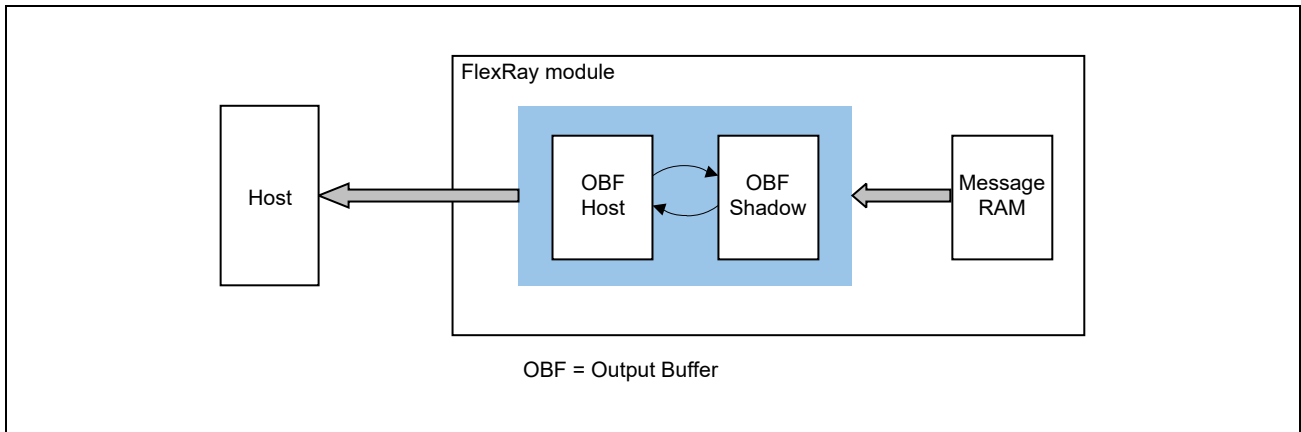


Figure 21.15 Double Buffer Structure Output Buffer

OBF host and OBF shadow as well as bits FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS, FLXAnFROBCM.RHSH, FLXAnFROBCM.RDSH and bits FLXAnFROBCR.OBRS, FLXAnFROBCR.OBRH are switched under control of bits FLXAnFROBCR.VIEW and FLXAnFROBCR.REQ.

Writing bit FLXAnFROBCR.REQ to ‘1’ copies bits FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS and bits FLXAnFROBCR.OBRS to an internal memory (see **Figure 21.16**).

After setting FLXAnFROBCR.REQ to ‘1’, FLXAnFROBCR.OBSYS is set to ‘1’, and the transfer of the message buffer selected by FLXAnFROBCR.OBRS from the message RAM to OBF shadow is started. After the transfer between the message RAM and OBF shadow has completed, the FLXAnFROBCR.OBSYS bit is set back to ‘0’. Bits FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW can only be set to ‘1’ while FLXAnFROBCR.OBSYS is ‘0’.

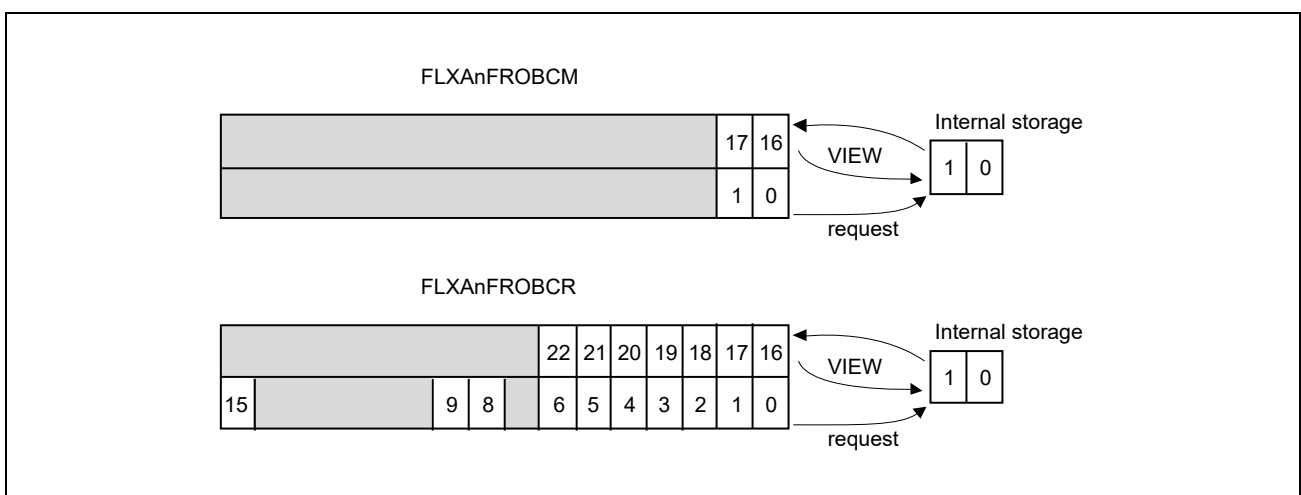


Figure 21.16 Switching of FLXAnFROBCM and FLXAnFROBCR Bits

OBF host and OBF shadow are switched by setting bit FLXAnFROBCR.VIEW to ‘1’ while bit FLXAnFROBCR.OBSYS is ‘0’ (see **Figure 21.15**).

In addition, bits FLXAnFROBCR.OBRH and bits FLXAnFROBCM.RHSH, FLXAnFROBCM.RDSH are switched with the registers internal memory, thus assuring that the message buffer number stored in FLXAnFROBCR.OBRH and the mask configuration stored in FLXAnFROBCM.RHSH, FLXAnFROBCM.RDSH matches the transferred data stored in OBF host (see **Figure 21.16**).

Now the host can read the transferred message buffer from OBF host while the message handler transfers the next message from the message RAM to OBF shadow.

If bits REQ and VIEW are set to '1' with the same write access while FLXAnFROBSYS is '0', FLXAnFROBSYS is automatically set to '1' and OBF shadow and OBF host are switched. Additionally mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are switched with the registers internal storage to keep them attached to the respective output buffer transfer. Afterwards FLXAnFROBCR.OBRS is copied to the register internal storage, mask bits FLXAnFROBCM.RDSS and FLXAnFROBCM.RHSS are copied to register FLXAnFROBCM internal memory, and the transfer of the selected message buffer from the message RAM to OBF shadow is started. While the transfer is ongoing the host can read the message buffer transferred by the previous transfer from OBF host. When the current transfer between message RAM and OBF shadow has completed, completion of the transfer is notified by setting FLXAnFROBCR.OBSYS back to '0'.

Example of an 8/16/32-bit host access to a single message buffer:

If a single message buffer has to be read out, two separate write accesses to FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW are necessary:

- Wait until FLXAnFROBCR.OBSYS is reset
- Write output buffer command mask FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS
- Request transfer of message buffer to OBF shadow by writing FLXAnFROBCR.OBRS and FLXAnFROBCR.REQ (in case of an 8-bit host interface, FLXAnFROBCR.OBRS has to be written before FLXAnFROBCR.REQ).
- Wait until FLXAnFROBCR.OBSYS is reset
- Toggle OBF shadow and OBF host by writing FLXAnFROBCR.VIEW = '1'
- Read out transferred message buffer by reading FLXAnFRRDDs, FLXAnFRRDHS1...3, and FLXAnFRMBS

Example of an 8/16/32-bit host access sequence:

Request transfer of 1st message buffer to OBF shadow

- Wait until FLXAnFROBCR.OBSYS is reset
- Write output buffer command mask FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS for 1st message buffer
- Request transfer of 1st message buffer to OBF shadow by writing FLXAnFROBCR.OBRS and FLXAnFROBCR.REQ (in case of an 8-bit host interface, FLXAnFROBCR.OBRS has to be written before FLXAnFROBCR.REQ).

Toggle OBF shadow and OBF host to read out 1st transferred message buffer and request transfer of 2nd message buffer:

- Wait until FLXAnFROBCR.OBSYS is reset to '0'
- Write output buffer command mask FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS for 2nd message buffer
- Toggle OBF shadow and OBF host and start transfer of 2nd message buffer to OBF shadow simultaneously by writing FLXAnFROBCR.OBRS of 2nd message buffer, FLXAnFROBCR.REQ, and FLXAnFROBCR.VIEW (in

case of and 8-bit host interface, FLXAnFROBCR.OBRS has to be written before FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW).

- Read out 1st transferred message buffer by reading FLXAnFRRDDSx, FLXAnFRRDHS1...3, and FLXAnFRMBS

Demand access to last requested message buffer without a request of another message buffer:

- Wait until FLXAnFROBCR.OBSYS is reset to '0'
- Demand access to last transferred message buffer by writing FLXAnFROBCR.VIEW
- Read out last transferred message buffer by reading FLXAnFRRDDSx, FLXAnFRRDHS1...3, and FLXAnFRMBS

Table 21.104 Assignment of FLXAnFROBCM Bits

Pos.	Access	Bit	Function
17	r	RDSH	Data Section Accessible
16	r	RHSH	Header Section Accessible
1	r/w	RDSS	Read Data Section Shadow
0	r/w	RHSS	Read Header Section Shadow

Table 21.105 Assignment of FLXAnFROBCR Bits

Pos.	Access	Bit	Function
22...16	r	OBRH	OBH Host Transfer Request Flag, number of message buffer accessible
15	r	OBSYS	OBH Busy Shadow Flag, signals ongoing transfer from message RAM to OBH shadow
9	r/w	REQ	Request transfer from message RAM to OBH shadow
8	r/w	VIEW	View OBH Shadow, swap OBH shadow and OBH host
6...0	r/w	OBRS	OBH Request Shadow, number of message buffer for next request

21.4.12.3 FlexRay Protocol Controller Access to Message RAM

The two temporary buffers (TBF A and TBF B) are used to buffer the data for transfer between the two FlexRay protocol controllers and the message RAM.

Each temporary buffer is build up as a double buffer, and able to store two complete FlexRay messages. There is always one buffer assigned to the corresponding protocol controller while the other one is accessible by the message handler.

If for example the message handler writes the next message to be sent to temporary buffer Tx, the FlexRay channel protocol controller can access temporary buffer Rx to store the message it is actually receiving. During transmission of the message stored in temporary buffer Tx, the message handler transfers the last received message stored in temporary buffer Rx to the message RAM (if it passes acceptance filtering) and updates the corresponding message buffer.

Data transfers between the temporary buffers and the shift registers of the FlexRay channel protocol controllers are done in words of 32 bit. This enables the use of a 32 bit shift register regardless of the length of the FlexRay messages.

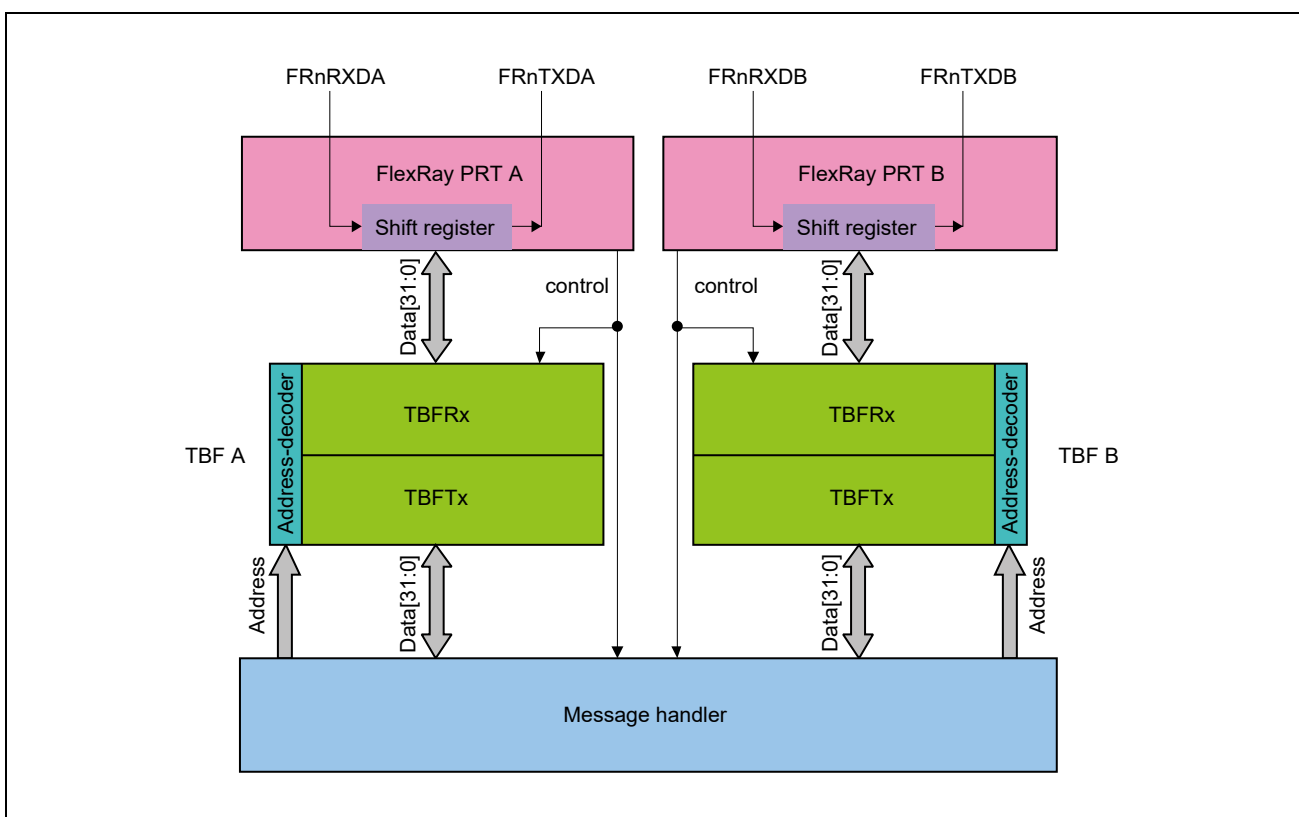


Figure 21.17 Access to Temporary Buffers

21.4.13 Message RAM

To avoid conflicts between host access to the message RAM and FlexRay message reception/transmission, the host cannot directly access the message buffers in the message RAM. These accesses are handled via the input and output buffers. The Message RAM is able to store up to 128 message buffers depending on the configured payload length.

The message RAM is able to store up to 2048 32-bit words. To achieve the required flexibility with respect to different numbers of data bytes per FlexRay frame (0...254), the message RAM has a structure as shown in **Figure 21.18**.

When the message buffer of the data section to be allocated after the header partition is set as a reception buffer (by setting the FLXAnFRWRHS1.CFG bit to 0) or as a reception FIFO buffer, set an unused area of at least 32 bits at the start of the data section. In this case, the data partition can be started from the nth word in the message RAM where n calculated from (the setting of the FLXAnFRMRC.LCB[7:0] bits + 1) × 4 + 1.

When the message buffer of the data section to be allocated after the header partition is set as a transmission buffer (by setting the FLXAnFRWRHS1.CFG bit to 1), the data partition can be started from the nth word in the message RAM where n calculated from (the setting of the FLXAnFRMRC.LCB[7:0] bits + 1) × 4.

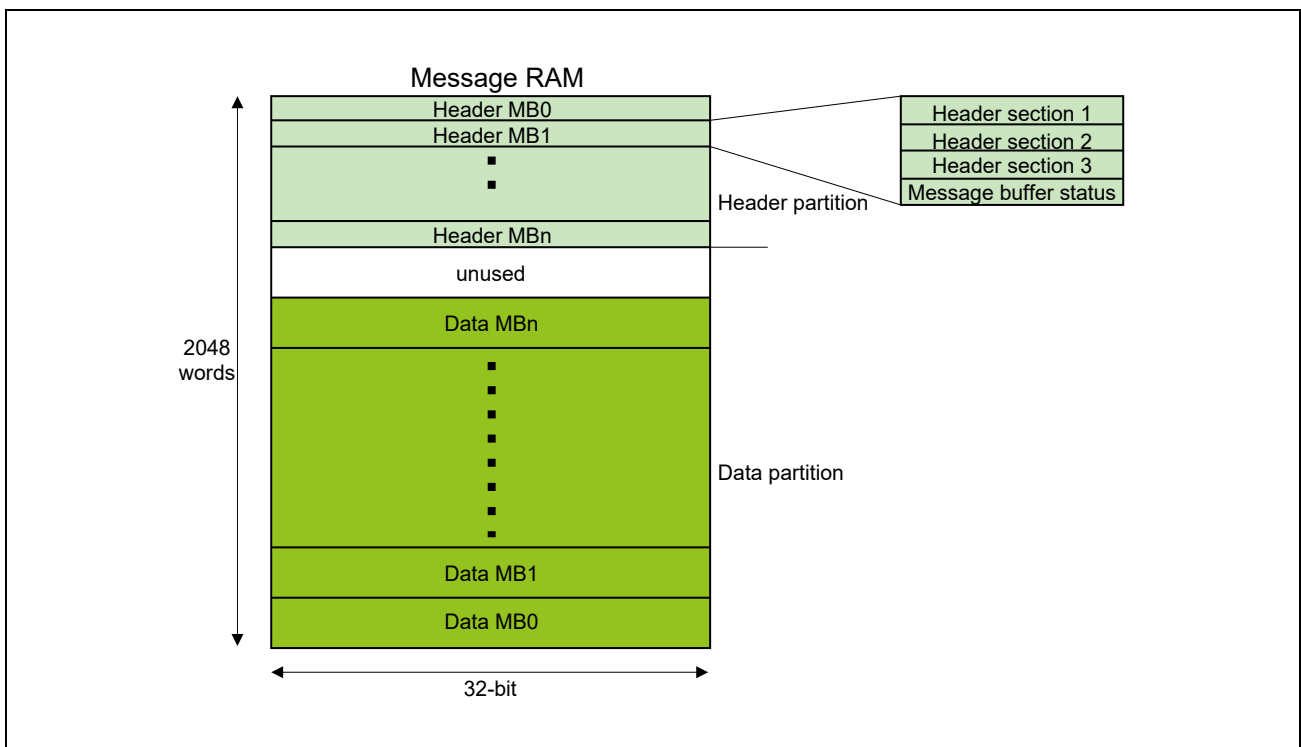


Figure 21.18 Configuration Example of Message Buffers in the Message RAM

Header Partition

Stores header sections of the configured message buffers:

- Supports a maximum of 128 message buffers
- Each message buffer has a header section of four 32-bit words
- Header 3 of each message buffer holds the 11-bit data pointer to the respective data section in the data partition

Data Partition

Flexible storage of data sections with different length. Some maximum values are:

- 30 message buffers with 254 byte data section each
- Or 56 message buffers with 128 byte data section each
- Or 128 message buffers with 48 byte data section each

CAUTION

Header partition + data partition may not occupy more than 2048 32-bit words.

21.4.13.1 Header Partition

The elements used for configuration of a message buffer as well as the actual message buffer status are stored in the header partition of the message RAM as listed in **Table 21.106** below. Configuration of the header sections of the message buffers is done via IBF (FLXAnFRWRHS1...3). Read access to the header sections is done via OBF (FLXAnFRRDHS1...3 + FLXAnFRMBS). The data pointer has to be calculated by the user to define the starting point of the data section for the corresponding message buffer in the data partition of the message RAM. The data pointer should not be modified during runtime. For message buffers belonging to the receive FIFO (re)configuration is possible in DEFAULT_CONFIG or CONFIG state only.

The header section of each message buffer occupies four 32-bit words in the header partition of the message RAM. The header of message buffer 0 starts with the first word in the message RAM.

For transmit buffers the Header CRC has to be calculated by the host.

The followings are updated from received valid data frames only:

Payload Length Received (PLR bit), Receive Cycle Count (RCC bit), Received on Channel Indicator (RCI bit), Startup Frame Indicator (SFI bit), Sync Frame Indicator (SYN bit), Null Frame Indicator (NFI bit), Payload Preamble Indicator (PPI bit), and Reserved Bit (RES).

Table 21.106 Header Section of a Message Buffer in the Message RAM

Bit Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0			M B I	T X M	P P I T	C F G	CH			Cycle code																Frame ID							
1		Payload length received								Payload length configured															Tx buffer: Header CRC configured Rx buffer: Header CRC configured								
2			R E S	P I S	N I S	S Y N	S F I	S F I	R C I			Receive cycle count													Data pointer								
3			R E S	P I S	N I S	S Y N	S F I	S F I	R C I			Cycle count status						F T P	F T A	M L S T	E S B	E S A	T C I B	T C I A	S V O B	S V O A	C E O B	C E O A	S E O B	S E O A	V F R B	V F R A	
...	...																																
...	...																																

- Frame configuration
- Filter configuration
- Message buffer control
- Message RAM configuration
- Updated from received data frame
- Message buffer status (MBS)
- Unused

(1) Header Section 1 (word 0)

Write access via FLXAnFRWRHS1, read access via FLXAnFRRDHS1:

- Frame ID
Slot counter filtering configuration
- Cycle Code
Cycle counter filtering configuration
- CH
Channel filtering configuration
- CFG
Message buffer direction configuration: reception or transmission
- PPIT
Payload Preamble Indicator Transmit
- TXM
Transmit mode configuration: single-shot or continuous
- MBI
Message buffer receive/transmit interrupt enable

(2) Header Section 2 (word 1)

Write access via FLXAnFRWRHS2, read access via FLXAnFRRDHS2:

- Header CRC
Transmit Buffer: Configured by the host (calculated from frame header)
Receive Buffer: Updated from received frame
- Payload Length Configured
Length of data section (2-byte words) as configured by the host
- Payload Length Received
Length of payload segment (2-byte words) stored from received frame

(3) Header Section 3 (word 2)

Write access via FLXAnFRWRHS3, read access via FLXAnFRRDHS3:

- Data Pointer
Pointer to the beginning of the corresponding data section in the data partition

Read access via FLXAnFRRDHS3, valid for receive buffers only, updated from received frames:

- Receive Cycle Count
Cycle count from received frame

- RCI
Received on Channel Indicator
- SFI
Startup Frame Indicator
- SYN
Sync Frame Indicator
- NFI
Null Frame Indicator
- PPI
Payload Preamble Indicator
- RES
Reserved bit

(4) Message Buffer Status FLXAnFRMBS (word 3)

Read access via FLXAnFRMBS, updated by the CC at the end of the configured slot.

- VFRA
Valid Frame Received on channel A
- VFRB
Valid Frame Received on channel B
- SEOA
Syntax Error Observed on channel A
- SEOB
Syntax Error Observed on channel B
- CEOA
Content Error Observed on channel A
- CEOB
Content Error Observed on channel B
- SVOA
Slot boundary Violation Observed on channel A
- SVOB
Slot boundary Violation Observed on channel B
- TCIA
Transmission Conflict Indication channel A
- TCIB
Transmission Conflict Indication channel B

- ESA
Empty Slot Channel A
- ESB
Empty Slot Channel B
- MLST
Message Lost
- FTA
Frame Transmitted on Channel A
- FTB
Frame Transmitted on Channel B
- Cycle Count Status
Actual cycle count when status was updated
- RCIS
Received on Channel Indicator Status
- SFIS
Startup Frame Indicator Status
- SYNS
Sync Frame Indicator Status
- NFIS
Null Frame Indicator Status
- PPIS
Payload Preamble Indicator Status
- RESS
Reserved bit Status

21.4.13.2 Data Partition

The data partition of the message RAM stores the data sections of the message buffers configured for reception/transmission as defined in the header partition. The number of data bytes for each message buffer can vary from 0 to 254. To optimize the data transfer between the shift registers of the two FlexRay protocol controllers and the message RAM as well as between the host interface and the message RAM, the physical width of the message RAM is set to 4 bytes.

The data partition starts after the last word of the header partition. When configuring the message buffers in the message RAM the user has to ensure that the data pointers point to addresses within the data partition. **Table 21.107** below shows an example how the data sections of the configured message buffers can be stored in the data partition of the message RAM.

The beginning and the end of a message buffer's data section is determined by the data pointer and the payload length configured in the message buffer's header section, respectively. This enables a flexible usage of the available RAM space for storage of message buffers with different data length.

If the size of the data section is an odd number of 2-byte words, the remaining 16 bits in the last 32-bit word are unused (see **Table 21.107** below).

Table 21.107 Example for Structure of the Data Partition in the Message RAM

Bit Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
...	Unused								Unused								Unused								Unused							
...	Unused								Unused								Unused								Unused							
...	MBn data 3								MBn data 2								MBn data 1								MBn data 0							
...							
...							
...	MBn data (m)								MBn data (m-1)								MBn data (m-2)								MBn data (m-3)							
...							
...							
...							
...	MB1 data 3								MB1 data 2								MB1 data 1								MB1 data 0							
...							
2046	MB0 data 3								MB0 data 2								MB0 data 1								MB0 data 0							
2047	Unused								Unused								MB0 data 5								MB0 data 4							

21.4.13.3 Message Data Integrity Check

A data integrity checking mechanism is implemented in the FlexRay core to assure the integrity of the data stored in the related RAM. Each RAM has a checksum generator and checker attached as shown in. **Figure 21.19**.

When data is written to a RAM, the local checksum generator generates the checksum. The checksum is stored together with the respective data word. The checksum is checked each time a data word is read from a RAM.

If a checksum error is detected, the respective access error flag is set. The access error flags FLXAnFRMHDS.AMR, FLXAnFRMHDS.ATBF1, FLXAnFRMHDS.ATBF2 and the faulty message buffer indicators FLXAnFRMHDS.FMBD, FLXAnFRMHDS.MFMB, FLXAnFRMHDS.FMB are located in the FlexRay message handler status register. These single access error flags control the error interrupt flag FLXAnFREIR.AERR.

Figure 21.19 shows the data paths between the input buffer, temporary buffer and message RAM.

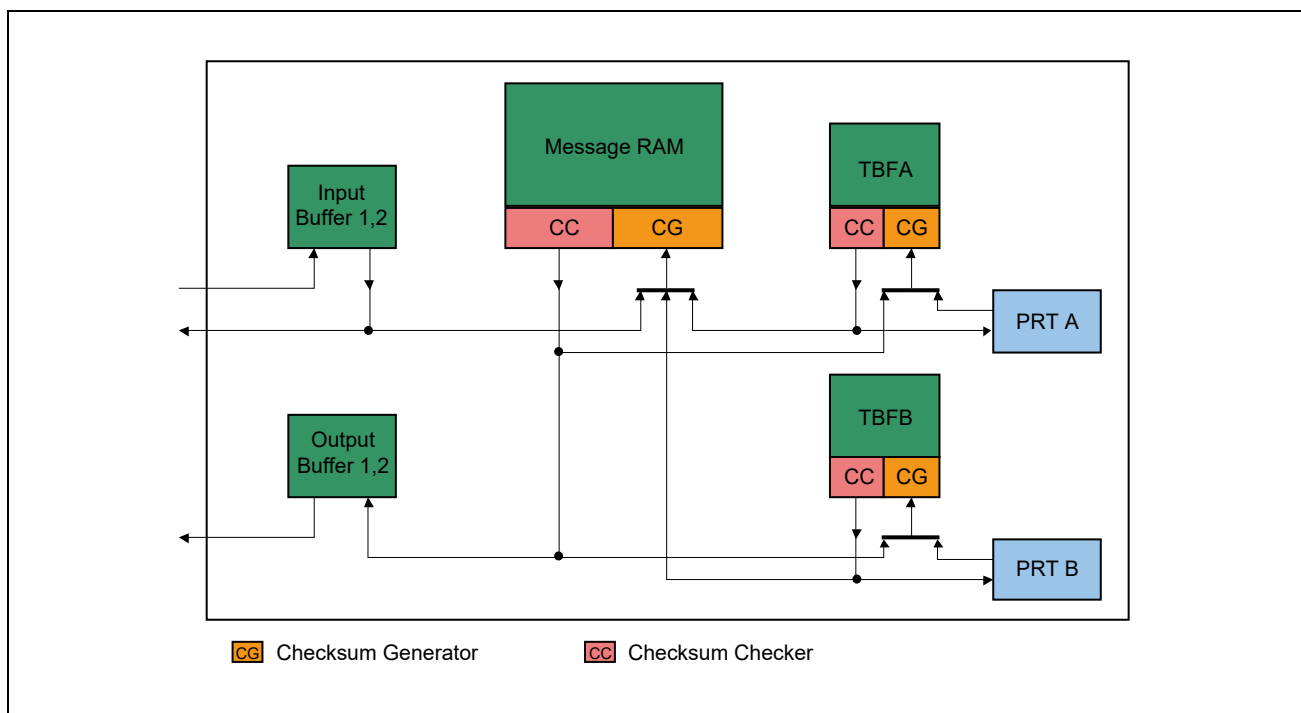


Figure 21.19 Checksum Generation and Check

When an access error has been detected the following actions will be performed:

In all cases:

- The respective access error flag in FLXAnFRMHDS register is set
- The access error flag FLXAnFREIR.AERR is set and, if enabled, a module interrupt to the host will be generated.

Additionally in specific cases:

- (1) Access error during data transfer from input buffer 1 or 2 to message RAM when reading header section of respective message buffer from message RAM:
 - FLXAnFRMHDS.AMR is set.
 - FLXAnFRMHDS.FMBD bit is set to indicate that there is an error in the message buffer to which FLXAnFRMHDS.FMB points.

- FLXAnFRMHDS.FMB indicates the number of the message buffer that has an error.
 - The data section of the corresponding message buffer is not updated.
 - Transmit buffer: Transmission request for the corresponding message buffer is not set.
- (2) Access error during scan of header sections in message RAM:
- FLXAnFRMHDS.AMR is set.
 - FLXAnFRMHDS.FMBD bit is set to indicate that there is an error in the message buffer to which FLXAnFRMHDS.FMB points.
 - FLXAnFRMHDS.FMB indicates the number of the message buffer that has an error.
 - The message buffer is ignored (skipped).
- (3) Access error during data transfer from message RAM to temporary buffer 1 or 2:
- FLXAnFRMHDS.AMR is set.
 - FLXAnFRMHDS.FMBD bit is set to indicate that there is an error in the message buffer to which FLXAnFRMHDS.FMB points.
 - FLXAnFRMHDS.FMB indicates the number of the message buffer that has an error.
 - Frame is not transmitted. After the frame CRC is set to 0, the frame that is being transmitted is invalidated.
- (4) Access error during data transfer from temporary buffer 1 or 2 to message RAM when reading header section of respective message buffer from message RAM:
- FLXAnFRMHDS.AMR is set.
 - FLXAnFRMHDS.FMBD bit is set to indicate that there is an error in the message buffer to which FLXAnFRMHDS.FMB points.
 - FLXAnFRMHDS.FMB indicates the number of the message buffer that has an error.
 - The data section of the corresponding message buffer is not updated.
- (5) Access error during data transfer from message RAM to output buffer:
- The access error flag FLXAnFRMHDS.AMR is set.
 - FLXAnFRMHDS.FMBD bit is set to indicate that there is an error in the message buffer to which FLXAnFRMHDS.FMB points.
 - FLXAnFRMHDS.FMB indicates the number of the message buffer that has an error.
- (6) Access error during a data transfer from temporary buffer 1 or 2 to protocol controller 1 or 2:
- FLXAnFRMHDS.ATBF1 and FLXAnFRMHDS.ATBF2 bits are set.
 - After the frame CRC is set to 0, the frame that is being transmitted is invalidated.

- (7) Access error during data transfer from temporary buffer 1 or 2 to message RAM when reading temporary buffer 1 or 2:
- FLXAnFRMHDS.ATBF1 and FLXAnFRMHDS.ATBF2 bits are set.
 - FLXAnFRMHDS.FMBD bit is set to indicate that there is an error in the message buffer to which FLXAnFRMHDS.FMB points.
 - FLXAnFRMHDS.FMB indicates the number of the message buffer that has an error.
- (8) Access error during data read of Transient Buffer RAM 1 or 2:
- When an access error occurs while the message handler read a frame with network management information (PPI = '1') from Transient Buffer RAM 1 or 2, the corresponding network management vector registers FLXAnFRNMV1 to 3 are not updated from this frame.

21.4.13.4 Host Handling of Access Errors

Access error caused by temporary bit flips can be fixed by:

(1) Self-healing

Access errors located in the data section of message RAM, Transient Buffer RAM A or Transient Buffer RAM B is overwritten with the next write access to the disturbed bit(s) caused by host access or by FlexRay communication.

(2) CLEAR_RAMs Command

The POC command CLEAR_RAMs initializes the message RAM to zero, when called in the DEFAULT_CONFIG or CONFIG state.

(3) Temporary Unlocking of Header Section

An access error in the header section of a locked message buffer can be fixed by a transfer from the input buffer to the locked buffer header section. For this transfer, the write access to the FLXAnFRIBCR (specifying the message buffer number) must be immediately preceded by the unlock sequence normally used to leave CONFIG state (see **Section 21.3.3.1, FLXAnFRLCK — FlexRay Lock Register**).

For that single transfer the message buffer header is unlocked, regardless of whether it belongs to the FIFO or whether its locking is controlled by FLXAnFRMRC.SEC, and will be updated with new data.

21.4.14 Interrupts

In general, interrupts provide a close link to the protocol timing as they are triggered almost immediately when an error or status change is detected by the CC, a frame is received or transmitted, a configured timer interrupt is activated, or a stop watch event occurred. This enables the host to react very quickly on specific error conditions, status changes, or timer events. On the other hand, too many interrupts can cause the host to miss deadlines required by the application. Therefore the CC supports enable and disable controls for each individual interrupt source separately.

An interrupt may be triggered when

- An error was detected
- A status flag is set to '1'
- A timer reaches a configured value
- A message transfer from input buffer to message RAM or from message ram to output buffer has completed
- A message transfer from the Local RAM/Cluster RAM to message RAM or from message RAM to Local RAM/Cluster RAM has completed
- A stop watch event occurred.

Tracking status and generating interrupts when a status change or an error occurs are two independent tasks. Regardless of whether an interrupt is enabled or not, the corresponding status is tracked and indicated by the CC. The host has access to the actual status and error information by reading registers FLXAnFREIR, FLXAnFRSIR, FLXAnFROS, FLXAnFROTS and FLXAnFRITS.

The general purpose interrupt lines to the host, FlexRay Interrupt 0, FlexRay Interrupt 1, are controlled by the enabled interrupts in FLXAnFREIES and FLXAnFRSIES. In addition each of the two interrupt lines can be enabled or disabled separately by programming bit FLXAnFRILE.EINT0 and FLXAnFRILE.EINT1.

The input data transfer interrupt lines to the host, INTFLXAnIQE, INTFLXAnIQF, are controlled by the enabled interrupts in FLXAnFRITS. In addition each of the input data transfer interrupts can be enabled or disabled separately by programming the related bits in FLXAnFRITC.

The output data transfer interrupt lines to the host, INTFLXAnFW, INTFLXAnOW, INTFLXAnFDA, INTFLXAnOT, are controlled by the enabled interrupts in FLXAnFROTS. In addition each of the output data transfer interrupts can be enabled or disabled separately by programming the related bits in FLXAnFROTC.

The three timer interrupts lines to the host are controlled by the enabled interrupts in FLXAnFROS. In addition, each of the interrupt lines can be enabled or disabled separately by programming bit FLXAnFROC.T0IE, FLXAnFROC.T1IE and FLXAnFROC.T2IE.

When a transfer between IBF/OBF and the message RAM has completed, bit FLXAnFRSIR.TIBC or FLXAnFRSIR.TOBC is set to '1'.

21.4.15 Assignment of FlexRay Configuration Parameters

Table 21.108 FlexRay Configuration Parameters (1/2)

Parameter	Bit(field)
pKeySlotUsedForStartup	FLXAnFRSUCC1.TXST
pKeySlotUsedForSync	FLXAnFRSUCC1.TXSY
gColdStartAttempts	FLXAnFRSUCC1.CSA
pAllowPassiveToActive	FLXAnFRSUCC1.PTA
pWakeupChannel	FLXAnFRSUCC1.WUCS
pSingleSlotEnabled	FLXAnFRSUCC1.TSM
pAllowHaltDueToClock	FLXAnFRSUCC1.HCSE
pChannels	FLXAnFRSUCC1.CCH
pdListenTimeOut	FLXAnFRSUCC2.LT
gListenNoise	FLXAnFRSUCC2.LTN
gMaxWithoutClockCorrectionPassive	FLXAnFRSUCC3.WCP
gMaxWithoutClockCorrectionFatal	FLXAnFRSUCC3.WCF
gNetworkManagementVectorLength	FLXAnFRNEMC.NML
gdTSSTransmitter	FLXAnFRPRTC1.TSST
gdCASRxLowMax	FLXAnFRPRTC1.CASM
gdSampleClockPeriod	FLXAnFRPRTC1.BRP
pSamplesPerMicrotick	FLXAnFRPRTC1.BRP
gdWakeupSymbolRxWindow	FLXAnFRPRTC1.RXW
pWakeupPattern	FLXAnFRPRTC1.RWP
gdWakeupSymbolRxIdle	FLXAnFRPRTC2.RXI
gdWakeupSymbolRxLow	FLXAnFRPRTC2.RXL
gdWakeupSymbolTxIdle	FLXAnFRPRTC2.TXI
gdWakeupSymbolTxLow	FLXAnFRPRTC2.TXL
gPayloadLengthStatic	FLXAnFRMHDC.SFDL
pLatestTx	FLXAnFRMHDC.SLT
pMicroPerCycle	FLXAnFRGTUC1.UT
gMacroPerCycle	FLXAnFRGTUC2.MPC
gSyncNodeMax	FLXAnFRGTUC2.SNM
pMicroInitialOffset[A]	FLXAnFRGTUC3.UIOA
pMicroInitialOffset[B]	FLXAnFRGTUC3.UIOB
pMacroInitialOffset[A]	FLXAnFRGTUC3.MIOA
pMacroInitialOffset[B]	FLXAnFRGTUC3.MIOB
gdNIT	FLXAnFRGTUC4.NIT
gOffsetCorrectionStart	FLXAnFRGTUC4.OCS
pDelayCompensation[A]	FLXAnFRGTUC5.DCA
pDelayCompensation[B]	FLXAnFRGTUC5.DCB
pClusterDriftDamping	FLXAnFRGTUC5.CDD
pDecodingCorrection	FLXAnFRGTUC5.DEC
pdAcceptedStartupRange	FLXAnFRGTUC6.ASR
pdMaxDrift	FLXAnFRGTUC6.MOD
gdStaticSlot	FLXAnFRGTUC7.SSL
gNumberOfStaticSlots	FLXAnFRGTUC7.NSS

Table 21.108 FlexRay Configuration Parameters (2/2)

Parameter	Bit(field)
gdMinislot	FLXAnFRGTUC8.MSL
gNumberOfMinislots	FLXAnFRGTUC8.NMS
gdActionPointOffset	FLXAnFRGTUC9.APO
gdMinislotActionPointOffset	FLXAnFRGTUC9.MAPO
gdDynamicSlotIdlePhase	FLXAnFRGTUC9.DSI
pOffsetCorrectionOut	FLXAnFRGTUC10.MOC
pRateCorrectionOut	FLXAnFRGTUC10.MRC
pExternOffsetCorrection	FLXAnFRGTUC11.EOC
pExternRateCorrection	FLXAnFRGTUC11.ERC

21.4.16 Usage of Data Transfer

A mechanism is implemented to allow storage of FlexRay messages directly into the Local RAM/Cluster RAM (user RAM) and have transfers between the FlexRay internal message RAM and the Local RAM/Cluster RAM and vice versa with minimum CPU support. The data in the Local RAM/Cluster RAM should be indexed by data structure pointers located in data pointer tables stored in the Local RAM/Cluster RAM.

Data transfer from the Local RAM/Cluster RAM to the FlexRay internal message RAM (input transfer) needs to be initiated by the application. These transfers can be used to configure message buffers or to update transmit data.

A data transfer from the FlexRay internal message RAM to the Local RAM/Cluster RAM (output transfer) is initiated automatically by a reception into a receive message buffer or FlexRay internal FIFO or by a change in the slot status. It can be initiated also by a specific user transfer request.

The input and output data transfer can be activated independently. When the input data transfer is activated the application should not directly access message buffers using the FlexRay input buffer. When the output data transfer is activated the application cannot directly access message buffers using the FlexRay output buffer.

21.4.16.1 Input Data Transfer

When the automatic input data transfer function is enabled, committed input data structures are transferred from the Local RAM/Cluster RAM to the FlexRay internal message RAM with minimum CPU support.

(1) Activation and Deactivation

The input data transfer function should be activated before usage. The activation of the input transfer handler initializes the input queue put index (FLXAnFRITS.IPIDX) and get index (FLXAnFRITS.IGIDX) to zero. In addition, set the interrupt status flags in the FLXAnFRITS register (IQEIS and IQFIS) to '0'.

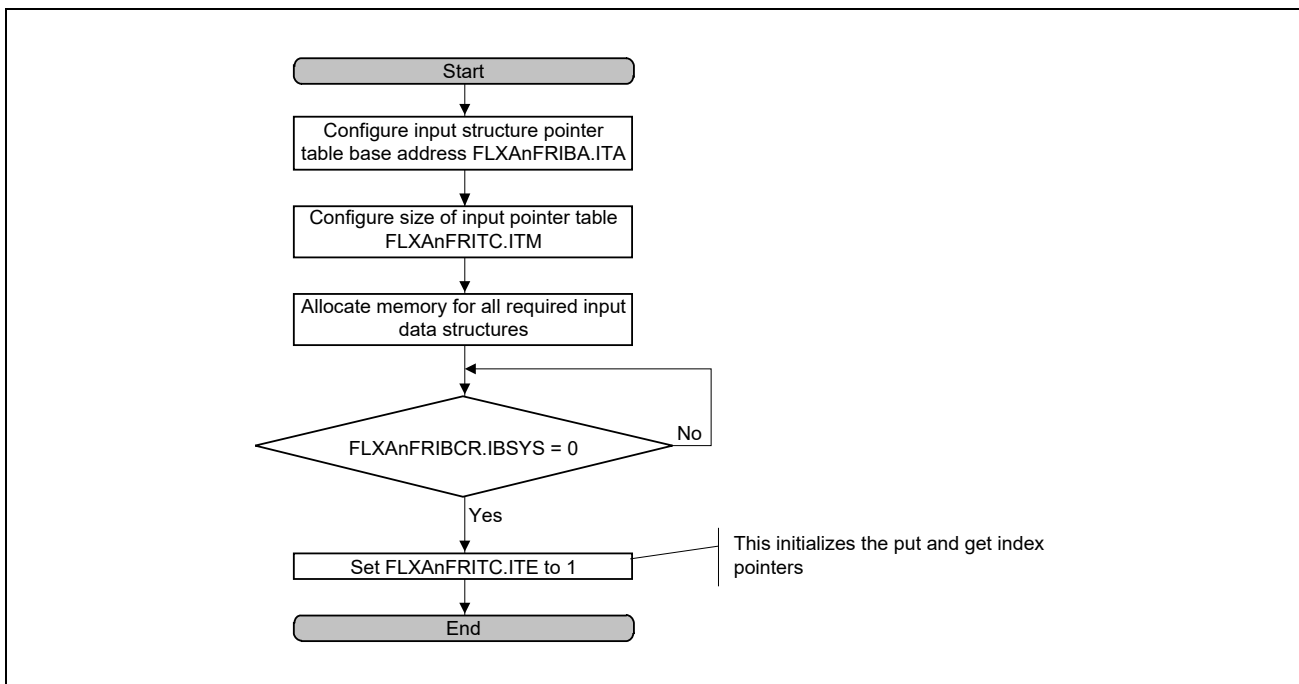


Figure 21.20 Input Transfer Enable Flow

A deactivation request of the input transfer function can be made at any time. The input queue put index and the input queue status are maintained independently from the input transfer function state.

Before the transfer function is disabled (status indicated by FLXAnFRITS.ITS), user requested input transfers and all committed input transfers will be completed.

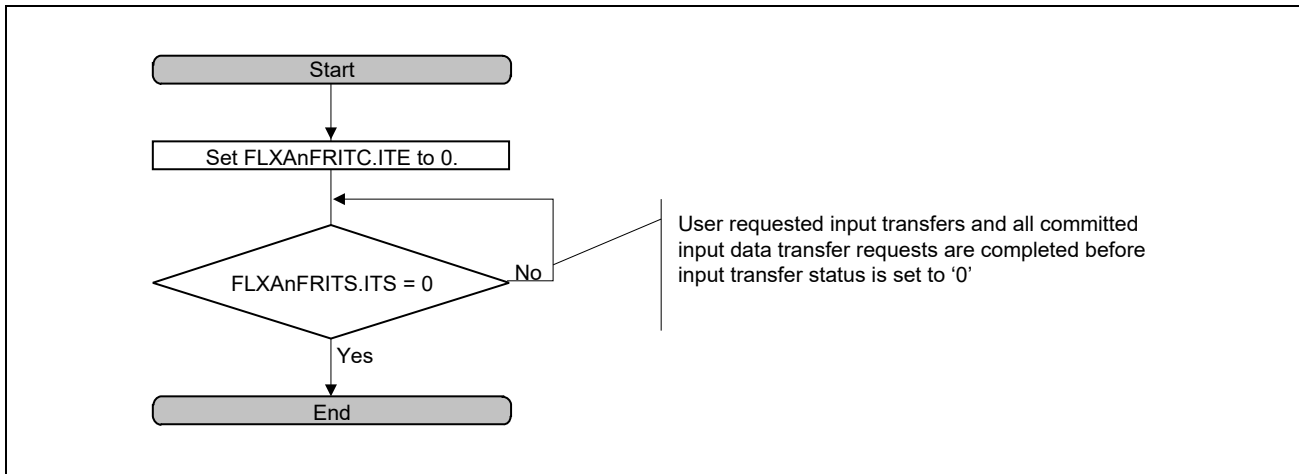


Figure 21.21 Input Transfer Disable Flow

(2) Input Data Structure

The application has to reserve a location in the Local RAM/Cluster RAM to provide the content for message buffer configuration (input data structure).

The location of this input data structure needs to also be defined by an input data structure pointer located in the Local RAM/Cluster RAM.

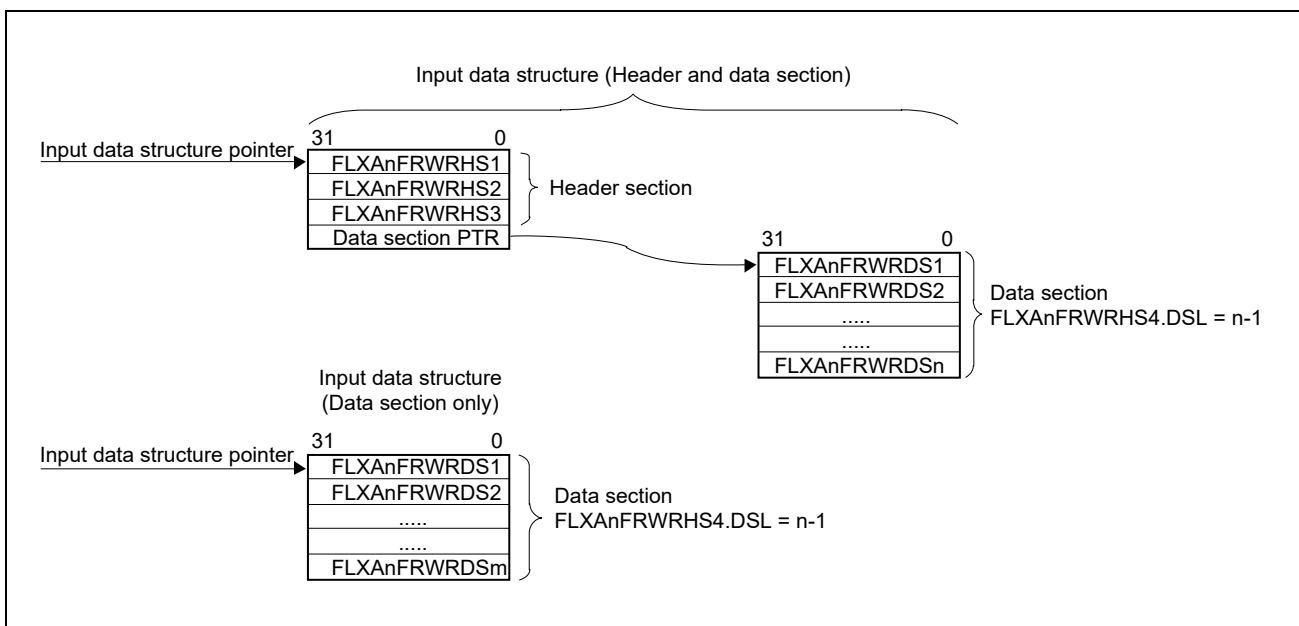


Figure 21.22 Input Data Structure

In general the input data structure consists of two sections, the header and the data section.

The header section consists of FLXAnFRWRHS1, FLXAnFRWRHS2, FLXAnFRWRHS3 and the data section pointer.

For bit alignment and bit function in the header section, see **Section 21.4.13.1, Header Partition**.

Depending on the settings in the control field (FLXAnFRWRHS4) located in the input pointer table, the data structure pointer is a reference to the address of FLXAnFRWRHS1 or FLXAnFRWRDS1. The data structure pointer has to be aligned to a 32 bit address.

If the bit LHS in the address related to FLXAnFRWRHS4 is set to '1' it is required to provide a valid header section. In this case, FLXAnFRWRHS1 is the first element of the data structure.

If the bit LHS in the address related to FLXAnFRWRHS4 is set to '0' a header section is not required. In this case FLXAnFRWRDS1 is the first element of the data structure.

If the bit LDS in the address related to FLXAnFRWRHS4 is set to '1' it is required to provide a valid data section. The pointer to the data section is a reference to the address of the first payload long word (FLXAnFRWRDS1) and has to be aligned to a 32 bit address.

If the bit LDS in the address related to FLXAnFRWRHS4 is set to '0' a data section is not required. The data section pointer is not evaluated by the input handler.

The byte order for the FlexRay payload data in the input data structure is determined by FLXAnFROC.BEC. For information about the payload data alignment within the data section, refer to **Section 21.4.13.1, Header Partition** and **Section 21.4.17, Byte Alignment**.

The length of the data section and the size to be allocated in the Local RAM/Cluster RAM depend on the configuration of the bits DSL in the address related to FLXAnFRWRHS4.

For the transfer into the FlexRay core internal message RAM the number of 16 bit words configured by FLXAnFRWRHS2.PLC is used. The application has to ensure that a proper number of data words are provided in the Local RAM/Cluster RAM. In case the buffer is configured by FLXAnFRWRHS2.PLC to hold an odd payload length, the application has to write zero to the last 16 bit of the payload section to ensure that the padding data is all zero.

(3) Input Pointer Table

To transfer data from the input data structures located in the Local RAM/Cluster RAM to the FlexRay internal message RAM, the related input data structure pointer and control field need to be added to the input pointer table which is located in the Local RAM/Cluster RAM.

The location of the first element of this table is identified by the input pointer table base address (FLXAnFRIBA.ITA). This base address has to be aligned to a 32 bit address.

The maximum number of input requests that can be queued is defined by the Input queue Table Max register (FLXAnFRITC.ITM).

Each Input pointer table entry requires two long words. The required address range of the input pointer table for the queued transfer requests can be calculated by

Input pointer table size (byte) = $((\text{FLXAnFRITC.ITM} + 1) \times D \times 2) \times 4$

Equation 1

The input pointer entry for the user requested input transfer should be added after the end of the input pointer table. The pointer table index related to this entry and hence the number to be written to FLXAnFRUIR.UIDX, is FLXAnFRITC.ITM+1. The address in the input pointer table related to the user requested input transfer (user input address) can be calculated by

$$\text{User input address} = \text{FLXAnFRIBA.ITA} + \text{Input pointer table size}$$

Equation 2

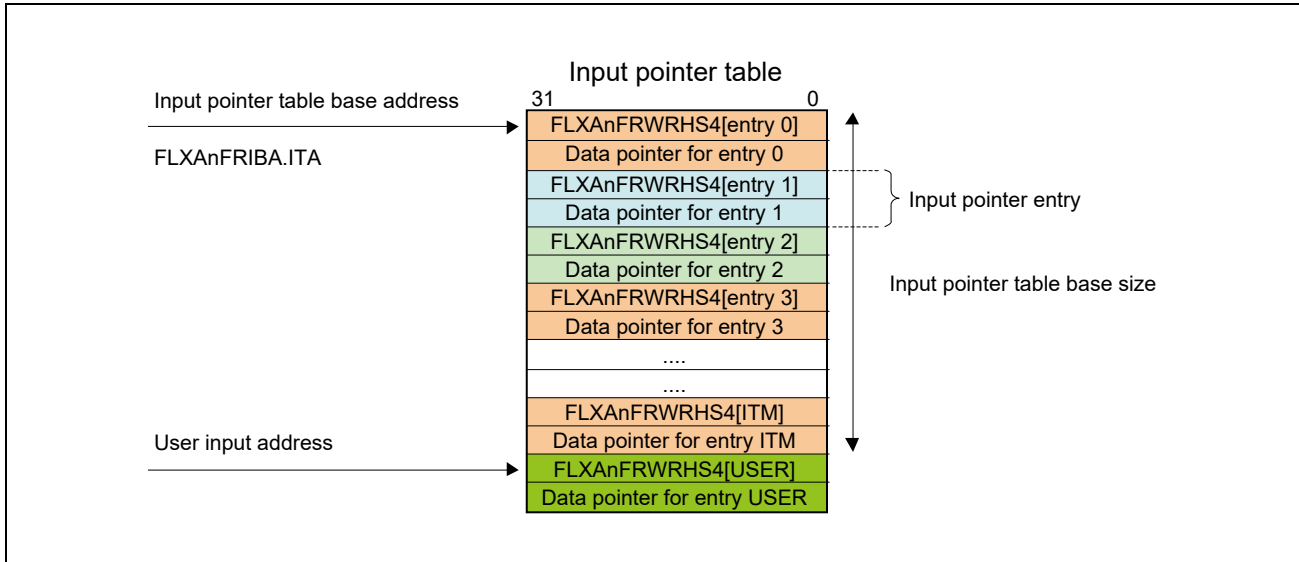


Figure 21.23 Input Pointer Table

The input pointer table holds the control field FLXAnFRWRHS4 and the pointers to the Local RAM/Cluster RAM location where the message buffer content (header section and/or data section) is stored.

The application has to write FLXAnFRWRHS4 and the input data structure pointer at the addresses in the input pointer table related to the put index position before a transfer request is initiated.

FLXAnFRWRHS4:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	DSL[5:0]					
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	INV	STR	LDS	LHS	—	IMBNR[6:0]						
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 21.109 FLXAnFRWRHS4 Register Contents

Bit Position	Bit Name	Function
31 to 22	—	Reserved The read value is undefined. When writing, always write 0.
21 to 16	DSL	Data Section Length Specifies the length of the data section in terms of 32 bit values.
15 to 12	—	Reserved The read value is undefined. When writing, always write 0.
11	INV	Invalidate Entry 0: The data structure is valid and will be transferred to the FlexRay internal message RAM. 1: The data structure is invalid. FlexRay internal message RAM is not updated using this input pointer entry.
10	STR	Set Transmission Request 0: The bit FLXAnFRTXRQx.TXR for the message buffer selected by the bits IMBNR is set to '0'. No data from this message buffer is transmitted. 1: The bit FLXAnFRTXRQx.TXR for the message buffer selected by the bits IMBNR is set to '1' to release the message buffer for transmission. The application should not set the bit STR to '1' for receive buffers.
9	LDS	Load Data Section 0: No update of data section 1: Data section for the message buffer selected by the bits IMBNR is updated.
8	LHS	Load Header Section 0: No update of header section 1: Header section for the message buffer selected by the bits IMBNR is updated.
7	—	Reserved The read value is undefined. When writing, always write 0.
6 to 0	IMBNR	Message Buffer Number Update Select the target message buffer number in the FlexRay internal message RAM for transfer.

Note that the LHS bit cannot be set for protected message buffers.

The bit LDS defines if the data section of the message buffer selected by the bits IMBNR is updated.

If LDS is set to '1', (DSL + 1) 32-bit words of payload data are transferred from the Local RAM/Cluster RAM to the message buffer selected by the bits IMBNR.

If LDS is set to '0', no payload data is transferred from the Local RAM/Cluster RAM.

Note that the payload transferred is independent from the configured payload length (bits PLC in the address related to FLXAnFRWRHS2).

The bit INV can be used to invalidate a transmitted data structure. This bit should be only used to cancel the transfer of committed data structures when the input queue is halted (see **Section 21.4.16.1(5), Halting the Input Queue**).

When this bit is set to '1', the message buffer number IMBNR is not updated. When the bit is set to '0', the message buffer number IMBNR is updated.

(4) Transfer Function of Input Data Structure

To use the input data structure transfer function, the input transfer has to be activated (see **Section 21.4.16.1(1), Activation and Deactivation**). The activation process requires the setup of the input pointer table (see **Section 21.4.16.1(3), Input Pointer Table**) in order to specify the source location (input data structures) for the data structures to be transferred. When the input transfer is enabled the get index pointer is initialized to zero.

All FlexRay internal message buffers can be updated using the input transfer queue which is built in the input pointer table. The application has to write the pointer and control field (table entry) to the data structure to be transferred into the input pointer table. For that purpose the application has to maintain a put index for the input pointer table that indicates where the pointer has to be written to.

To transmit this table entry to the input handler, the application has to write the target message buffer number to the input queue control register (FLXAnFRIQC.IMBNR). Afterwards the application increments the application internal put index.

By writing to the input queue control register the data available flag (FLXAnFRDAi.DA[IMBNR]) is automatically set to '1'. The input transfer handler also maintains the put index pointer in the status register (FLXAnFRITS.IPIDX).

In case the input queue is full (number of queued input transfer requests is equal to the input queue table size), FLXAnFRITS.IQFP and FLXAnFRITS.IQFIS are set to '1'. The input queue full condition pending flag (FLXAnFRITS.IQFP) changes from '1' to '0' when there are entries in the input queue available, whereby the input queue full interrupt status flag (FLXAnFRITS.IQFIS) needs to be cleared by the application.

The application cannot make any further write access to the bit IMBNR in the FLXAnFRIQC register as long as the bit IQFP in the FLXAnFRITS register is '1'.

In case the input queue becomes empty (number of queued input transfer requests changes to zero) FLXAnFRITS.IQEIS is set to '1'.

The input queue empty interrupt status flag (FLXAnFRITS.IQEIS) needs to be cleared by the application.

The transfer of the input data structures to the FlexRay message RAM is controlled by a get index pointer which is handled inside the FlexRay module and flagged in FLXAnFRITS.IGIDX. Note that the index is referring to the input entry, but not the address offset in the input pointer table.

If the input queue is not empty, the transfer handler reads the input pointer table entry of the transfer queue and starts the transfer of the input data structure from the address the input pointer is referring to. When all required data words are transferred to the FlexRay module, the data available flag for the transferred message buffer number is set to '0' and the get index in the transfer handler is incremented by one.

In case of an invalidated data structure (see **Section 21.4.16.1(5), Halting the Input Queue**) no FlexRay internal message buffer is updated and the related data available flag is automatically set to '0'. The change of the data available flag can be used to confirm the cancellation of a transmit request.

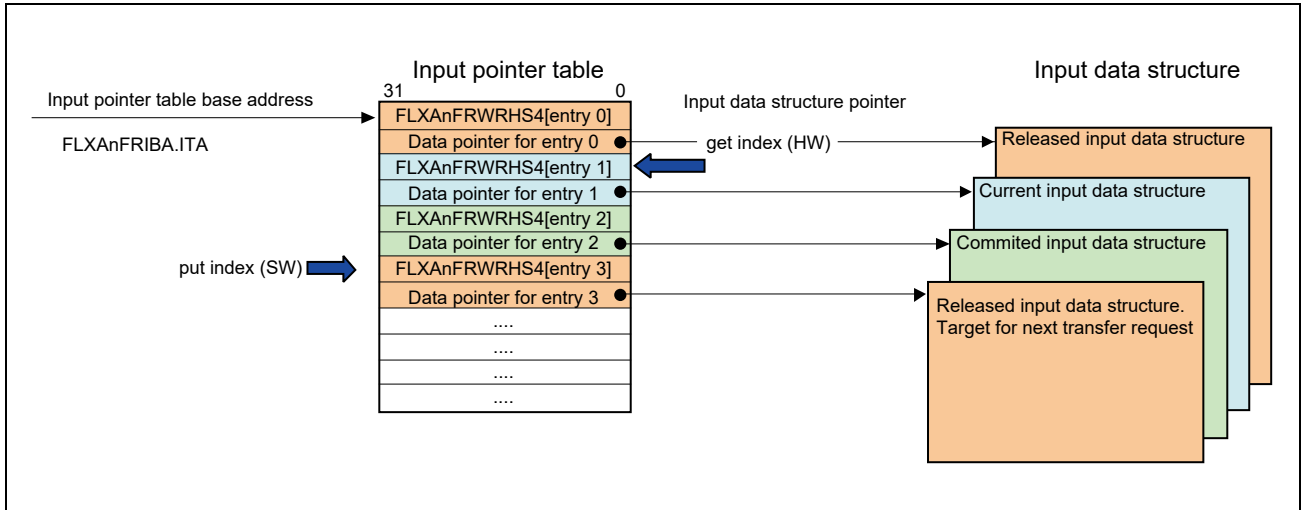


Figure 21.24 Input Pointer Table

Receive message buffers can be also configured using the input data transfer by setting up the required header sections and mark only the header section (`FLXAnFRWRHS4.LDS = '0'`, `FLXAnFRWRHS4.LHS = '1'`) to be updated in the FlexRay module.

(5) Halting the Input Queue

Committed data structures cannot be removed, but can be invalidated or updated when the input queue is halted.

To cancel data structures already committed to the input queue, the queue can be halted by writing '1' to FLXAnFRITC.IQHR.

After the ongoing input transfer has been completed, the queue is halted and FLXAnFRITS.IQH changes from '0' to '1'.

To invalidate an entry of the input queue, FLXAnFRWRHS4.INV has to be set to '1'. All other bits in FWRHS4 cannot be changed.

Following flow shall be used to analyze whether a transmitted message has been already transferred to the FlexRay internal message RAM or not.

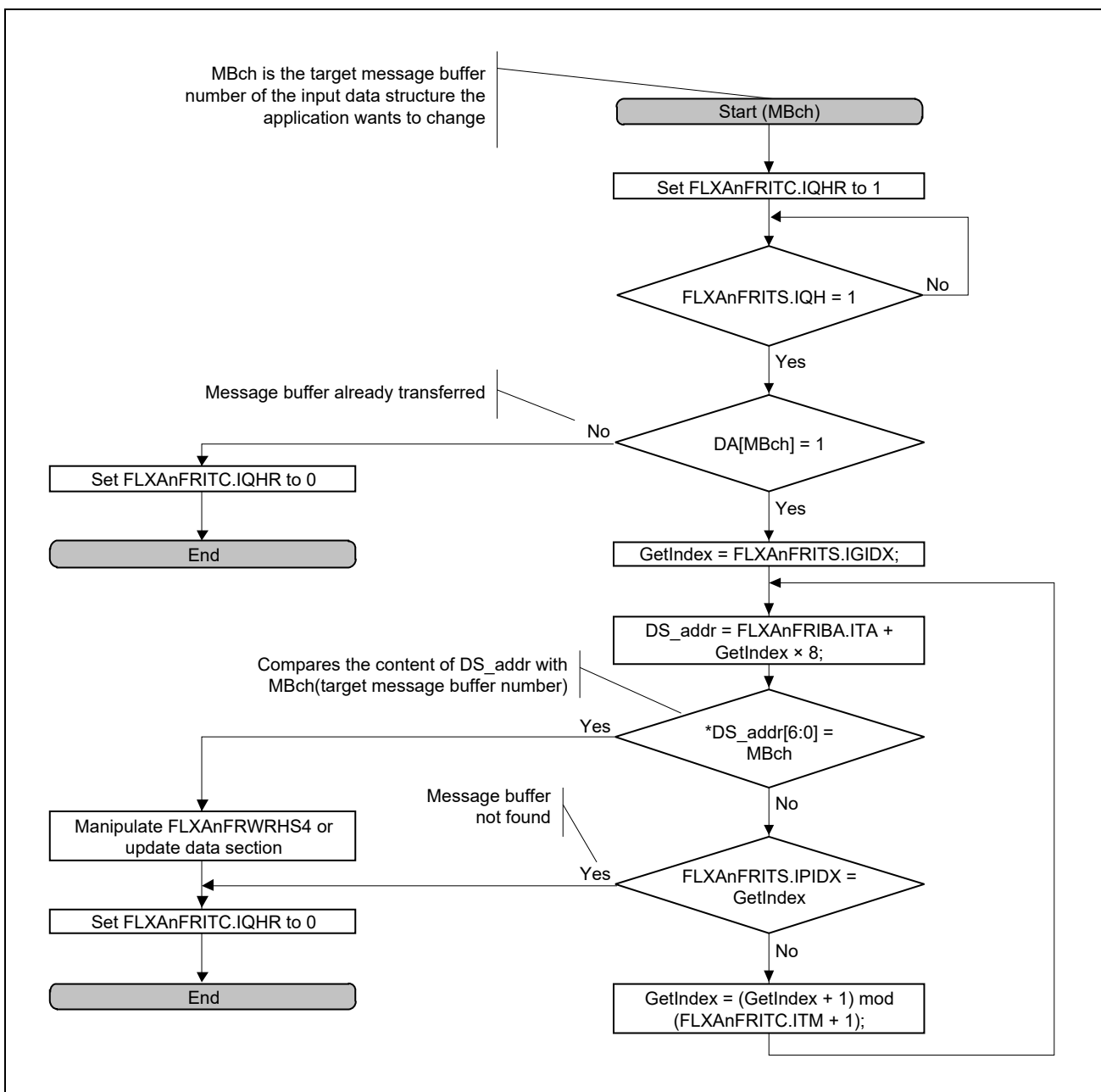


Figure 21.25 Input Table Analysis

In case the message buffer was already transferred to the FlexRay internal message RAM, the user input transfer request can be used to bypass the actual queue and update the required message buffer (see **Section 21.4.16.1(6), Transfer Function of User Requested Input Transfers**).

(6) Transfer Function of User Requested Input Transfers

To use this function, the input transfer has to be activated (see **Section 21.4.16.1(1), Activation and Deactivation**).

The application is capable, by using FLXAnFRUIR.UIDX, to request a transfer of an input data structure. The user input transfer request is executed first.

The application has to write the pointer and control field (table entry) to the data structure to be transferred into the input pointer table. Add the table entry for the user input transfer request after the end of the input pointer table (see **Section 21.4.16.1(3), Input Pointer Table**).

To transmit this table entry to the input handler, the application has to write the index (FLXAnFRITC.ITM + 1) to the user input transfer request register (FLXAnFRUIR.UIDX).

By writing to the user input transfer request register, the user input transfer request pending flag (FLXAnFRITS.UIRP) is automatically set to '1'.

As long as this flag is '1', the application cannot make any further user input transfer requests.

The user input transfer request pending flag (FLXAnFRITS.UIRP) changes from '1' to '0' when the requested input transfer is completed. Subsequently, the pending transfers are processed.

21.4.16.2 Output Data Transfer

When the output data transfer function is enabled, received messages (either in dedicated message buffers or in the FlexRay receive FIFO) are transferred to the Local RAM/Cluster RAM by the output data handler. The output data handler can also transfer the message buffer content to the Local RAM/Cluster RAM on application request. When enabled, the output handler is also capable of initiating a transfer when the message buffer status has changed.

(1) Activation and Deactivation

The output data transfer function should be activated before usage. The activation of the output transfer handler will initialize the FIFO put and get index pointer and FIFO fill level (FLXAnFROTS.FGIDX and FLXAnFROTS.FFL) to zero, set the bits FLXAnFROTS.FDA, FLXAnFROTS.OWP, FLXAnFROTS.FWP and FLXAnFROTS.UORP to '0'. Also the interrupt status flags FLXAnFROTS.OTIS, FLXAnFROTS.FIS, FLXAnFROTS.OWIS and FLXAnFROTS.FWIS) are set to '0'.

The activation has no effect on the data available flags (FLXAnFRDAi.DA) which are related to the dedicated buffers; these flags have to be cleared by the application.

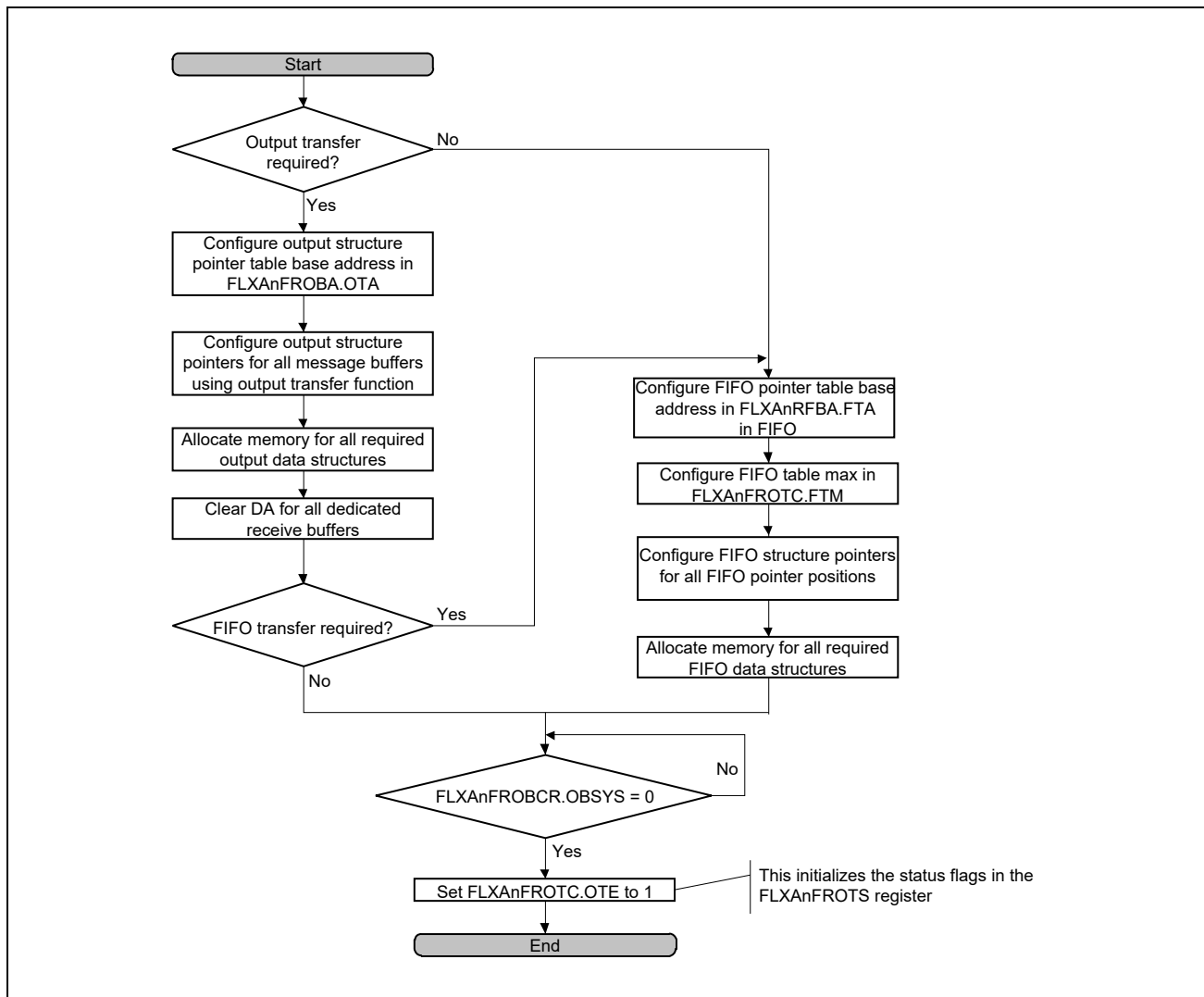


Figure 21.26 Output Transfer Enable Flow

A deactivation request of the output data transfer function can be made at any time. An ongoing transfer will be completed and the completion of this transfer will be flagged. During this time FLXAnFROTS.OTS remains '1'.

When FLXAnFROTS.OTS changes from '1' to '0', the output transfer function is deactivated. The data available status flags and the FIFO get index are still maintained when the output transfer function is disabled.

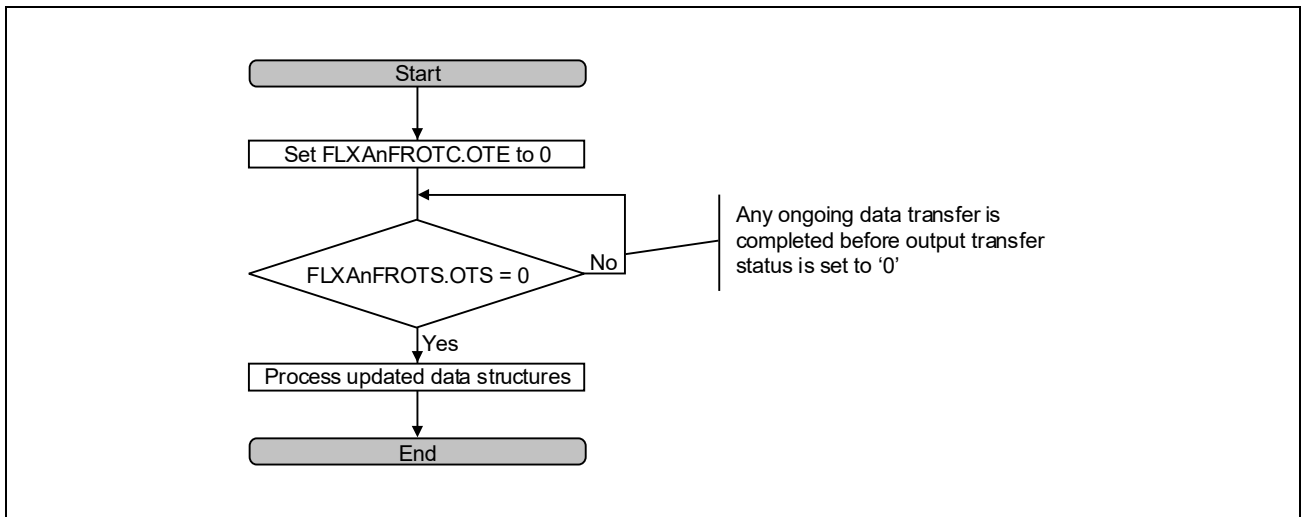


Figure 21.27 Output Transfer Disable Flow

(2) Output Transfer Data Structure

The data in the Local RAM/Cluster RAM is stored in an output data structure. The locations of the output data structures are determined by output data structure pointers also located in the Local RAM/Cluster RAM. The output data structure and indexing is visualized in **Figure 21.28**.

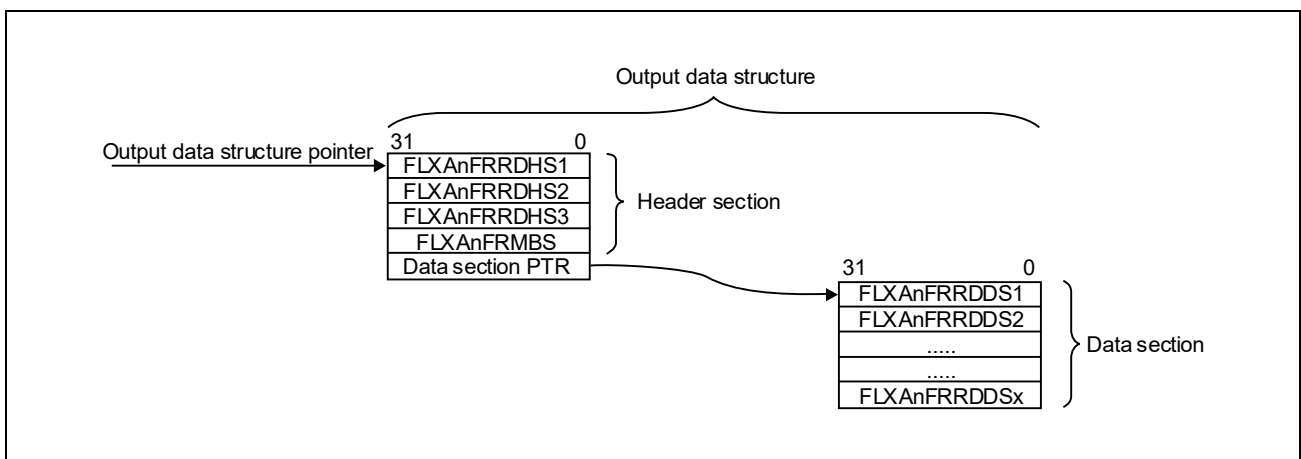


Figure 21.28 Output Data Structure

The output data structure consists of two sections, the header and data section. The header section consists of FLXAnFRRDHS1, FLXAnFRRDHS2, FLXAnFRRDHS3, FLXAnFRMBS and the data section pointer. FLXAnFRRDHS1 is the first element of the structure and has to be aligned to a 32 bit address. The data structure pointer is a reference to the address of FLXAnFRRDHS1. For information about the bit alignment and bit function within the header section, refer to **Section 21.4.13.1, Header Partition**.

FLXAnFRRDDS1 is the first element of the data section. The data section pointer is a reference to the address of FLXAnFRRDDS1 and has to be aligned to a 32 bit address. The byte order for the FlexRay payload data in the output data structure is determined by the bit BEC in the FLXAnFROC register. For information about the payload data alignment within the data section, refer to **Section 21.4.13.2, Data Partition**.

The length of the data section and the total structure size to be allocated in the Local RAM/Cluster RAM depend on the configured payload length (FLXAnFRRDHS2.PLC) of the related message buffer. In case the configured payload length is an odd number of words or the received payload length (FLXAnFRRDHS2.PLR) is shorter than the configured payload length, the remaining data words in the Local RAM/Cluster RAM are unused and cannot be used by the application.

The output data structure is unique for all three kinds of output transfers. In case only the header section is transferred, the data section pointer is not evaluated by the output handler and the data section remains unchanged.

(3) Output Pointer Table

For the output data transfer function, the application needs to set up an output pointer table in the Local RAM/Cluster RAM. The location of the first element of this table must be programmed into the output pointer table base address (FLXAnFROBA.OTA). This base address has to be aligned to a 32 bit address.

The size of the output pointer table is defined by the maximum of: the last configured dedicated message buffer and the highest message buffer number that will be used for the user output transfer request.

The output pointer table holds pointers (output data structure pointers) to the Local RAM/Cluster RAM location where a memory space is reserved for the target message buffer content (header section and data section).

There is a fixed linear relationship between the address of the entries in the output pointer table and the number of the related message buffers (see **Figure 21.29**) the output pointer table starts with the entry for message buffer number 0 at the address configured in FLXAnFROBA.OTA and continues in ascending order for each following message buffer number, by 32 bit aligned address (e.g. message buffer 1 at address OTA + 4, message buffer 2 at address OTA + 8, etc.) for all possible message buffers.

When a set ND bit is the only transfer condition (FLXAnFROTC.OTCS is set to 0), only message buffers configured as a dedicated receive buffer or that will be used for user output transfer requests need have valid pointer entries.

When a set ND bit or a set MBC bit is the transfer condition (FLXAnFROTC.OTCS is set to 1), all dedicated receive buffer and dedicated transmit buffers need to have valid pointer entries.

(4) FIFO Output Pointer Table

The FlexRay module internal FIFO can be extended by a queued buffer structure in the Local RAM/Cluster RAM.

If the FlexRay module internal FIFO is used, the application needs to set up the FIFO output pointer table. The location of the first element of this table is identified by the FIFO pointer table base address (FLXAnFRFBA.FTA). This base address has to be aligned to a 32 bit address.

The size of the FIFO pointer table and hence the maximum number of messages that can be added to the queue, is defined by FIFO Table Max (FLXAnFROTC.FTM).

The FIFO pointer table holds pointers (output data structure pointers) to the Local RAM/Cluster RAM location where a memory space is reserved for the target message buffer content (header section and data section). For each table entry, a data pointer shall be configured in this table.

(5) Transfer Function of Dedicated Message Buffers

To use this transfer function, the output transfer has to be activated (see **Section 21.4.16.2(1), Activation and Deactivation**). The activation process requires setting up the output pointer table (see **Section 21.4.16.2(3), Output Pointer Table**) in order to specify the destination location (output data structures) for the data to transfer. **Figure 21.29** shows how the output pointer table references the output data structures.

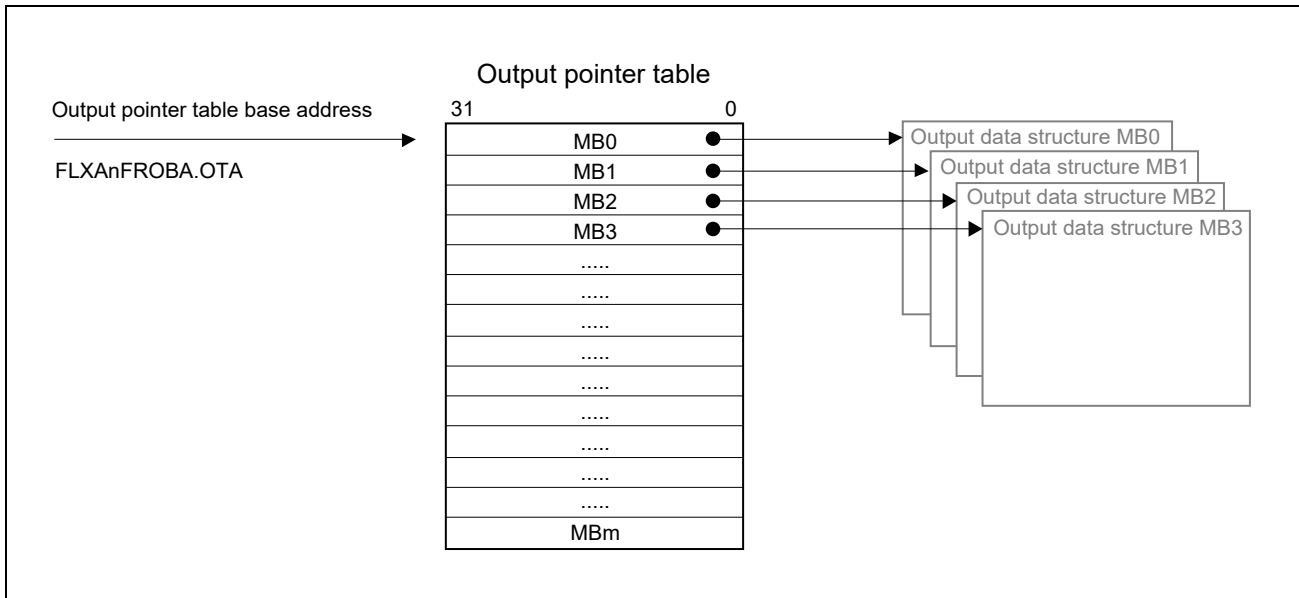


Figure 21.29 Output Data Structure and Indexing

With FLXAnFROTC.OTCS, the output transfer condition can be selected between ‘New data only mode’ and the ‘New data and status changed mode’.

In ‘New data only mode’, an output data transfer is initiated when a valid FlexRay data frame has been stored into a dedicated receive buffer that causes the related ND flag in the FLXAnFRNDAT_i register to be set. The ND flag in the FLXAnFRNDAT_i register is automatically set to ‘0’ during the transfer procedure. The header section is also transferred and hence the MBC flag in the FLXAnFRMBSC_i register is set to ‘0’.

In ‘New data and status changed mode’, an output data transfer is initiated as described in ‘New data only mode’. In addition, an output data transfer is initiated when only the message buffer status has been changed, which causes the related MBC flag in the FLXAnFRMBSC_i register to be set. In this case, only the header section is transferred. The MBC flag in the FLXAnFRMBSC_i register is automatically set to ‘0’ during the transfer procedure.

After transferring the message buffer data from the FlexRay internal message RAM to the output data structure, the corresponding data available flag in the FLXAnFRDA_i ($i = 0$ to 3) registers is set to ‘1’. The update of the output data structure is also flagged by the setting of the output transfer interrupt status flag (FLXAnFROTS.OTIS).

As long as the data available flag remains ‘1’, the corresponding output data structure will not be updated.

In the case

- The data available flag is ‘1’ and a valid received message was stored or
- When FLXAnFROTC.OTCS is ‘1’ and the message buffer status was updated,

the output transfer warning interrupt flag (FLXAnFROTS.OWIS) is set to '1', notifying the application that new data is available but the output data structure transfer cannot be processed. In addition, the FLXAnFROTS.OWP flag, which continuously indicates the status of the output transfer warning condition, is set to '1'.

If a valid receive message in the FlexRay internal message RAM is overwritten by an additional receive message, the message lost flag (FLXAnFRMBS.MLST) is set to '1'. This flag can be evaluated after the message buffer has been transferred into an output data structure.

The following sections describe how output data structures can be handled.

(a) Data Section Copy Method

One option is to copy the information from the output data structure to a different location of the Local RAM/Cluster RAM and then to release the output data structure by clearing the related data enable flag. The application can use the copied information for further processing.

(b) Data Structure Pointer Method

A different option is to modify the output data structure pointer in the output pointer table and to release the output data structure by clearing the related data enable flag. The changed output data pointer should refer to a free data structure. The application needs to use the old data structure for further processing.

(c) Data Section Pointer Method

A third option is to modify the data section pointer in the output data structure and to release the output data structure by clearing the related data enable flag. The changed data section pointer should refer to a free memory area. The application should use the old data section for further processing by forwarding the data section pointer.

(6) Transfer Function of FIFO Message Buffers

To use this buffer transfer function, the output transfer has to be activated (see **Section 21.4.16.2(1), Activation and Deactivation**). The activation process requires the setup of the FIFO pointer table (see **Section 21.4.16.2(4), FIFO Output Pointer Table**) in order to specify a location in the Local RAM/Cluster RAM reserved for the storage of the required output data structures.

A FIFO data transfer is initiated when a valid FlexRay data frame has been stored in the FlexRay internal FIFO.

After transfers from the internal FIFO to the output data structure, the FIFO interrupt status flag (FLXAnFROTS.FIS) and FLXAnFROTS.FDA are set to '1'. The bit FLXAnFROTS.FIS can be used as an interrupt source. The bit FLXAnFROTS.FDA indicates that the FIFO is not empty.

Up to FLXAnFROTS.FTM output structures configured in the register can be queued.

The transfer to the extended FIFO buffer structure is controlled by index pointers. This put index is controlled by the FIFO transfer handler and is incremented after transferring a message to the output data structure. The FIFO reception handler also maintains a get index which is flagged in FLXAnFROTS.FGIDX. The value of this get index is known by the application by either reading the status or maintaining a software variable. The get index (initially set to the zero) is incremented by one when the application releases the oldest entry of the FIFO queue by writing '1' to FLXAnFROTS.FDA. By comparing the put index and the get index, the FIFO handler acquires the current fill level of the queued buffer structure.

The current FIFO fill level is set in FLXAnFROTS.FFL. When FLXAnFROTS.FDA is '1', there is at least one entry in the FIFO queue.

In case the queued buffer structure in the Local RAM/Cluster RAM is full (FLXAnFROTS.FFL = FLXAnFROTC.FTM + 1), no further transfers are initialized, new messages remain in the FlexRay internal FIFO and the FIFO transfer warning interrupt status flag (FLXAnFROTS.FWIS) is set to '1'.

In case the FlexRay internal FIFO structure becomes full, messages in the FlexRay internal FIFO structure will be overwritten. The related status flags and configuration registers of the FlexRay core module are used to generate desired warning notifications.

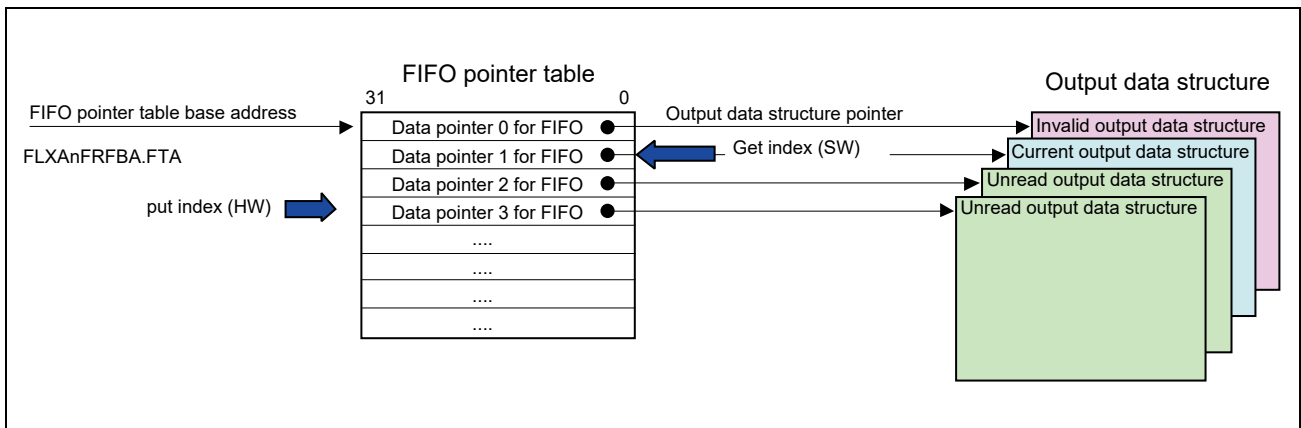


Figure 21.30 FIFO Pointer Table

(7) Transfer Function of User Output Transfer Requests

To use this transfer function, the output transfer has to be activated (**Section 21.4.16.2(1), Activation and Deactivation**). The activation process requires to set up the output pointer table (see **Section 21.4.16.2(4), FIFO Output Pointer Table**) in order to specify the location in the Local RAM/Cluster RAM reserved for the transfer of the data (output data structures).

The application can, by using FLXAnFRUOR.UMBNR, request a transfer of dedicated message buffer to an output data structure. Except in CONFIG state, message buffers which are part of the FlexRay internal FIFO cannot be requested.

The header section is always transferred to the output data structure. The transfer of the data section can be enabled by setting FLXAnFRUOR.URDS to '1'. The selected message buffer content is stored in the output data structure location determined by the pointers in the output pointer table.

The data available status and transfer blocking by the DA bits in the FLXAnFRDAi register is also used for the user requested transfers. Therefore, the DA bit in the FLXAnFRDAi register related to the requested buffer number (FLXAnFRUOR.UMBNR) needs to be released before the transfer request is made.

After writing to FLXAnFRUOR.UMBNR, the bit FLXAnFROTS.UORP is set to '1' to indicate that there is a pending user transfer request. When the transfer has been processed, the bit FLXAnFROTS.UORP is set to '0', the bit FLXAnFROTS.OTIS is set to '1' and the DA bit in the FLXAnFRDAi register related to the requested buffer number (FLXAnFRUOR.UMBNR) is set to '1'.

User output transfer requests cannot be queued. The application needs to check the bit FLXAnFROTS.UORP before writing to FLXAnFRUOR.UMBNR.

User output transfer requests cannot be made for message buffers that are pending in the input transfer queue.

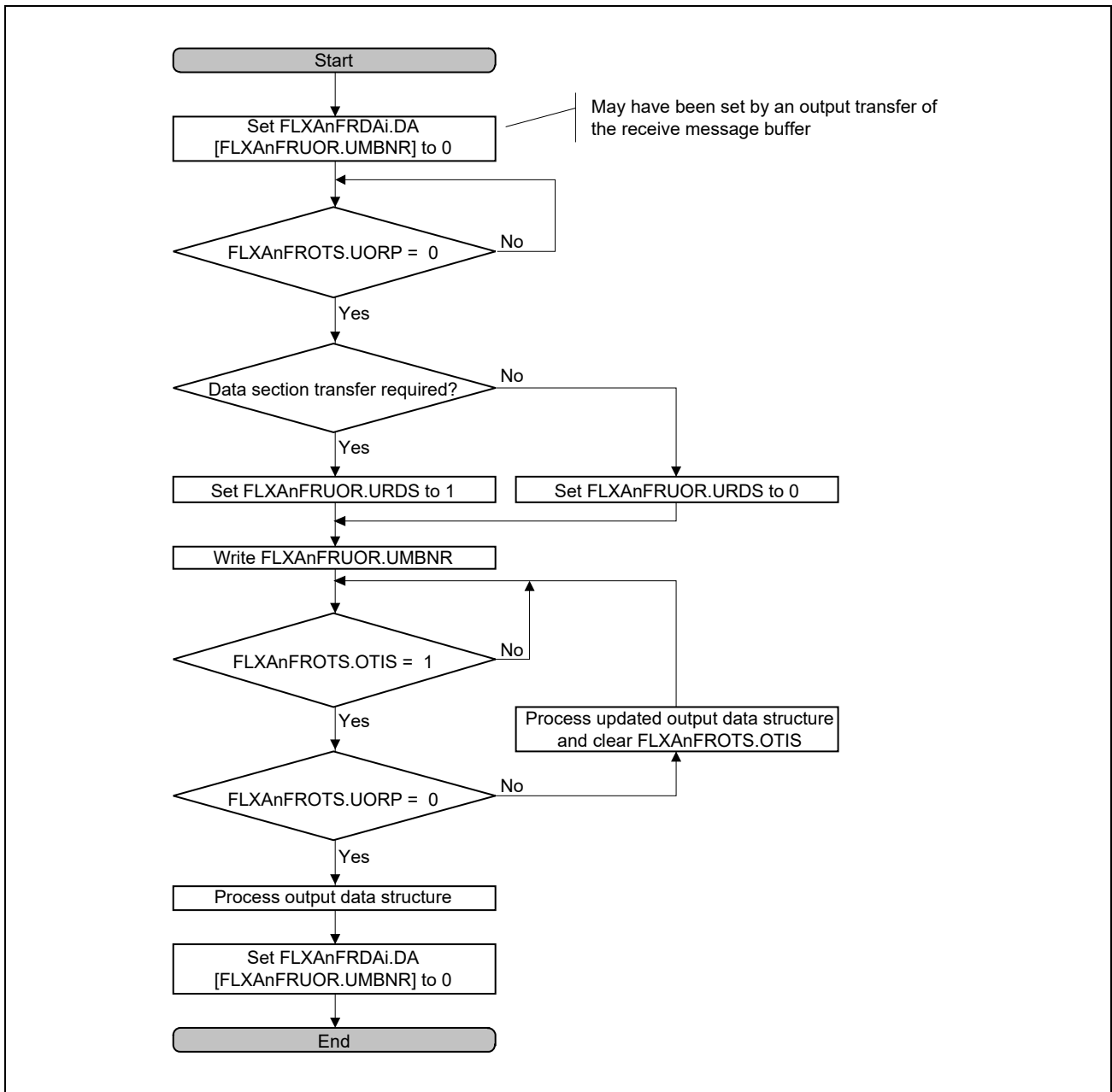


Figure 21.31 User Output Transfer Request Flow

Note that it may be possible that the data structure addressed by the user request is being updated due to a receive message buffer update (which causes the bit DA in the FLXAnFRDAi register to be set). This DA flag setting inhibits the user output transfer request. Therefore, polling FLXAnFROTS.UORP is not a secure method to identify when the transfer of a requested message buffer has been completed. The bits FLXAnFROTS.OTIS or the DA bit in the FLXAnFRDAi register can be used instead. The exact flow depends on the software architecture.

21.4.16.3 Data Structure Transfer Scheduling

The different types of transfer requests are checked cyclically. In order to guarantee a certain transfer time, the different types of transfers have different priorities.

Use requested input transfers that have highest priority followed by the transfer of data structures to be transmitted into the active input transfer queue. No new output transfer will be started as long as there is a pending input transfer request.

The three output transfer request types are checked in a specific order:

(1) All Dedicated Message Buffers in Ascending Order

When FLXAnFROTC.OTCS is set to '0', set flags in the FLXAnFRNDATi register are causing a transfer of the message buffer to the output data structure if the destination area is free (DA bit in the FLXAnFRDAi register is '0').

When FLXAnFROTC.OTCS is set to '1', set flags in the FLXAnFRNDATi register or set flags in the FLXAnFRMBSCi register are causing a transfer of the message buffer to the output data structure if the destination area is free (DA bit in the FLXAnFRDAi register is '0').

(2) Flexray Internal FIFO

When the FlexRay internal FIFO is not empty and there is a free destination area, one FIFO message is transferred into the output data structure addressed by the FIFO pointer table.

(3) User Output Request

If there is a pending user output transfer request, one message buffer is transferred into the corresponding output data structure.

The check sequence is suspended when an input transfer is generated.

21.4.16.4 Behavior in Case of Data Transfer Access Error

The memory areas accessed by the data transfer function may be protected by a memory protection unit (MPU). When the MPU flags an access to a protected address caused by an input or output transfer, an access error event is generated and the related bit in the FLXAnFRAES register is set.

The ongoing transfer is immediately terminated but completed transfers are processed and may generate further access errors. Any following access errors are only flagged in FLXAnFRAES.MAE. The other status flags are not updated.

(1) Access Error During Input Transfer

When an access error occurs during an input transfer:

- The ongoing transfer is immediately terminated. The FlexRay internal message RAM will not be updated
- The address the FlexRay module attempted to access is written in the FLXAnFRAEA register
- FLXAnFRAES.IAE is set to '1'
- The input pointer table index is flagged in FLXAnFRAES.EIDX
- In case of a normal input transfer the transfer related DA bit in the FLXAnFRDAi register is set to '0'
- In case of the user input transfer request FLXAnFRITS.UIRP is set to '0'

With the given status information, the application is able to identify and correct the faulty data structure.

In addition the application needs to clear the input access error flag (FLXAnFRAES.IAE).

(2) Access Error During Output Transfer

When an access error occurs during an output transfer:

- The ongoing transfer is immediately terminated but the update of the data structure may be started.
- The address that the FlexRay module intended to access is written in the FLXAnFRAEA register
- FLXAnFRAES.OAE is set to '1'
- The output pointer table index is flagged in FLXAnFRAES.EIDX
- In case of a normal output transfer, the related DA flag in the FLXAnFRDAi register remains '0' and no output transfer interrupt is generated
- In case of the user output transfer request, FLXAnFROTS.UORP is set to '0'

With the given status information, the application is able to identify and correct the faulty data structure. The data structure in the Local RAM/Cluster RAM cannot be treated as valid.

In addition the application needs to clear the output access error flag (FLXAnFRAES.OAE).

The FlexRay module internal transfer of the message buffer is completed before the Local RAM/Cluster RAM access error is detected. The output transfer cannot be resumed. To avoid loss of data, the application can perform the user output transfer request of this message buffer to a correct Local RAM/Cluster RAM location.

(3) Access Error During FIFO Transfer

When an access error occurs during an FIFO transfer:

- The ongoing transfer is immediately terminated
- The address, the FlexRay module intended to access to, is written in the FLXAnFRAEA register
- FLXAnFRAES.FAE is set to '1'
- The FIFO pointer table index is written in FLXAnFRAES.EIDX
- The FIFO index pointer is not changed and hence the FIFO status flags are unchanged

With the given status information, the application is able to identify and correct the faulty data structure.

In addition, the application needs to clear the FIFO access error flag (FLXAnFRAES.FAE).

The data in the Local RAM/Cluster RAM cannot be treated as valid and is not released to the application. The message cannot be recovered.

21.4.16.5 Behaviors in Case of RAM Read Errors

The FlexRay internal message RAM has an ECC checking mechanism. In case an uncorrectable RAM read error occurs, the application has to analyze the status in the FLXAnFRMHDS register and re-execute as described in **Section 21.4.16.3, Data Structure Transfer Scheduling**. The input and output transfer handler re-executes also on these errors detected in the message RAM when the error is related to an active transfer.

In addition, Transient Buffer RAM A and Transient Buffer RAM B have an ECC checking mechanism as well.

An uncorrectable RAM read errors does not affect the data transfer functionality but have to be handled as described in **Section 21.4.13.1(4), Message Buffer Status FLXAnFRMBS (word 3)**.

In all cases, data causing a read error is never transferred to the Local RAM/Cluster RAM. If there is no recovery available in the application, the message is lost.

(1) Read Error during Transfer from TBF to MBF

This internal transfer is performed for each valid FlexRay message received.

A read error can only occur when the header section in the FlexRay message RAM (see read error flags in FLXAnFRMHDS) is read. In this case, the message buffer needs to be re-configured.

For dedicated receive message buffers, the related ND flag in the FLXAnFRNDATi register will not be set. Consequently the affected message buffer will not be transferred to the output data structure.

For the FlexRay internal FIFO buffers, the ND flag in the FLXAnFRNDATi register is not set, but the FlexRay internal FIFO put index is incremented. Due to this, a transfer procedure from the FlexRay internal FIFO buffer to the output buffer is started. However, if the read error is still present in the header section, updating of the output data structure will not start (see **Section 21.4.16.5(2), Read Error during Transfer from MBF to OBF**); thus the data in the Local RAM/Cluster RAM remains correct.

Note that the correction or any other reconfiguration of FIFO related to message buffers, while there are pending FIFO transfers, may result in incorrect data in the Local RAM/Cluster RAM. It is strongly recommended to deactivate the output data transfer before starting the reconfiguration and flush the FlexRay internal FIFO before reactivation of the output data transfer.

(2) Read Error during Transfer from MBF to OBF

This internal transfer is performed for every output data transfer (dedicated reception, FIFO, user requested).

A read error can occur in the header and data section (see read error flags in FLXAnFRMHDS). In both cases, the message gets lost. If the error is located in the header section, the message buffer needs to be re-configured. If the error is located in the data section, the error is corrected with the next data section update.

When a read error occurs during the transfer from the message RAM to the output buffer, the output data structure will not be updated and the data enable bit will not be set to '1'. The FIFO put index and the FIFO fill level are not changed also. In case of user output transfer request, FLXAnFROTS.UORP is set to '0' even if there was no update of the output data structure.

(3) Read Error during Transfer from IBF to MBF

This internal transfer is performed for every input data transfer.

A read error can occur only when the header section requested (the bit LHS in FLXAnFRWRHS4 is set to “0”) cannot be updated due to the reading of the header section from the message RAM (see read error flags in FLXAnFRMHDS). In this case, the message buffer needs to be re-configured.

When a read error occurs during the input data transfer, the actually transferred message in the input queue is lost.

(4) Message RAM Read Errors

Read errors when reading the header section are flagged in the FLXAnFRMHDS register.

Depending on the buffer type and set buffer protection, the message buffer may not be reconfigured.

The input transfer function cannot be used to reconfigure a locked message buffer using the method described in **Section 21.4.13.4(3), Temporary Unlocking of Header Section.**

Before reconfiguring a locked buffer, the user needs to disable the input transfer function and the output transfer function.

21.4.17 Byte Alignment

The alignment of the bytes received by the FlexRay protocol and the alignment of the bytes required by the application may be different. The FlexRay module provides with FLXAnFROC.BEC a byte alignment function to support different byte ordering styles.

Figure 21.32 shows the payload byte alignment in a FlexRay frame.

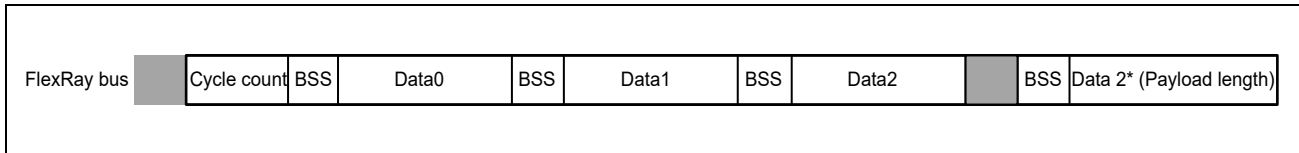


Figure 21.32 Byte Alignment on the FlexRay Bus

21.4.17.1 Little Endian Alignment

When FLXAnFROC.BEC is '0', the byte alignment is set to Little Endian.

(1) FLXAnFRNMVm (m= 1 to 3)

The byte alignment of the NMV bytes is as follows.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXAn FRNMV1	Data 3								Data 2								Data 1								Data 0							
FLXAn FRNMV2	Data 7								Data 6								Data 5								Data 4							
FLXAn FRNMV3	Data 11								Data 10								Data 9								Data 8							

(2) FLXAnFRWRDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay input buffer and the input data structure is as follows.

- FLXAnFRWRDSx.MD[7:0] = Data_{4x-4}
- FLXAnFRWRDSx.MD[15:8] = Data_{4x-3}
- FLXAnFRWRDSx.MD[23:16] = Data_{4x-2}
- FLXAnFRWRDSx.MD[31:24] = Data_{4x-1}

Transmission order on the FlexRay bus is FLXAnFRWRDSx.MD[7:0], FLXAnFRWRDSx.MD [15:8], FLXAnFRWRDSx.MD [23:16], FLXAnFRWRDSx.MD [31:24] with the most significant bit (MSB) transmitted first.

(3) FLXAnFRRDDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay output buffer and the output data structure are as follows.

- FLXAnFRRDDSx.MD[7:0] = Data_{4x-4}
- FLXAnFRRDDSx.MD[15:8] = Data_{4x-3}
- FLXAnFRRDDSx.MD[23:16] = Data_{4x-2}
- FLXAnFRRDDSx.MD[31:24] = Data_{4x-1}

Reception order on the FlexRay bus is FLXAnFRRDDSx.MD[7:0], FLXAnFRRDDSx.MD[15:8], FLXAnFRRDDSx.MD[23:16], FLXAnFRRDDSx.MD[31:24] with the most significant bit (MSB) transmitted first.

21.4.17.2 Big Endian Alignment

When FLXAnFROC.BEC is '1', the byte alignment is set to Big Endian.

(1) FLXAnFRNMVm (m = 1 to 3)

The byte alignment of the NMV bytes is as follows.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXAn FRNMV1	Data 0								Data 1								Data 2								Data 3							
FLXAn FRNMV2	Data 4								Data 5								Data 6								Data 7							
FLXAn FRNMV3	Data 8								Data 9								Data 10								Data 11							

(2) FLXAnFRWRDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay input buffer and the input data structure are as follows.

- FLXAnFRWRDSx.MD[7:0] = Data_{4x-1}
- FLXAnFRWRDSx.MD[15:8] = Data_{4x-2}
- FLXAnFRWRDSx.MD[23:16] = Data_{4x-3}
- FLXAnFRWRDSx.MD[31:24] = Data_{4x-4}

Transmission order on the FlexRay bus is FLXAnFRWRDSx.MD[31:24], FLXAnFRWRDSx.MD[23:16], FLXAnFRWRDSx.MD[15:8], and FLXAnFRWRDSx.MD[7:0] with the most significant bit (MSB) transmitted first.

(3) FLXAnFRRDDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay output buffer and the output data structure are as follows.

- FLXAnFRRDDSx.MD[7:0] = Data_{4x-1}
- FLXAnFRRDDSx.MD[15:8] = Data_{4x-2}
- FLXAnFRRDDSx.MD[23:16] = Data_{4x-3}
- FLXAnFRRDDSx.MD[31:24] = Data_{4x-4}

Reception order on the FlexRay bus is FLXAnFRRDDSx.MD[7:0], FLXAnFRRDDSx.MD [15:8], FLXAnFRRDDSx.MD [23:16], FLXAnFRRDDSx.MD [31:24] with the most significant bit (MSB) transmitted first.

Section 22 Renesas High Speed Bus (RHSB)

This section contains a generic description of the Renesas high speed bus (RHSB).

The first part in this section describes the features specific to this microcontroller including register base addresses and input/output signals. The ensuing sections describe the functions and registers of RHSB.

22.1 Features of RHSB

22.1.1 Number of Units

This microcontroller has the following number of Renesas high speed bus (RHSB) units.

Table 22.1 Number of Units

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Units	2	2
Unit Name	RHSBj (j = 0,1)	RHSBj (j = 0,1)

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Units	2	2
Unit Name	RHSBj (j = 0,1)	RHSBj (j = 0,1)

Table 22.2 Indices

Index	Description
j	The individual RHSB units are identified by the index "j" (j = 0, 1).
x	The individual chip select signals are identified by the index "x" (x = 0, 1).
y	The individual upstream serial input signals are identified by the index "y" (y = 0, 1).
k	The individual sub XBARs are identified by the index "k" (k = 0, 1, 2, 3).

Table 22.3 RHSB Chip Select Index

RHSB Instance	RHSB Chip Select Index
RHSB0	RHSB0CSDx (x = 0, 1)
RHSB1	RHSB1CSDx (x = 0, 1)

Table 22.4 RHSB Upstream Serial Input Index

RHSB Channels	RHSB Upstream Serial Input Index
RHSB0	RHSB0Sly (y = 0, 1)
RHSB1	RHSB1Sly (y = 0, 1)

22.1.2 Register Base Addresses

The RHSB base addresses are listed in the following table.

The RHSB register addresses are given as offsets from the base address.

Table 22.5 Register Base Addresses

RHSBj	RHSBj_Base	Bus Group
<RHSB0_base>	FFEE 0000 _H	Peripheral Group 6
<RHSB1_base>	FFEE 1000 _H	Peripheral Group 6

22.1.3 Clock Supply

The RHSB clock supply is shown in the following table.

Table 22.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
RHSBj	PCLK	CLK_LSB
	clkc	CLK_RHSB

22.1.4 Interrupt Requests

The RHSB interrupt requests are listed in the following table.

Table 22.7 Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number	
				sDMA	DTS
RHSB0					
INTRHSB0L0	RHSB0DTSF / RHSB0DTF	Data frame transmission started Data frame transmission complete	467	—	—
INTRHSB0L1	RHSB0CTF / RHSB0TSF	Command frame transmission complete Transmission started	468	—	—
INTRHSB0L2	RHSB0ETF	Emergency frame transmission complete	469	—	—
INTRHSB0L3	RHSB0DRF	Data receive	470	—	—
INTRHSB0L4	RHSB0UEF / RHSB0TOF / RHSB0DLF	Upstream error Timeout detected Data lost	471	—	—
INTRHSB0DWNDATADMA	RHSB0DTSF	Data frame transmission started	—	group0-189	group1-97
INTRHSB0DWNCMDDMA	RHSB0CTF	Command frame transmission complete	—	group0-191	group1-99
INTRHSB0UPDMA	RHSB0DRF	Data receive	—	group0-190	group1-98
RHSB1					
INTRHSB1L0	RHSB1DTSF / RHSB1DTF	Data frame transmission started Data frame transmission complete	472	—	—
INTRHSB1L1	RHSB1CTF / RHSB1TSF	Command frame transmission complete Transmission started	473	—	—
INTRHSB1L2	RHSB1ETF	Emergency frame transmission complete	474	—	—
INTRHSB1L3	RHSB1DRF	Data receive	475	—	—
INTRHSB1L4	RHSB1UEF / RHSB1TOF / RHSB1DLF	Upstream error Timeout detected Data lost	476	—	—
INTRHSB1DWNDATADMA	RHSB1DTSF	Data frame transmission started	—	group0-192	group1-100
INTRHSB1DWNCMDDMA	RHSB1CTF	Command frame transmission complete	—	group0-194	group1-102
INTRHSB1UPDMA	RHSB1DRF	Data receive	—	group0-193	group1-101

22.1.5 Reset Sources

The RHSB reset sources are listed in the following table. RHSB is initialized by these reset sources.

Table 22.8 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
RHSB0	All registers	✓	✓	✓	✓	—	—
RHSB1	All registers	✓	✓	✓	✓	—	—

22.1.6 External Input/Output Signals

The external input/output signals of RHSB are listed below.

Table 22.9 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
RHSB0		
RHSB0FCLP	RHSB0 downstream serial clock differential output	RHSB0FCLP
RHSB0FCLN	RHSB0 downstream serial clock differential output	RHSB0FCLN
RHSB0SOP	RHSB0 downstream serial data differential output	RHSB0SOP
RHSB0SON	RHSB0 downstream serial data differential output	RHSB0SON
RHSB0CSD0	RHSB0 Slave 0 downstream chip selected output	RHSB0CSD0
RHSB0CSD1	RHSB0 Slave 1 downstream chip selected output	RHSB0CSD1
RHSB0SI0	RHSB0 Slave 0 upstream serial input	RHSB0SI0
RHSB0SI1	RHSB0 Slave 1 upstream serial input	RHSB0SI1
RHSB0EMRG	RHSB0 Emergency input	RHSB0EMRG
RHSB1		
RHSB1FCLP	RHSB1 downstream serial clock differential output	RHSB1FCLP
RHSB1FCLN	RHSB1 downstream serial clock differential output	RHSB1FCLN
RHSB1SOP	RHSB1 downstream serial data differential output	RHSB1SOP
RHSB1SON	RHSB1 downstream serial data differential output	RHSB1SON
RHSB1CSD0	RHSB1 Slave 0 downstream chip selected output	RHSB1CSD0
RHSB1CSD1	RHSB1 Slave 1 downstream chip selected output	RHSB1CSD1
RHSB1SI0	RHSB1 Slave 0 upstream serial input	RHSB1SI0
RHSB1SI1	RHSB1 Slave 1 upstream serial input	RHSB1SI1
RHSB1EMRG	RHSB1 Emergency input	RHSB1EMRG

To enable single-ended signaling, disable (set “0” to) each LVDS in the “(4) LVDSCTRLB - LVDS Control B Register”, “(5) LVDSCTRLC - LVDS Control C Register” described in **Section 2.4.2.4, Other Registers**.

The pins that can be used for single-ended signaling are as follows:

- (1) Pins available for single-ended signaling when LVDS is disabled

P13_1: RHSB0FCLP, P13_3: RHSB0SOP

P10_0: RHSB1FCLP, P10_2: RHSB1SOP

- (2) Pins available for single-ended signaling with no LVDS setting

P12_4: RHSB0FCLP, P12_2: RHSB0SOP

The following pins are not available for single-ended signaling even when LVDS is disabled.

P13_0: RHSB0FCLN, P13_2: RHSB0SON

P10_1: RHSB1FCLN, P10_3: RHSB1SON

CAUTION

This AC specification is applied to the specific pin groups only. For details, see Appendix file “Limited_conditions_for_AC_specification.xlsx”.

22.2 Overview

The RHSB is designed to be used as a master node and provides two chip selects. It is designed to be compliant with MSC 1.0 (BOSCH/Infineon, July 2003) specification and MSC 1.0.0 (IPextreme, June 27, 2007).

22.2.1 Functional Overview

22.2.1.1 Downstream Channel Communication Functions

- Downstream communication
 - Synchronous serial transmission
 - Clock phase is selectable and clock activity is configurable during passive phases
 - Configurable repetition period is independent from data frame length (1 to 512_D bits (data bit))
 - Configurable downstream bit rate is $f_{DW} = f_{PE}/X$ with $X = (2, 4, 5, 8, 16, 32, 64, 128, 256)$. f_{PE} is typically 80 MHz. 160 MHz can be selected by using OPBT8.
 - Indication of the status of ongoing downstream transmission, for evaluation purposes
- Up to 2 slaves are supported with this individual configuration
 - Active level of the chip select and data bits
 - Selection bit for data frame present/not present
 - Length of assertion time (0 to 7_D bits) and of de-assertion time (0 to 7_D bits)
 - Selectable pin types of downstream channel interface (FCL, SO): LVDS output drivers or GPIO pins.
 - The LVDS driver is compliant with the ANSI/TIA/EIA-644 standard.
- Downstream data frame
 - Each data frame (1 to 64 bits) is composed of 1 to 4 DFTEs with individual length specification.
 - The user can define an individual source for each bit of a DFTE (e.g. timer input, data register).
 - The frame passive phase after each data frame is configurable from 1 to 64 bits.
 - Three transmission modes to schedule data transmission
- Downstream command frame
 - Command frame transmission with/without request of remote data
 - Dedicated register is available for defining 1 to 32 bits for command frame transmission.
 - Three different methods to insert command frames into the data frame schedule
- Emergency function
 - Handling of emergency condition can be enabled/disabled.
 - Emergency condition is identified by an input pin of the MCU with programmable active level.
 - Emergency input is selectable between level and edge sensitive.
 - After transmission of an emergency frame, data transmission can be stopped automatically.
 - Emergency data transmission DFTE bits are individually programmable through a dedicated register.

22.2.1.2 Upstream Channel Communication Functions

- Upstream communication
 - Asynchronous serial reception
 - Automatic selection of slave configuration when applying a command frame with data request
 - Timeout detection can be activated to monitor slave response on a command frame with data request
 - Status indication of ongoing upstream reception transmission, for evaluation purposes.
- Up to 2 slaves are supported for upstream communication with individual configurations.
 - Active level of data line
 - Frame with 8 or 12 data bits, even or odd parity bit, and 2 or 3 stop bits
 - Upstream bit rate derived from downstream bit rate
 - $f_{UP} = f_{DW}/2^x$ with $x = 3$ to 9 ; results in $/8$ to $/512$ with $f_{PE} = 80$ MHz
 - $f_{UP} = f_{DW}/2^x$ with $x = 4$ to 9 ; results in $/16$ to $/512$ with $f_{PE} = 160$ MHz
- Upstream status of up to 2 slaves
 - Valid data received
 - Data lost (overrun of data register)
 - Timeout error
 - Receive error (parity bit error or stop bit error)

22.2.1.3 Other Features

The RHSB module supports a test mode for key-on test. In this test mode, a simple check of the data path is possible by loopback of downstream transmission to the upstream reception.

22.2.1.4 Caution

This product supports only the driving ability with the setting of drive strength 4 (High) as the output buffer characteristics of pins (driving ability) for the chip select signals of the RHSB. For setting of output buffer characteristics, see the PUCcN Register in **Section 2, Pin Function**.

22.2.2 Block Diagram

Figure 22.1 shows a top level block diagram of the RHSB module.

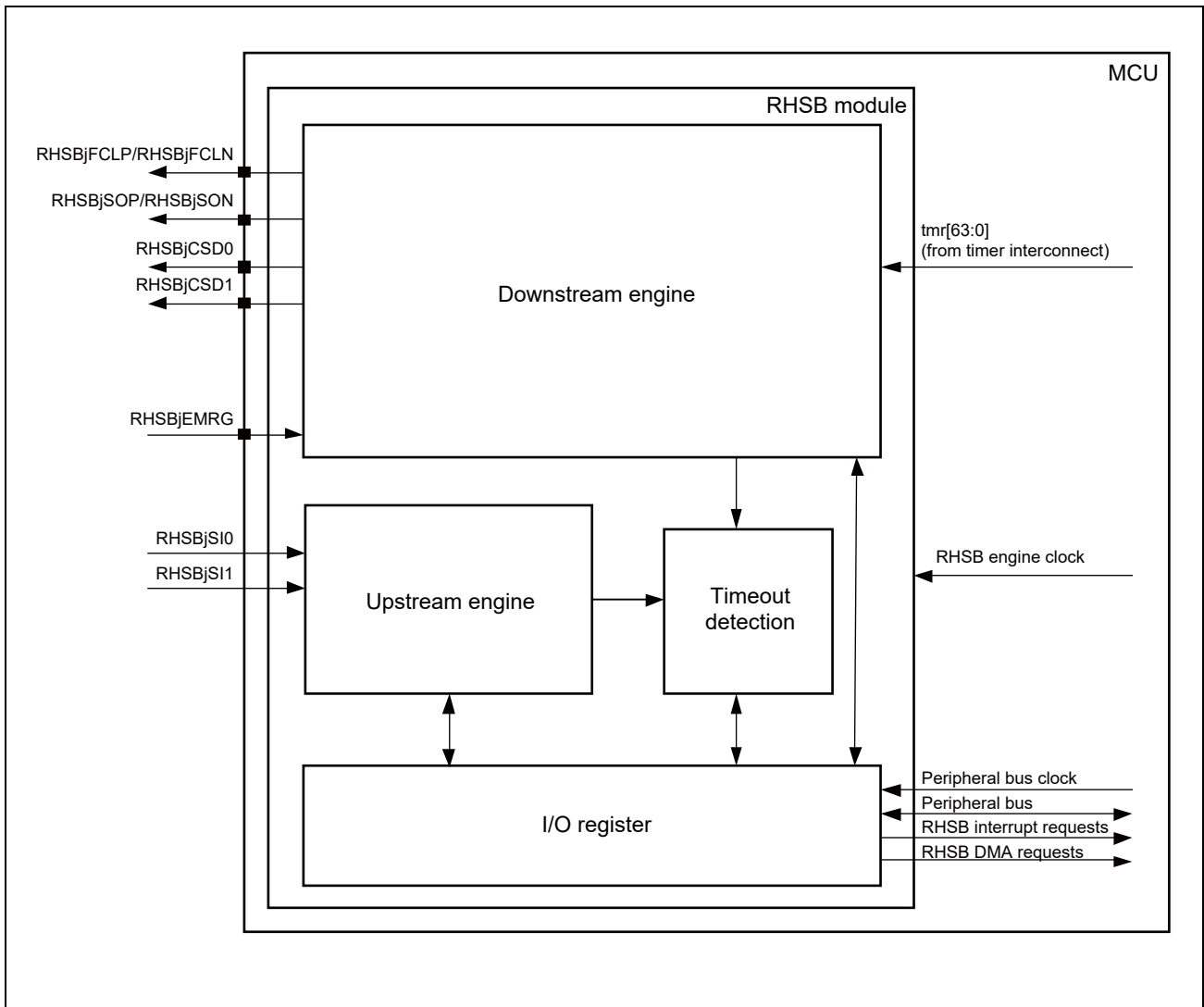


Figure 22.1 Block Diagram of RHSB

The tmr[63:0] bits of the timer correspond to the signals selected by XBAR. For details on XBAR, see **Section 22.6, Cross Bar (XBAR)**.

22.3 Registers

This section describes in detail the I/O registers of the RHSB module.

In case of a concurrent update of the same register bit, RHSB module write (hardware write) has higher priority than CPU write (software write).

22.3.1 List of Registers

The register map of a RHSB module is shown in **Table 22.10**.

The RHSB module remains in reset while RHSBjGC.RHSBjOPS indicates the RESET state.

For further information about the RHSB operation modes, see **Section 22.4, Operation**.

The module specific base address of a RHSB module needs to be added to each of the specified offset addresses.

Table 22.10 List of Registers

Module Name	Register Name	Symbol*1	Address	Access Size	Access Protect
RHSBj	Global Configuration Register	RHSBjGC	<RHSBj_base> + 00 _H	32, 16, 8	—
RHSBj	Module Status Register	RHSBjMSR	<RHSBj_base> + 08 _H	32, 16, 8	—
RHSBj	Downstream Configuration Register	RHSBjDCR	<RHSBj_base> + 10 _H	32, 16	—
RHSBj	Data Element Configuration Register	RHSBjDEC	<RHSBj_base> + 14 _H	32, 16, 8	—
RHSBj	Slave Device Configuration Register i (i = 0)	RHSBjSDCi	<RHSBj_base> + 18 _H + 4 × i	32, 16	—
RHSBj	Data Element Bit Assignment Register m (m = 0 to 3)	RHSBjDEBAm	<RHSBj_base> + 20 _H + 4 × m	32, 16	—
RHSBj	Emergency Bit Enable Register i (i = 0, 1)	RHSBjEBEi	<RHSBj_base> + 30 _H + 4 × i	32, 16	—
RHSBj	Downstream Transmission Control Register	RHSBjDTC	<RHSBj_base> + 38 _H	32, 16, 8	—
RHSBj	Downstream Command Data Register	RHSBjDCD	<RHSBj_base> + 3C _H	32, 16	—
RHSBj	Downstream Data Register i (i = 0, 1)	RHSBjDDRi	<RHSBj_base> + 40 _H + 4 × i	32, 16	—
RHSBj	Downstream Emergency Data Register i (i = 0, 1)	RHSBjDEDi	<RHSBj_base> + 48 _H + 4 × i	32, 16	—
RHSBj	Upstream Configuration Register	RHSBjUCR	<RHSBj_base> + 50 _H	32, 16, 8	—
RHSBj	Upstream Channel Configuration Register	RHSBjUCC	<RHSBj_base> + 54 _H	32, 16, 8	—
RHSBj	Upstream Channel Selection Register	RHSBjUCS	<RHSBj_base> + 58 _H	32, 16, 8	—
RHSBj	Upstream Data Read Register	RHSBjUDR	<RHSBj_base> + 5C _H	32, 16, 8	—
RHSBj	Upstream Data Register i (i = 0, 1)	RHSBjUDi	<RHSBj_base> + 60 _H + 4 × i	32, 16, 8	—
RHSBj	Upstream Status Summary Register	RHSBjUSS	<RHSBj_base> + 70 _H	32, 16, 8	—
RHSBj	Interrupt Control Register	RHSBjIC	<RHSBj_base> + 74 _H	32, 16, 8	—
RHSBj	Interrupt Status Register	RHSBjIS	<RHSBj_base> + 78 _H	32, 16, 8	—
RHSBj	Downstream Configuration Register for Period 1	RHSBjDCR1	<RHSBj_base> + 7C _H	32, 16, 8	—

The access size mentioned in **Table 22.10** is obligatory for write accesses. In addition, all registers can be read in 8-, 16- and 32-bit units.

The user should not write to bits a value specified as 'invalid' in the I/O register description.

22.3.2 Legend

This section explains the module state dependent abbreviations used for the I/O registers description in **Section 22.3.3, Description of the Registers in the Common Controller** to **Section 22.3.6, Interrupt Registers**.

- Conditions:

<Indexes used for register and register bit names>

i: Universal index

m: Index related to DFTE m

n: Index related to slave n

22.3.3 Description of the Registers in the Common Controller

22.3.3.1 RHSBjGC — Global Configuration Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	RHSBjDCDE	RHSBjDDE	RHSBjUDE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RHSBjOPS [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 22.11 RHSBjGC Register Contents

Bit Position	Bit Name	Function
31 to 19	—	Reserved. These bits are always read as 0. The write value should be always 0.
18	RHSBjDCDE	Downstream Command DMA Request Enable 0 _B : Downstream command DMA request is disabled. 1 _B : Downstream command DMA request is enabled.
17	RHSBjDDE	Downstream Data DMA Request Enable 0 _B : Downstream data DMA request is disabled. 1 _B : Downstream data DMA request is enabled.
16	RHSBjUDE	Upstream DMA Request Enable 0 _B : Upstream DMA request is disabled. 1 _B : Upstream DMA request is enabled.
15 to 2	—	Reserved. These bits are always read as 0. The write value should be always 0.
1, 0	RHSBjOPS[1:0]	Operation Status 00 _B : RESET state 01 _B : CONFIG state 10 _B : ACTIVE state 11 _B : TEST state

Downstream Command DMA Request Enable (RHSBjGC.RHSBjDCDE)

The user should not set this bit to 1_B when RHSBjIS.RHSBjCTF is used as interrupt source (RHSBjIC.RHSBjCTIE is 1_B).

This bit defines if the downstream command data register (RHSBjDCD) and downstream transmission control register (RHSBjDTC) are updated by downstream DMA transfer and command data are transferred to slaves when RHSBjIS.RHSBjCTF is 1_B.

For details about downstream command DMA, see **Section 22.5.6.2, DMA Usage for Downstream Command Transmission.**

Downstream Data DMA Enable (RHSBjGC.RHSBjDDE)

The user should not set this bit to 1_B when RHSBjIS.RHSBjDTSF is used as interrupt source (RHSBjIC.RHSBj DTSIE is 1_B).

This bit defines if the downstream data registers (RHSBjDDRi) are updated by downstream DMA transfer when RHSBjIS.RHSBjDTSF is 1_B.

For details about downstream data DMA, see **Section 22.5.6.1, DMA Usage for Downstream Data Transmission**.

Upstream DMA Enable (RHSBjGC.RHSBjUDE)

The user should not set this bit to 1_B when RHSBjIS.RHSBjDRF is used as interrupt source (RHSBjIC.RHSBjDRIE is 1_B).

This bit defines if the upstream data register (RHSBjUDR) is transferred by upstream DMA when RHSBjIS.RHSBjDRF is 1_B.

For details about upstream DMA, see **22.5.6.3, DMA Usage for Upstream Data Reception**.

Operation Status (RHSBjGC.RHSBjOPS)

The user cannot write 11_B (TEST) to these bits while RHSBjGC.RHSBjOPS is 00_B or 10_B.

The user cannot write 10_B (ACTIVE) to these bits while RHSBjGC.RHSBjOPS is 00_B or 11_B.

These bits define the current operation state of the RHSB module.

See **Section 22.4, Operation**.

22.3.3.2 RHSBjMSR — Module Status Register

Value after reset: 0001 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	—	—	RHSBjDFA [2:0]			—	—	—	—	—	—	—	RHSBj TPS	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RHSBj UFA	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.12 RHSBjMSR Register Contents

Bit Position	Bit Name	Function
31 to 27	—	Reserved. These bits are always read as 0. The write value should be always 0.
26 to 24	RHSBjDFA[2:0]	Downstream Frame Activity 000 _B : Inter-frame passive or disabled 001 _B : Passive phase of data frame 010 _B : Active phase of command frame 011 _B : Passive phase of command frame 100 _B : Active phase of data frame (DFTE0) 101 _B : Active phase of data frame (DFTE1) 110 _B : Active phase of data frame (DFTE2) 111 _B : Active phase of data frame (DFTE3)
23 to 17	—	Reserved. These bits are always read as 0. The write value should be always 0.
16	RHSBjTPS	Transmission Period Status 0 _B : Period 0 1 _B : Period 1
15 to 1	—	Reserved. These bits are always read as 0. The write value should be always 0.
0	RHSBjUFA	Upstream Frame Activity 0 _B : Upstream is idle 1 _B : Upstream frame decoding is ongoing

Downstream Frame Activity (RHSBjMSR.RHSBjDFA)

These bits indicate details of the downstream frame transmission. These bits are only valid when the downstream communication rate is equal to or falls below 5 megabits per second.

DTFE being transferred is flagged for the active phase of a data frame. A flag is placed in the slave select bit (assert phase or deassert phase) as a part of related DFTE.

[Changing conditions]

- These bits are updated when the downstream communication phase changes.
- These bits are updated when the number of DFTE used for transmission changes.
- These bits are set to 000_B when entering CONFIG state.

Transmission Period Status (RHSBjTPS bit in RHSBjMSR)

- This bit is only valid in multi-period repetition mode.
- This bit counts and specifies the transmission period.
- This bit starts to operate from 0_B when a transmission starts.

[Changing condition]

When multi-period repetition mode is selected, this bit is updated at the timing of time tick.

[Setting conditions]

- When a user writes 1_B to the RHSBjDTE bit in RHSBjDTC, reading the value of this bit returns 1_B.
- When the emergency function enable bit (the RHSBjEE bit in RHSBjDCR) is set to 1 1_B, reading the value of this bit returns 1_B.
- When entering the CONFIG state, reading the value of this bit returns 1_B.

Upstream Frame Activity (RHSBjMSR.RHSBjUFA)

This bit indicates that the upstream decoder has detected a start bit and a frame is currently being decoded.

[Clearing condition]

This bit is set to 0_B when the upstream decoder leaves RX-Active state.

[Setting condition]

This bit is set to 1_B when the upstream decoder enters RX-Active state.

22.3.4 Downstream (Tx) Registers

22.3.4.1 RHSBjDCR — Downstream Configuration Register

Value after reset: 0011 FF00_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RHSBjSLS [1:0]		RHSBjDFP [5:0]					RHSBjDBR [3:0]			RHSBjCIM [1:0]		RHSBjCTD	RHSBjREP[8]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RHSBjREP [7:0]							RHSBjCLP	RHSBjCAC	RHSBjEE [1:0]		RHSBjEIM	RHSBjEIP	RHSBjDMS [1:0]		
Value after reset	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.13 RHSBjDCR Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	RHSBjSLS[1:0]	Sequence Length Setting 00 _B : 1 DFTE (DFTE0) 01 _B : 2 DFTEs (DFTE0, DFTE1) 10 _B : 3 DFTEs (DFTE0 to DFTE2) 11 _B : 4 DFTEs (DFTE0 to DFTE3)
29 to 24	RHSBjDFP[5:0]	Data Frame Passive Length Length of the data frame passive phase
23 to 20	RHSBjDBR[3:0]	Downstream Bit Rate 0 _D : Invalid 1 _D : $f_{DW} = f_{PE}/2$ 2 _D : Invalid 3 _D : $f_{DW} = f_{PE}/4$ 4 _D : $f_{DW} = f_{PE}/5$ 5 _D : Invalid 6 _D : Invalid 7 _D : $f_{DW} = f_{PE}/8$ 8 _D : $f_{DW} = f_{PE}/16$ 9 _D : $f_{DW} = f_{PE}/32$ 10 _D : $f_{DW} = f_{PE}/64$ 11 _D : $f_{DW} = f_{PE}/128$ Others: $f_{DW} = f_{PE}/256$
19, 18	RHSBjCIM[1:0]	Command Frame Insertion Method 00 _B : Time-slot method 01 _B : Immediate method 10 _B : Best-effort method 11 _B : Invalid

Table 22.13 RHSBjDCR Register Contents (2/2)

Bit Position	Bit Name	Function
17	RHSBjCTD	Command Frame Transmission Delay 0 _B : No restriction 1 _B : Transmission of consecutive command frames prevented
16 to 8	RHSBjREP[8:0]	Repetition Period Length Length of the data frame repetition period
7	RHSBjCLP	Clock Line Phase 0 _B : Change on rising edge 1 _B : Change on falling edge
6	RHSBjCAC	Clock Active Control 0 _B : Clock is inactive during the passive phases 1 _B : Clock is always active
5, 4	RHSBjEE[1:0]	Emergency Enable 00 _B : Emergency condition detection disabled 01 _B : Emergency condition detection enabled and automatic stop disabled 10 _B : Invalid 11 _B : Emergency condition detection enabled and automatic stop enabled
3	RHSBjEIM	Emergency Input Mode 0 _B : Edge sensitive 1 _B : Level sensitive
2	RHSBjEIP	Emergency Input Polarity 0 _B : Active level is high / Rising edge 1 _B : Active level is low / Falling edge
1, 0	RHSBjDMS[1:0]	Downstream Mode Select 00 _B : Single-period repetition mode 01 _B : Triggered mode 10 _B : Multi-period repetition mode 11 _B : Invalid

Sequence Length Setting (RHSBjDCR.RHSBjSLS)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).
These bits define the number of DFTEs used to assemble one downstream data frame.

Data Frame Passive Length (RHSBjDCR.RHSBjDFP)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

These bits define the length of the frame passive phase (1_D to 64_D bits) for data frames.

When these bits are 0_D, the frame passive phase has the length of 1_D bit.

Note that a frame passive phase of 1_D bit is not valid according to the MSC specifications.

Downstream Bit Rate (RHSBjDCR.RHSBjDBR)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

These bits define how the downstream bit rate is derived from the f_{PE} clock.

Command Frame Insertion Method (RHSBjDCR.RHSBjCIM)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines how a command frame is inserted into an ongoing downstream schedule.

The following methods are available:

- Time-slot method:

Command frame transmission is delayed to be handled within the defined repetition period. The repetition period is not affected by command frame transmission.

- Immediate method:

Command frame transmission is handled as soon as possible. The repetition period can be re-adjusted in case of command frame transmission.

- Best-effort method:

Command frame transmission is handled as soon as possible with minimum impact to the data frame transmission. The repetition period is not affected by command frame transmission.

Command Frame Transmission Delay (RHSBjDCR.RHSBjCTD)

The user can write to this bit only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines if the command frame transmission is restricted to have at least one data frame between two command frames.

Repetition Period Length (RHSBjDCR.RHSBjREP)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

These bits define the length of the repetition period (1_D to 512_D bits) used for downstream communication in single-period repetition mode and multi-period repetition mode.

That is, when these bits are 0_D, the repetition period is 1_D bit.

See **Section 22.5.1.3, Downstream Modes, (2) Repetition Mode (Single-Period Repetition, Multi-Period Repetition)** for details.

Clock Line Phase (RHSBjDCR.RHSBjCLP)

The user can write to this bit only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines if the serial data line (RHSBjSOP, RHSBjSON) and the chip select lines (RHSBjCSD0, RHSBjCSD1) are changing with the rising or falling edge of the serial clock line (RHSBjFCLP, RHSBjFCLN).

Clock Active Control (RHSBjDCR.RHSBjCAC)

This bit defines the activity of the downstream serial clock line (RHSBjFCLP, RHSBjFCLN) during the passive phases. During active phases, the clock is always active.

Emergency Input Enable (RHSBjDCR.RHSBjEE)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

These bits enable the emergency function of the RHSB module.

In addition, these bits define if the data transmission is stopped after an emergency frame has been transmitted.

Emergency Input Mode (RHSBjDCR.RHSBjEIM)

The user can write to this bit only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines how the emergency condition is derived from the external emergency signal (RHSBjEMRG).

Emergency Input Polarity (RHSBjDCR.RHSBjEIP)

The user can write to this bit only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines the active level of the external emergency signal (RHSBjEMRG).

Downstream Mode Select (RHSBjDCR.RHSBjDMS)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

These bits define the timing mode of the downstream channel.

22.3.4.2 RHSBjDCR1 — Downstream Configuration Register for Period 1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RHSBjSLS1 [1:0]		—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.14 RHSBjDCR1 Register Contents

Bit Position	Bit Name	Function
31, 30	RHSBjSLS1[1:0]	Sequence Length Setting for Period 1 00 _B : 1 DFTE (DFTE0) 01 _B : 2 DFTEs (DFTE0, DFTE1) 10 _B : 3 DFTEs (DFTE0 to DFTE2) 11 _B : 4 DFTEs (DFTE0 to DFTE3)
29 to 0	—	Reserved. These bits are always read as 0. The write value should be always 0.

Sequence Length Setting for Period 1 (RHSBjDCR1.RHSBjSLS1)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

These bits define the number of DFTEs used to assemble one downstream data frame.

These bits are available only in multi-period repetition mode. RHSBjDCR1.RHSBjSLS1 value should be equal to or smaller than RHSBjDCR.RHSBjSLS value.

22.3.4.3 RHSBjDEC — Data Element Configuration Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	RHSBjSSD0 [1:0]		RHSBjNDB0 [3:0]			—	—	RHSBjSSD1 [1:0]		RHSBjNDB1 [3:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	RHSBjSSD2 [1:0]		RHSBjNDB2 [3:0]			—	—	RHSBjSSD3 [1:0]		RHSBjNDB3 [3:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	

Table 22.15 RHSBjDEC Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved. These bits are always read as 0. The write value should be always 0.
29, 28	RHSBjSSD0[1:0]	Slave Selected for DFTE0 Slave selected when transmitting DFTE0
27 to 24	RHSBjNDB0[3:0]	Number of Data Bits 0 Number of data bits used from DFTE0
23, 22	—	Reserved. These bits are always read as 0. The write value should be always 0.
21, 20	RHSBjSSD1[1:0]	Slave Selected for DFTE1 Slave selected when transmitting DFTE1
19 to 16	RHSBjNDB1[3:0]	Number of Data Bits 1 Number of data bits used from DFTE1
15, 14	—	Reserved. These bits are always read as 0. The write value should be always 0.
13, 12	RHSBjSSD2 [1:0]	Slave Selected for DFTE2 Slave selected when transmitting DFTE2
11 to 8	RHSBjNDB2[3:0]	Number of Data Bits 2 Number of data bits used from DFTE2
7, 6	—	Reserved. These bits are always read as 0. The write value should be always 0.
5, 4	RHSBjSSD3 [1:0]	Slave Selected for DFTE3 Slave selected when transmitting DFTE3
3 to 0	RHSBjNDB3[3:0]	Number of Data Bits 3 Number of data bits used from DFTE3

Slave Selected for DFTE_m (RHSBjDEC.RHSBjSSD_m) (m = 0 to 3)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

These bits define which slave is addressed by downstream frame transmission element m (DFTE_m). Setting 10_B and 11_B is prohibited.

Number of Data Bits m (RHSBjDEC.RHSBjNDBm) (m = 0 to 3)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

These bits define the number of bits (1 to 16) used from DFTE_m for data frame transmission.

When these bits are 0_D, 1 bit from DFTE_m is transmitted.

22.3.4.4 RHSBjSDCi — Slave Device Configuration Register i (i = 0)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	RHSBjAPLn [2:0]			—	—	RHSBj CSLPn	RHSBj SOLPn	RHSBj CPSn	RHSBjDPLn [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RHSBjAPLn [2:0]			—	—	RHSBj CSLPn	RHSBj SOLPn	RHSBj CPSn	RHSBjDPLn [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 2 × i (for bits 31 to 16)
n = 2 × i + 1 (for bits 15 to 0)

Table 22.16 RHSBjSDCi Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	—	Reserved. These bits are always read as 0. The write value should be always 0.
26 to 24	RHSBjAPLn[2:0]	Assertion Phase Length Assertion phase length for slave 0
23, 22	—	Reserved. These bits are always read as 0. The write value should be always 0.
21	RHSBjCSLPn	Chip Select Line Polarity Chip select line polarity for slave 0 0 _B : Active high 1 _B : Active low
20	RHSBjSOLPn	Serial Out Line Polarity Serial out line polarity for slave 0 0 _B : Inversion disabled 1 _B : Inversion enabled
19	RHSBjCPSn	Content Phase Selection Bit Enable Content phase selection bit enable for slave 0 0 _B : No selection bit is present 1 _B : Selection bit is present
18 to 16	RHSBjDPLn[2:0]	Deassertion Phase Length Deassertion phase length for slave 0
15 to 11	—	Reserved. These bits are always read as 0. The write value should be always 0.
10 to 8	RHSBjAPLn[2:0]	Assertion Phase Length Assertion phase length for slave 1
7, 6	—	Reserved. These bits are always read as 0. The write value should be always 0.
5	RHSBjCSLPn	Chip Select Line Polarity Chip select line polarity for slave 1 0 _B : Active high 1 _B : Active low

Table 22.16 RHSBjSDCi Register Contents (2/2)

Bit Position	Bit Name	Function
4	RHSBjSOLPn	Serial Out Line Polarity Serial out line polarity for slave 1 0 _B : Inversion disabled 1 _B : Inversion enabled
3	RHSBjCPSn	Content Phase Selection Bit Enable Content phase selection bit enable for slave 1 0 _B : No selection bit is present 1 _B : Selection bit is present
2 to 0	RHSBjDPLn[2:0]	Deassertion Phase Length Deassertion phase length for slave 1

Assertion Phase Length (RHSBjSDCi.RHSBjAPLn) (n = 0, 1)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

These bits define the assertion phase (0_D to 7_D bits) used for frames transmitted to slave n. If these bits are 0_D, no assertion phase is present.

Chip Select Line Polarity (RHSBjSDCi.RHSBjCSLPn) (n = 0, 1)

The user can write to this bit only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines the polarity of the chip select line (RHSBjCSDn (n = 0, 1)) for slave n.

Serial Out Line Polarity (RHSBjSDCi.RHSBjSOLPn) (n = 0, 1)

The user can write to this bit only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines the inversion of the serial data line (RHSBjSOP, RHSBjSON (j = 0, 1)) when slave n is selected.

Content Phase Selection Bit enable (RHSBjSDCi.RHSBjCPSn) (n = 0, 1)

The user can write to this bit only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines if there is a selection bit in data frames transmitted to slave n.

Note that in command frames, the selection bit is always present.

Deassertion Phase Length (RHSBjSDCi.RHSBjDPLn) (n = 0, 1)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

These bits define the deassertion phase (0_D to 7_D bits) used for frames transmitted to slave n. If these bits are 0_D, no deassertion phase is present.

22.3.4.5 RHSBjDEBAm — Data Element Bit Assignment Register m (m = 0 to 3)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RHSBjDSS15 [1:0]		RHSBjDSS14 [1:0]		RHSBjDSS13 [1:0]		RHSBjDSS12 [1:0]		RHSBjDSS11 [1:0]		RHSBjDSS10 [1:0]		RHSBjDSS9 [1:0]		RHSBjDSS8 [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RHSBjDSS7 [1:0]		RHSBjDSS6 [1:0]		RHSBjDSS5 [1:0]		RHSBjDSS4 [1:0]		RHSBjDSS3 [1:0]		RHSBjDSS2 [1:0]		RHSBjDSS1 [1:0]		RHSBjDSS0 [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.17 RHSBjDEBAm Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	RHSBjDSS15 [1:0]	Data Source Select 15 Data source select for bit 15 in DFTE _m
29, 28	RHSBjDSS14 [1:0]	Data Source Select 14 Data source select for bit 14 in DFTE _m
27, 26	RHSBjDSS13 [1:0]	Data Source Select 13 Data source select for bit 13 in DFTE _m
25, 24	RHSBjDSS12 [1:0]	Data Source Select 12 Data source select for bit 12 in DFTE _m
23, 22	RHSBjDSS11 [1:0]	Data Source Select 11 Data source select for bit 11 in DFTE _m
21, 20	RHSBjDSS10 [1:0]	Data Source Select 10 Data source select for bit 10 in DFTE _m
19, 18	RHSBjDSS9 [1:0]	Data Source Select 9 Data source select for bit 9 in DFTE _m
17, 16	RHSBjDSS8 [1:0]	Data Source Select 8 Data source select for bit 8 in DFTE _m
15, 14	RHSBjDSS7 [1:0]	Data Source Select 7 Data source select for bit 7 in DFTE _m
13, 12	RHSBjDSS6 [1:0]	Data Source Select 6 Data source select for bit 6 in DFTE _m
11, 10	RHSBjDSS5 [1:0]	Data Source Select 5 Data source select for bit 5 in DFTE _m
9, 8	RHSBjDSS4 [1:0]	Data Source Select 4 Data source select for bit 4 in DFTE _m
7, 6	RHSBjDSS3 [1:0]	Data Source Select 3 Data source select for bit 3 in DFTE _m
5, 4	RHSBjDSS2 [1:0]	Data Source Select 2 Data source select for bit 2 in DFTE _m
3, 2	RHSBjDSS1 [1:0]	Data Source Select 1 Data source select for bit 1 in DFTE _m

Table 22.17 RHSBjDEBAm Register Contents (2/2)

Bit Position	Bit Name	Function
1, 0	RHSBjDSS0 [1:0]	Data Source Select 0 Data source select for bit 0 in DFTE _m

Data Source Select i (RHSBjDEBAm.RHSBjDSSi) (i = 0 to 15)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

These bits define the data source of bit i in RHSBjDFTE_m.

00_B: Bit from timer input is selected

01_B: Inverted bit from timer input is selected

10_B: Bit from downstream data register RHSBjDDR0 or RHSBjDDR1 is selected

11_B: Invalid

See **Section 22.5.1.6, Data Frame Assembling** for details about the bit selection and the bit mapping.

22.3.4.6 RHSBjEBEi — Emergency Bit Enable Register i (i = 0, 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RHSBjEBE[15 + 32 × i:8 + 32 × i]								RHSBjEBE[7 + 32 × i:32 × i]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RHSBjEBE[31 + 32 × i:24 + 32 × i]								RHSBjEBE[23 + 32 × i:16 + 32 × i]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.18 RHSBjEBEi Register Contents

Bit Position	Bit Name	Function
31 to 24	RHSBjEBE [15 + 32 × i:8 + 32 × i]	Emergency Bit Enable Emergency Enable of DFTE i × 2 (DFTE0, DFTE2)
23 to 16	RHSBjEBE [7 + 32 × i:32 × i]	0 _B : Emergency function for this bit is disabled 1 _B : Emergency function for this bit is enabled
15 to 8	RHSBjEBE [31 + 32 × i:24 + 32 × i]	Emergency Bit Enable Emergency Enable of DFTE 1+i × 2 (DFTE1, DFTE3) 0 _B : Emergency function for this bit is disabled 1 _B : Emergency function for this bit is enabled
7 to 0	RHSBjEBE [23 + 32 × i:16 + 32 × i]	0 _B : Emergency function for this bit is disabled 1 _B : Emergency function for this bit is enabled

Emergency Bit Enable (RHSBjEBEi.RHSBjEBE)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

These bits define if the emergency function is enabled for a data frame bit.

Each bit enables the emergency function for the corresponding bit in an RHSBjDFTE. RHSBjDFTE0 is controlled by RHSBjEBE0[31:16], RHSBjDFTE1 by RHSBjEBE0[15:0], RHSBjDFTE2 by RHSBjEBE1[31:16] and RHSBjDFTE3 by RHSBjEBE1[15:0].

See **Section 22.5.1.6, Data Frame Assembling** for details.

22.3.4.7 RHSBjDTC — Downstream Transmission Control Register

Do not rewrite this register by using a bit-manipulation instruction.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	RHSBjNCB [4:0]				—	—	RHSBjSSCF [1:0]		—	—	RHSBjCTR [1:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RHSBj TSR	RHSBj DTE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 22.19 RHSBjDTC Register Contents

Bit Position	Bit Name	Function
31 to 29	—	Reserved. These bits are always read as 0. The write value should be always 0.
28 to 24	RHSBjNCB[4:0]	Number of Command Bits Number of bits used for command frame transmission
23, 22	—	Reserved. These bits are always read as 0. The write value should be always 0.
21, 20	RHSBjSSCF [1:0]	Slave Selection for Command Frame 00 _B : Command is send to slave 0 01 _B : Command is send to slave 1 10 _B : Setting prohibited 11 _B : Setting prohibited.
19, 18	—	Reserved. These bits are always read as 0. The write value should be always 0.
17, 16	RHSBjCTR [1:0]	Command Transmission Request 00 _B : No request 01 _B : Requests a command that does not require a response from the receive slave. 10 _B : Invalid 11 _B : Requests a command that requires a response from the receive slave.
15 to 2	—	Reserved. These bits are always read as 0. The write value should be always 0.
1	RHSBjTSR	Transmission Stop Request 0 _B : No action 1 _B : Request pending
0	RHSBjDTE	Data Transmission Enable 0 _B : Disabled 1 _B : Enabled

Number of Command Bits (RHSBjDTC.RHSBjNCB)

The user can write to these bits only if RHSBjDTC.RHSBjCTR is 00_B.

The user should not write to these bits if downstream command DMA is enabled (RHS-BjGC.RHSBjDCDE is 1_B).

These bits define the number of bits (1 to 32) used from the RHSBjDCD register for command frame transmission.

When these bits are 0_D, 1 bit from the RHSBjDCD register is transmitted.

Slave Selection for Command Frame (RHSBjDTC.RHSBjSSCF)

The user can write to these bits only if RHSBjDTC.RHSBjCTR is 00_B.

The user should not write to these bits if downstream command DMA is enabled (RHS-BjGC.RHSBjDCDE is 1_B).

These bits define the slave addressed for command frame transmission.

Command Transmission Request (RHSBjDTC.RHSBjCTR)

The user cannot write to these bits if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

The user cannot write 11_B to these bits if RHSBjUCS.RHSBjBSY is 1_B.

The user can write to these bits only if RHSBjDTC.RHSBjCTR is 00_B.

The user should not write to these bits if downstream command DMA is enabled (RHS-BjGC.RHSBjDCDE is 1_B).

These bits are used to request a command frame transmission.

[Changing conditions]

- These bits are set to 00_B when a command frame has been transmitted.
- These bits are set to 00_B when entering the CONFIG state.

Transmission Stop Request (RHSBjDTC.RHSBjTSR)

The user can only write 1_B to this bit.

The user cannot set this bit to 1_B if RHSBjDTC.RHSBjDTE is 0_B.

This bit defines if the data transmission should be stopped when there is no ongoing data frame transmission.

[Clearing conditions]

This bit is set to 0_B when data transmission is disabled (RHSBjDTC.RHSBjDTE is 0_B).

Data Transmission Enable (RHSBjDTC.RHSBjDTE)

The user cannot write to this bit if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

The user can only write 1_B to this bit.

This bit defines if the data frame transmission is enabled.

[Clearing conditions]

- This bit is set to 0_B when there is a transmission stop request (RHSBjDTC.RHSBjTSR is 1_B) and this request has been processed.
- This bit is set to 0_B when emergency condition detection and automatic stop is enabled (RHSBjDCR.RHSBjEE is 11_B) and an emergency frame has been transmitted.
- This bit is set to 0_B when the CONFIG state is entered.

22.3.4.8 RHSBjDCD — Downstream Command Data Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RHSBjCB[31:24]								RHSBjCB[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RHSBjCB[15:8]								RHSBjCB[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.20 RHSBjDCD Register Contents

Bit Position	Bit Name	Function
31 to 0	RHSBjCB[31:0]	Command Bit These bits define the data for command frame transmission

Command Bit (RHSBjDCD.RHSBjCB)

The user can write to these bits only if RHSBjDTC.RHSBjCTR is 00_B.

The user should not write to these bits if downstream command DMA is enabled (RHS-BjGC.RHSBjDCDE is 1_B).

These bits define the data bits used to assemble command frames.

The transmission order is LSB first.

See **Section 22.5.1.7, Command Frame Assembling** for details.

22.3.4.9 RHSBjDDRi — Downstream Data Register i (i = 0, 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RHSBjDB[15 + 32 × i:8 + 32 × i]								RHSBjDB[7 + 32 × i:32 × i]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RHSBjDB[31 + 32 × i:24 + 32 × i]								RHSBjDB[23 + 32 × i:16 + 32 × i]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.21 RHSBjDDRi Register Contents

Bit Position	Bit Name	Function
31 to 16	RHSBjDB [15 + 32 × i:32 × i]	Data Bit These bits define the data of DFTE i × 2 (DFTE0, DFTE2)
15 to 0	RHSBjDB [31 + 32 × i:16 + 32 × i]	Data Bit These bits define the data of DFTE 1+i × 2 (DFTE1, DFTE3)

Data Bit (RHSBjDDRi.RHSBjDB)

The user can write to these bits only if RHSBjIS.RHSBjDTSF is 1_B or only if RHSBjDTC.RHSBjDTE is 0_B.

If the user writes to these bits in other cases, the write operation is ignored and the written data is lost.

The user should not write to these bits if downstream data DMA is enabled (RHSBjGC.RHSBjDDE is 1_B).

These bits define the data bits of the RHSBjDFTEs used to assemble data frames.

The transmission order of each RHSBjDFTE is LSB first.

RHSBjDFTE0 is assembled from RHSBjDDR0[31:16], RHSBjDFTE1 from RHSBjDDR0[15:0], RHSBjDFTE2 from RHSBjDDR1[31:16] and RHSBjDFTE3 from RHSBjDDR1[15:0].

See **Section 22.5.1.6, Data Frame Assembling** for details.

Which bits of RHSBjDDRi can be a trigger of downstream transmission when written depends on RHSBjDCR.RHSBjSLS (the number of DFTE to be used).

For details, see **Section 22.5.1.10, Data Update and Data Frame Transmission Request**.

22.3.4.10 RHSBjDEDi — Downstream Emergency Data Register i (i = 0, 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RHSBjEB[15 + 32 × i:8 + 32 × i]								RHSBjEB[7 + 32 × i:32 × i]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RHSBjEB[31 + 32 × i:24 + 32 × i]								RHSBjEB[23 + 32 × i:16 + 32 × i]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.22 RHSBjDEDi Register Contents

Bit Position	Bit Name	Function
31 to 16	RHSBjEB [15 + 32 × i:32 × i]	Emergency Data Bit These bits define the emergency data of DFTE i × 2 (DFTE0, DFTE2)
15 to 0	RHSBjEB [31 + 32 × i:16 + 32 × i]	Emergency Data Bit These bits define the emergency data of DFTE 1+i × 2 (DFTE1, DFTE3)

Emergency Data Bit (RHSBjDEDi.RHSBjEB)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

These bits define the bits of the RHSBjDFTEs used to assemble data frames under emergency condition.

RHSBjDFTE0 is assembled from RHSBjDED0[31:16], RHSBjDFTE1 from RHSBjDED0[15:0], RHSBjDFTE2 from RHSBjDED1[31:16] and RHSBjDFTE3 from RHSBjDED1[15:0].

See **Section 22.5.1.6, Data Frame Assembling** for details.

22.3.5 Upstream (Rx) Registers

22.3.5.1 RHSBjUCR — Upstream Configuration Register

Value after reset: 0000 1800_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	RHSBjRTO[4:0]				—	—	—	—	RHSBjUE	RHSBjTOE	RHSBjUMS	RHSBjFSM	
Value after reset	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 22.23 RHSBjUCR Register Contents

Bit Position	Bit Name	Function
31 to 13	—	Reserved. These bits are always read as 0. The write value should be always 0.
12 to 8	RHSBjRTO[4:0]	Reception Timeout Period Timeout period in U bits
7 to 4	—	Reserved. These bits are always read as 0. The write value should be always 0.
3	RHSBjUE	Upstream Enable 0 _B : Disabled 1 _B : Enabled
2	RHSBjTOE	Timeout Enable 0 _B : Disabled 1 _B : Enabled
1	RHSBjUMS	Upstream Mode Select 0 _B : Dedicated mode 1 _B : Shared mode
0	RHSBjFSM	Frame Storing Method 0 _B : Normal storing method 1 _B : Addressed storing method For details, see Section 22.5.2.4, Frame Storing .

Reception Timeout Period (RHSBjUCR.RHSBjRTO)

These bits define the number of nominal upstream bit times before the timeout counter expires.

When these bits are 24_D (18_H), the timeout period has a length of 25 U bits.

Upstream Enable (RHSBjUCR.RHSBjUE)

The user can write to this bit only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines if reception on the upstream channels is enabled.

Timeout Enable (RHSBjUCR.RHSBjTOE)

The user can write to this bit only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines if the timeout function is enabled.

Upstream Mode Select (RHSBjUCR.RHSBjUMS)

The user can write to this bit only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines how the slaves are connected to the serial input lines of the RHSB module.

Frame Storing Method (RHSBjUCR.RHSBjFSM)

The user can write to this bit only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines where the result of upstream decoding is stored.

22.3.5.2 RHSBjUCC — Upstream Channel Configuration Register

Value after reset: 0F0F 0F0F_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RHSBjUFT0	RHSBjSBN0	RHSBjILP0	RHSBjPC0	RHSBjUBR0 [3:0]				RHSBjUFT1	RHSBjSBN1	RHSBjILP1	RHSBjPC1	RHSBjUBR1 [3:0]			
Value after reset	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.24 RHSBjUCC Register Contents

Bit Position	Bit Name	Function
31	RHSBjUFT0	Upstream Frame Type Defines if upstream frame on channel 0 uses an 8-bit or 12-bit data format
30	RHSBjSBN0	Stop Bit Number Defines if upstream frame on channel 0 uses 2 or 3 stop bit format
29	RHSBjILP0	Serial Input Line Polarity Defines if the serial input line of channel 0 (RHSBjSI0) is inverted
28	RHSBjPC0	Parity Control Defines if upstream frame on channel 0 uses even or odd parity
27 to 24	RHSBjUBR0[3:0]	Upstream Bit Rate Defines the upstream bit rate used for channel 0
23	RHSBjUFT1	Upstream Frame Type Defines if upstream frame on channel 1 uses an 8-bit or 12-bit data format
22	RHSBjSBN1	Stop Bit Number Defines if upstream frame on channel 1 uses 2 or 3 stop bit format
21	RHSBjILP1	Serial Input Line Polarity Defines if the serial input line of channel 1 (RHSBjSI1) is inverted
20	RHSBjPC1	Parity Control Defines if upstream frame on channel 1 uses even or odd parity
19 to 16	RHSBjUBR1[3:0]	Upstream Bit Rate Defines the upstream bit rate used for channel 1
15 to 0	Reserved	These bits are always read as their initial values. The write value should always be the initial one.

Upstream Frame Type (RHSBjUCC.RHSBjUFTn)

The user can write to this bit only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines the number of data bits of an upstream frame received on channel n.

0_B: 8-bit data field format

1_B: 12-bit data field format (including 4-bit address)

Stop Bit Number (RHSBjUCC.RHSBjSBNn)

The user can write to this bit only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines the number of stop bits of an upstream frame received on channel n.

0_B: 2 stop bits format

1_B: 3 stop bits format

Serial Input Line Polarity (RHSBjUCC.RHSBjILPn)

The user can write to this bit only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines if there is an inversion of the serial input line of channel n.

0_B: Inversion disabled

1_B: Inversion enabled

Note that when RHSBjUCR.RHSBjUMS is shared mode, RHSBjILP0 is used for all four channels (RHS-BjILP1 are ignored).

Parity Control (RHSBjUCC.RHSBjPCn)

The user can write to this bit only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

This bit defines the parity bit interpretation of an upstream frame received on channel n.

0_B: Even parity

1_B: Odd parity

Upstream Bit Rate (RHSBjUCC.RHSBjUBRn)

The user can write to these bits only if RHSBjGC.RHSBjOPS is 01_B (CONFIG).

These bits define the relation between the upstream bit rate and downstream bit rate for channel n.

0_D: Setting prohibited

1_D: Setting prohibited

2_D: Setting prohibited

3_D: $f_{UP} = f_{DW}/8$

4_D: $f_{UP} = f_{DW}/16$

5_D: $f_{UP} = f_{DW}/32$

6_D: $f_{UP} = f_{DW}/64$

7_D: $f_{UP} = f_{DW}/128$

8_D: $f_{UP} = f_{DW}/256$

Others: $f_{UP} = f_{DW}/512$

Not all combinations of upstream and downstream bit rates are valid. See **Section 22.5.2.5, Upstream Bit Rates** for details.

22.3.5.3 RHSBjUCS — Upstream Channel Selection Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RHSBjBSY	—	—	—	—	—	RHSBjACC [1:0]		—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.25 RHSBjUCS Register Contents

Bit Position	Bit Name	Function
31	RHSBjBSY	Busy 0 _B : Configuration ready 1 _B : Processing configuration change
30 to 26	—	Reserved. These bits are always read as 0. The write value should be always 0.
25, 24	RHSBjACC[1:0]	Active Channel Configuration 00 _B : Configuration given for channel 0 01 _B : Configuration given for channel 1 10 _B : Setting prohibited 11 _B : Setting prohibited
23 to 0	—	Reserved. These bits are always read as 0. The write value should be always 0.

Busy (RHSBjUCS.RHSBjBSY)

This bit indicates that the RHSB module is processing the change of active channel configuration.

[Clearing conditions]

- This bit is set to 0_B when the change of active channel configuration has been processed.
- This bit is set to 0_B when entering the CONFIG state.

[Setting conditions]

- This bit is set to 1_B when RHSBjUCS.RHSBjACC changes and RHSBjGC.RHSBjOPS is not 01_B (CONFIG).
- This bit is set to 1_B when leaving the CONFIG state.

Active Channel Configuration (RHSBjUCS.RHSBjACC)

The user cannot write to these bits if RHSBjUCS.RHSBjBYS is 1_B.

The user should not write to these bits after starting a command transmission by using an upstream data request (by setting RHSBjDTC.RHSBjCTR to 11_B) until the upstream reception has been completed. The exact period where an upstream reception is expected is application specific.

These bits define the channel configuration specified in RHSBjUCC register to be used for upstream reception. In dedicated mode (RHSBjUCR.RHSBjUMS = 0_B), these bits also define the serial input line.

Note that an ongoing reception is aborted and the timeout detection is deactivated when these bits are being changed.

[Changing condition]

These bits are updated to the value of RHSBjDTC.RHSBjSSCF when the user requests a command transmission with upstream data request by setting RHSBjDTC.RHSBjCTR to 11_B.

22.3.5.4 RHSBjUDR — Upstream Data Read Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RHSBjNDS [0:1]	—	—	—	—	RHSBjLUD [1:0]	—	—	—	RHSBj DL	RHSBj TO	RHSBj FERR	RHSBj PERR	RHSBj ND		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RHSBjEDF [3:0]			RHSBjDF [7:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.26 RHSBjUDR Register Contents (1/2)

Bit Position	Bit Name	Function
31	RHSBjNDS0	New Data Summary 0 _B : No new reception of a fault free frame on upstream channel 0 1 _B : New frame received without fault on upstream channel 0
30	RHSBjNDS1	New Data Summary 0 _B : No new reception of a fault free frame on upstream channel 1 1 _B : New frame received without fault on upstream channel 1
29 to 26	—	Reserved. These bits are always read as 0. The write value should be always 0.
25, 24	RHSBjLUD[1:0]	Last Updated Data 00 _B : UD0 (channel 0) 01 _B : UD1 (channel 1) 10 _B : invalid 11 _B : invalid
23 to 21	—	Reserved. These bits are always read as 0. The write value should be always 0.
20	RHSBjDL	Data Lost 0 _B : No data loss detected 1 _B : Data loss detected
19	RHSBjTO	Timeout Detected 0 _B : No timeout detected 1 _B : Timeout detected
18	RHSBjFERR	Frame Error 0 _B : No stop bit error detected 1 _B : Stop bit error detected
17	RHSBjPERR	Parity Error 0 _B : No parity error detected 1 _B : Parity error detected
16	RHSBjND	New Data 0 _B : No new reception of a fault free frame 1 _B : New frame received without fault
15 to 12	—	Reserved. These bits are always read as 0. The write value should be always 0.

Table 22.26 RHSBjUDR Register Contents (2/2)

Bit Position	Bit Name	Function
11 to 8	RHSBjEDF[3:0]	Extended Data Field 4-bit extended data field of last received frame
7 to 0	RHSBjDF[7:0]	Data Field 8-bit data field of last received frame

The user should not read the RHSBjUDR register when upstream DMA is enabled (RHSBjGC.RHSBjUDE is 1_B).

Note that reading this register sets RHSBjIS.RHSBjDRF to 0_B and sets RHSBjUDi.RHSBjND of channel i to 0_B. Channel number i is shown by RHSBjUDR.RHSBjLUD.

New Data Summary (RHSBjUDR.RHSBjNDSi)

These bits indicate the value of RHSBjUDi.RHSBjND from channel i (i = 0, 1).

[Changing condition]

This bit is updated when RHSBjUDi.RHSBjND changes.

Last Updated Data (RHSBjUDR.RHSBjLUD)

These bits indicate the number of Upstream Data register (RHSBjUDi) where the last data update occurred.

[Changing condition]

This bit is updated when the set condition of one of the RHSBjUDi.RHSBjND bits is fulfilled.

Data Lost (RHSBjUDR.RHSBjDL)

This bit indicates the value of RHSBjUDi.RHSBjDL from channel i. Channel number i is shown by RHSBjUDR.RHSBjLUD.

[Changing condition]

This bit is updated when RHSBjUDR.RHSBjLUD or RHSBjUDi.RHSBjDL (with i = RHSBjUDR.RHSBjLUD) changes.

Timeout Detected (RHSBjUDR.RHSBjTO)

This bit indicates the value of RHSBjUDi.RHSBjTO from channel i. Channel number i is shown by RHSBjUDR.RHSBjLUD.

[Changing condition]

This bit is updated when RHSBjUDR.RHSBjLUD or RHSBjUDi.RHSBjTO (with i = RHSBjUDR.RHSBjLUD) changes.

Frame Error (RHSBjUDR.RHSBjFERR)

This bit indicates the value of RHSBjUDi.RHSBjFERR from channel i. Channel number i is shown by RHSBjUDR.RHSBjLUD.

[Changing condition]

This bit is updated when RHSBjUDR.RHSBjLUD or RHSBjUDi.RHSBjFERR (with i = RHSBjUDR.RHSBjLUD) changes.

Parity Error (RHSBjUDR.RHSBjPERR)

This bit indicates the value of RHSBjUDi.RHSBjPERR from channel i. Channel number i is shown by RHSBjUDR.RHSBjLUD.

[Changing condition]

This bit is updated when RHSBjUDR.RHSBjLUD or RHSBjUDi.RHSBjPERR (with i = RHSBjUDR.RHSBjLUD) changes.

New Data (RHSBjUDR.RHSBjND)

This bit indicates the value of RHSBjUDi.RHSBjND from channel i. Channel number i is shown by RHSBjUDR.RHSBjLUD.

[Changing condition]

This bit is updated when RHSBjUDR.RHSBjLUD or RHSBjUDi.RHSBjND (with i = RHSBjUDR.RHSBjLUD) changes.

Extended Data Field (RHSBjUDR.RHSBjEDF)

This bit indicates the value of RHSBjUDi.RHSBjEDF from channel i. Channel number i is shown by RHSBjUDR.RHSBjLUD.

[Changing condition]

This bit is updated when RHSBjUDR.RHSBjLUD or RHSBjUDi.RHSBjEDF (with i = RHSBjUDR.RHSBjLUD) changes.

Data Field (RHSBjUDR.RHSBjDF)

This bit indicates the value of RHSBjUDi.RHSBjDF from channel i. Channel number i is shown by RHSBjUDR.RHSBjLUD.

[Changing condition]

This bit is updated when RHSBjUDR.RHSBjLUD or RHSBjUDi.RHSBjDF (with i = RHSBjUDR.RHSBjLUD) changes.

22.3.5.5 RHSBjUDI — Upstream Data Register i (i = 0, 1)

Do not clear the RHSBjDL, RHSBjTO, RHSBjFERR, RHSBjPERR, and RHSBjND bits in this register by using a bit-manipulation instruction.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	RHSBjDL	RHSBjTO	RHSBjFERR	RHSBjPERR	RHSBjND
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RHSBjEDF [3:0]				RHSBjDF [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.27 RHSBjUDI Register Contents

Bit Position	Bit Name	Function
31 to 21	—	Reserved. These bits are always read as 0. The write value should be always 0.
20	RHSBjDL	Data Lost 0 _B : No data loss detected 1 _B : Data lost situation detected
19	RHSBjTO	Timeout Detected 0 _B : No timeout detected 1 _B : Timeout detected
18	RHSBjFERR	Frame Error 0 _B : No stop bit error detected 1 _B : Stop bit error detected
17	RHSBjPERR	Parity Error 0 _B : No parity error detected 1 _B : Parity error detected
16	RHSBjND	New Data 0 _B : No valid data available 1 _B : Valid data available
15 to 12	—	Reserved. These bits are always read as 0. The write value should be always 0.
11 to 8	RHSBjEDF[3:0]	Extended Data Field 4-bit extended data field of received frame
7 to 0	RHSBjDF[7:0]	Data Field 8-bit data field of received frame

Data Lost (RHSBjUDI.RHSBjDL)

The user can only write 0_B to this bit.

This bit indicates that a fault free frame was received on channel i while the RHSBjND bit in this register is 1_B.

[Clearing condition]

This bit is set to 0_B when entering the CONFIG state.

[Setting condition]

This bit is set to 1_B when the set condition of ND fulfilled while the ND bit in this register is 1_B.

Timeout Detected (RHSBjUDi.RHSBjTO)

The user can only write 0_B to this bit.

This bit indicates that there was no valid reception after a command frame transmission based on an upstream data request within the timeout period.

[Clearing condition]

This bit is set to 0_B when entering the CONFIG state.

[Setting condition]

This bit is set to 1_B when the timeout counter expires and RHSBjUCS.RHSBjACC is indexing this channel.

Frame Error (RHSBjUDi.RHSBjFERR)

The user can only write 0_B to this bit.

This bit indicates that a frame with stop bit error was received on channel i.

[Clearing condition]

This bit is set to 0_B when entering the CONFIG state.

[Setting condition]

This bit is set to 1_B after an upstream reception with at least one faulty stop bit.

Parity Error (RHSBjUDi.RHSBjPERR)

The user can only write 0_B to this bit.

This bit indicates that a frame with parity error was received on channel i.

[Clearing condition]

This bit is set to 0_B when entering the CONFIG state.

[Setting condition]

This bit is set to 1_B after an upstream reception with faulty parity bit.

New Data (RHSBjUDi.RHSBjND)

The user can only write 0_B to this bit.

This bit indicates that RHSBjUDi.RHSBjEDF and RHSBjUDi.RHSBjDF containing valid data were received on channel i.

[Clearing conditions]

- This bit is set to 0_B when there is a read access to the RHSBjUDR and RHSBjUDR.RHSBjLUD is indexing this channel.
- This bit is set to 0_B when entering the CONFIG state.

[Setting condition]

This bit is set to 1_B after an upstream reception without parity and without stop bit errors.

Extended Data Field (RHSBjUDi.RHSBjEDF)

These bits indicate the 4-bit extended data field bits of received message.

When using an 8-bit data format, these bits are undefined.

[Changing condition]

These bits are updated when the set condition of RHSBjND is fulfilled.

Data Field (RHSBjUDi.RHSBjDF)

These bits indicate the 8-bit data field of received message.

[Changing condition]

These bits are updated when the set condition of RHSBjND is fulfilled.

22.3.5.6 RHSBjUSS — Upstream Status Summary Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	RHSBjDL0	RHSBjTO0	RHSBjFERR0	RHSBjPERR0	RHSBjND0	—	—	—	RHSBjDL1	RHSBjTO1	RHSBjFERR1	RHSBjPERR1	RHSBjND1
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.28 RHSBjUSS Register Contents

Bit Position	Bit Name	Function
31 to 29	—	Reserved. These bits are always read as 0. The write value should be always 0.
28	RHSBjDL0	Data Lost Data lost indication of channel 0 (RHSBjUD0.RHSBjDL).
27	RHSBjTO0	Timeout Detected Timeout indication of channel 0 (RHSBjUD0.RHSBjTO).
26	RHSBjFERR0	Frame Error Frame error indication of channel 0 (RHSBjUD0.RHSBjFERR).
25	RHSBjPERR0	Parity Error Parity error indication of channel 0 (RHSBjUD0.RHSBjPERR).
24	RHSBjND0	New Data New data indication of channel 0 (RHSBjUD0.RHSBjND).
23 to 21	—	Reserved. These bits are always read as 0. The write value should be always 0.
20	RHSBjDL1	Data Lost Data lost indication of channel 1 (RHSBjUD1.RHSBjDL).
19	RHSBjTO1	Timeout Detected Timeout indication of channel 1 (RHSBjUD1.RHSBjTO).
18	RHSBjFERR1	Frame Error Frame error indication of channel 1 (RHSBjUD1.RHSBjFERR).
17	RHSBjPERR1	Parity Error Parity error indication of channel 1 (RHSBjUD1.RHSBjPERR).
16	RHSBjND1	New Data New data indication of channel 1 (RHSBjUD1.RHSBjND).
15 to 0	—	Reserved. These bits are always read as 0. The write value should be always 0.

Data Lost (RHSBjUSS.RHSBjDLi)

This bit indicates the value of RHSBjUDi.RHSBjDL from channel i.

0_B: No data loss detected

1_B: Data loss detected

[Changing condition]

This bit is updated when RHSBjUDi.RHSBjDL changes.

Timeout Detected (RHSBjUSS.RHSBjTOi)

This bit indicates the value of RHSBjUDi.RHSBjTO from channel i.

0_B: No timeout detected

1_B: Timeout detected

[Changing condition]

This bit is updated when RHSBjUDi.RHSBjTO changes.

Frame Error (RHSBjUSS.RHSBjFERRi)

This bit indicates the value of RHSBjUDi.RHSBjFERR from channel i.

0_B: No stop bit error detected

1_B: Stop bit error detected

[Changing condition]

This bit is updated when RHSBjUDi.RHSBjFERR changes.

Parity Error (RHSBjUSS.RHSBjPERRi)

This bit indicates the value of RHSBjUDi.RHSBjPERR from channel i.

0_B: No parity error detected

1_B: Parity error detected

[Changing condition]

This bit is updated when RHSBjUDi.RHSBjPERR changes.

New Data (RHSBjUSS.RHSBjNDi)

This bit indicates the value of RHSBjUDi.RHSBjND from channel i.

0_B: No valid data available

1_B: Valid data available

[Changing condition]

This bit is updated when RHSBjUDi.RHSBjND changes.

22.3.6 Interrupt Registers

22.3.6.1 RHSBjIC — Interrupt Control Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	RHSBjDLIE	RHSBjTOIE	RHSBjUEIE	RHSBjDRIE	—	—	—	RHSBjTSIE	RHSBjETIE	RHSBjCTIE	RHSBjDTIE	RHSBjDTSIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.29 RHSBjIC Register Contents

Bit Position	Bit Name	Function
31 to 28	—	Reserved. These bits are always read as 0. The write value should be always 0.
27	RHSBjDLIE	Data Lost Interrupt Enable 0 _B : Disabled 1 _B : Enabled
26	RHSBjTOIE	Timeout Detected Interrupt Enable 0 _B : Disabled 1 _B : Enabled
25	RHSBjUEIE	Upstream Error Interrupt Enable 0 _B : Disabled 1 _B : Enabled
24	RHSBjDRIE	Data Received Interrupt Enable 0 _B : Disabled 1 _B : Enabled
23 to 21	—	Reserved. These bits are always read as 0. The write value should be always 0.
20	RHSBjTSIE	Transmission Started Interrupt Enable 0 _B : Disabled 1 _B : Enabled
19	RHSBjETIE	Emergency Frame Transmission Done Interrupt Enable 0 _B : Disabled 1 _B : Enabled
18	RHSBjCTIE	Command Frame Transmission Done Interrupt Enable 0 _B : Disabled 1 _B : Enabled
17	RHSBjDTIE	Data Frame Transmission Done Interrupt Enable 0 _B : Disabled 1 _B : Enabled
16	RHSBjDTSIE	Data Frame Transmission Started Interrupt Enable 0 _B : Disabled 1 _B : Enabled
15 to 0	—	Reserved. These bits are always read as 0. The write value should be always 0.

Data Lost Interrupt Enable (RHSBjIC.RHSBjDLIE)

While this bit is 1_B, an interrupt will be generated if RHSBjIS.RHSBjDLF is 1_B.

Timeout Detected Interrupt Enable (RHSBjIC.RHSBjTOIE)

While this bit is 1_B, an interrupt will be generated if RHSBjIS.RHSBjTOF is 1_B.

Upstream Error Interrupt Enable (RHSBjIC.RHSBjUEIE)

While this bit is 1_B, an interrupt will be generated if RHSBjIS.RHSBjUEF is 1_B.

Data Received Interrupt Enable (RHSBjIC.RHSBjDRIE)

The user should not set this bit to 1_B when the upstream DMA support is enabled (RHSBjGC.RHSBjUDE is 1_B).

While this bit is 1_B, an interrupt will be generated if RHSBjIS.RHSBjDRF is 1_B.

Transmission Started Interrupt Enable (RHSBjIC.RHSBjTSIE)

While this bit is 1_B, an interrupt will be generated if RHSBjIS.RHSBjTSF is 1_B.

Emergency Frame Transmission Done Interrupt Enable (RHSBjIC.RHSBjETIE)

While this bit is 1_B, an interrupt will be generated if RHSBjIS.RHSBjETF is 1_B.

Command Frame Transmission Done Interrupt Enable (RHSBjIC.RHSBjCTIE)

The user should not set this bit to 1_B when downstream command DMA transfer is enabled (RHSBjGC.RHSBjDCDE is 1_B).

While this bit is 1_B, an interrupt will be generated if RHSBjIS.RHSBjCTF is 1_B.

Data Frame Transmission Done Interrupt Enable (RHSBjIC.RHSBjDTIE)

While this bit is 1_B, an interrupt will be generated if RHSBjIS.RHSBjDTF is 1_B.

Data Frame Transmission Started Interrupt Enable (RHSBjIC.RHSBjDTSIE)

The user should not set this bit to 1_B when the downstream data DMA support is enabled (RHSBjGC.RHSBjDDE is 1_B).

While this bit is 1_B, an interrupt will be generated if RHSBjIS.RHSBjDTSF is 1_B.

22.3.6.2 RHSBjIS — Interrupt Status Register

Do not clear the bits in this register by using a bit-manipulation instruction.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	RHSBj DLF	RHSBj TOF	RHSBj UEF	RHSBj DRF	—	—	—	RHSBj TSF	RHSBj ETF	RHSBj CTF	RHSBj DTF	RHSBj DTSF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RHSBj ERF	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Table 22.30 RHSBjIS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	—	Reserved. These bits are always read as 0. The write value should be always 0.
27	RHSBjDLF	Data Lost Flag 0 _B : No data loss detected 1 _B : Data loss detected
26	RHSBjTOF	Timeout Detected Flag 0 _B : No timeout detected 1 _B : Timeout detected
25	RHSBjUEF	Upstream Error Flag 0 _B : No receive error detected 1 _B : Receive error detected
24	RHSBjDRF	Data Received Flag 0 _B : No reception of new valid data 1 _B : Reception of new valid data
23 to 21	—	Reserved. These bits are always read as 0. The write value should be always 0.
20	RHSBjTSF	Transmission Started Flag 0 _B : No transmission started 1 _B : Transmission has been started
19	RHSBjETF	Emergency Frame Transmission Done Flag 0 _B : No emergency transmission 1 _B : Emergency frame has been transmitted
18	RHSBjCTF	Command Frame Transmission Done Flag 0 _B : No command frame transmission 1 _B : Command frame has been transmitted
17	RHSBjDTF	Data Frame Transmission Done Flag 0 _B : No data frame transmission 1 _B : Data frame has been transmitted
16	RHSBjDTSF	Data Frame Transmission Started Flag 0 _B : No data transmission has been started 1 _B : Data transmission has been started

Table 22.30 RHSBjIS Register Contents (2/2)

Bit Position	Bit Name	Function
15 to 9	—	Reserved. These bits are always read as 0. The write value should be always 0.
8	RHSBjERF	Emergency Signal Rising Flag 0 _B : No edge detected 1 _B : Edge detected
7 to 0	—	Reserved. These bits are always read as 0. The write value should be always 0.

Data Lost Flag (RHSBjIS.RHSBjDLF)

The user can only write 0_B to this bit.

This bit indicates that data lost has been detected on at least one channel.

[Clearing conditions]

- This bit is set to 0_B when all RHSBjUDi.RHSBjDL (i = 0, 1) flags are 0_B.
- This bit is set to 0_B when entering the CONFIG state.

[Setting condition]

This bit is set to 1_B when the set condition of at least one of these flags (RHSBjUDi.RHSBjDL with i = 0, 1) is fulfilled.

Timeout Detected Flag (RHSBjIS.RHSBjTOF)

The user can only write 0_B to this bit.

This bit indicates that there was no valid reception requested by a command frame transmission within the timeout period.

[Clearing condition]

This bit is set to 0_B when entering the CONFIG state.

[Setting condition]

This bit is set to 1_B when the timeout counter expires.

Upstream Error Flag (RHSBjIS.RHSBjUEF)

The user can only write 0_B to this bit.

This bit indicates that a frame with parity or stop bit error was received.

[Clearing conditions]

- This bit is set to 0_B when all RHSBjUDi.RHSBjFERR (i = 0, 1) and all RHSBjUDi.RHSBjPERR (i = 0, 1) flags are 0_B.
- This bit is set to 0_B when entering the CONFIG state.

[Setting conditions]

- This bit is set to 1_B when the set condition of at least one of these flags (RHSBjUDi.RHSBjFERR with i = 0, 1) is fulfilled.
- This bit is set to 1_B when the set condition of at least one of these flags (RHSBjUDi.RHSBjPERR with i = 0, 1) is fulfilled.

Data Received Flag (RHSBjIS.RHSBjDRF)

The user can only write 0_B to this bit.

The user should not set this bit to 0_B when upstream DMA is enabled (RHSBjGC.RHSBjUDE is 1_B).

This bit indicates that at least one RHSBjUDi register is updated with new valid data.

[Clearing conditions]

- This bit is set to 0_B when all RHSBjUDi.RHSBjND (i = 0, 1) flag are 0_B.
- This bit is set to 0_B when there is a read access to RHSBjUDR.
- This bit is set to 0_B when entering the CONFIG state.

[Setting condition]

This bit is set to 1_B when the set condition of at least one of these flags (RHSBjUDi.RHSBjND with i = 0, 1) is fulfilled.

Transmission Started Flag (RHSBjIS.RHSBjTSF)

The user can only write 0_B to this bit.

This bit indicates that a command or data frame transmission has been started.

[Clearing condition]

This bit is set to 0_B when entering the CONFIG state.

[Setting condition]

This bit is set to 1_B when a new frame transmission starts.

Emergency Frame Transmission Done Flag (RHSBjIS.RHSBjETF)

The user can only write 0_B to this bit.

This bit indicates that a data frame assembled under emergency condition has been transmitted.

[Clearing condition]

This bit is set to 0_B when entering the CONFIG state.

[Setting condition]

This bit is set to 1_B when an emergency frame has been transmitted.

Command Frame Transmission Done Flag (RHSBjIS.RHSBjCTF)

The user can only write 0_B to this bit.

The user should not set this bit to 0_B when downstream command DMA transfer is enabled (RHSBjGC.RHSBjDCDE is 1_B).

This bit indicates that a command frame has been transmitted.

[Clearing conditions]

- This bit is set to 0_B when a new command frame transmission is requested.
- This bit is set to 0_B when entering the CONFIG state.

[Setting condition]

This bit is set to 1_B when a command frame has been transmitted.

Data Frame Transmission Done Flag (RHSBjIS.RHSBjDTF)

The user can only write 0_B to this bit.

This bit indicates that a data frame has been transmitted.

[Clearing conditions]

- This bit is set to 0_B when the last active portions of the downstream data registers (RHSBjDDRi) are written (see **Section 22.5.1.10, Data Update and Data Frame Transmission Request**).
- This bit is set to 0_B when entering the CONFIG state.

[Setting condition]

This bit is set to 1_B when a data frame has been transmitted.

Data Frame Transmission Started Flag (RHSBjIS.RHSBjDTSF)

The user can only write 0_B to this bit.

The user should not set this bit to 0_B when downstream data DMA is enabled (RHSBjGC.RHSBjDDE is 1_B).

This bit indicates that a data frame transmission has been started.

When this bit is 0_B, data frame transmission is disabled or the downstream data registers (RHSBjDDRi) contain pending transmit data.

When this bit is 1_B, data frame transmission has been started. The downstream data registers can be updated to define the data for the next data frame.

[Clearing conditions]

- This bit is set to 0_B when last active portions of the downstream data registers (RHSBjDDRi) are written (see **Section 22.5.1.10, Data Update and Data Frame Transmission Request**).
- This bit is set to 0_B when data frame transmission gets disabled (RHSBjDTC.RHSBjDTE changes from 1_B to 0_B).
- This bit is set to 0_B when entering the CONFIG state.

[Setting conditions]

- This bit is set to 1_B when a data frame transmission starts.
- This bit is set to 1_B when data frame transmission gets enabled (RHSBjDTC.RHSBjDTE changes from 0_B to 1_B) and RHSBjDCR.RHSBjDMS is 01_B (TRIGGERED).

Emergency Signal Rising Flag (RHSBjIS.RHSBjERF)

The user can only write 0_B to this bit.

This bit indicates that an edge to the active level of the external emergency signal (RHSBjEMRG (j = 0, 1)) has been detected.

[Clearing condition]

This bit is set to 0_B when entering the CONFIG state.

[Setting conditions]

- This bit is set to 1_B when a transition from inactive to active level of the emergency signal has been detected while RHSBjGC.RHSBjOPS is ACTIVE or TEST.
- This bit is set to 1_B when there is an active level of the emergency signal when leaving the CONFIG state.

22.4 Operation

After the release of MCU's hardware reset, the RHSB module is in RESET state. The user can trigger changes of the operation state by writing the target state to RHSBjGC.RHSBjOPS.

The current operation state is indicated by RHSBjGC.RHSBjOPS.

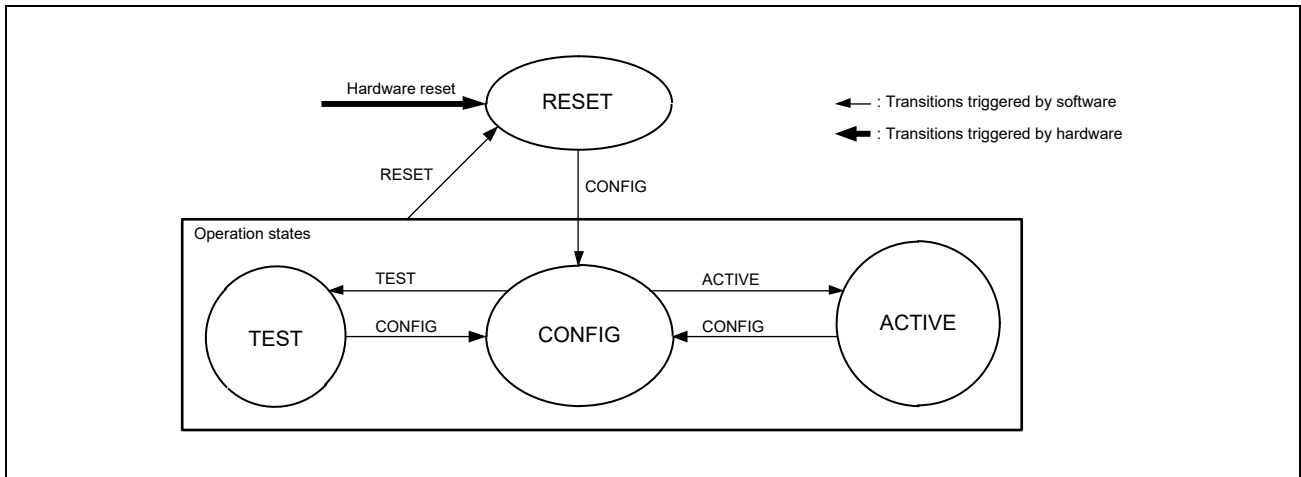


Figure 22.2 Module State Transitions

Figure 22.2 illustrates the implemented operation states and the possible state transitions. The RESET state can be entered from all operation states (TEST, CONFIG, ACTIVE).

22.4.1 Operation Description

22.4.1.1 RESET State

The RESET state is considered as the power down and reset state of the RHSB module. All internal clocks are stopped to reduce the power consumption. All configuration, control and status registers are reset.

The user can only write to RHSBjGC.RHSBjOPS to leave the RESET state. All other registers are reset and writing to these registers does not have any effect.

22.4.1.2 CONFIG State

In the CONFIG state, the user has write access to all registers to define the module configuration.

22.4.1.3 ACTIVE State

In the ACTIVE state, the RHSB module is able to perform up- and downstream communication.

Upstream communication can only be used in ACTIVE state when this function is enabled by RHSBjUCR.RHSBjUE.

Downstream communication can be individually enabled and disabled during ACTIVE and TEST state in the RHSBjDTC register.

22.4.1.4 TEST State

The TEST state is intended to support key-on testing and fault localization. By looping back the downstream data to the upstream channel, the RHSB internal data paths can be checked.

See **Section 22.5.4, Test Mode Operation** for details and restrictions of this mode.

22.4.2 Activation of the RHSB Module (Leaving RESET State)

This flow should be used to leave the RESET state. The flow assumes that the peripheral bus clock and the RHSB engine clock are already enabled.

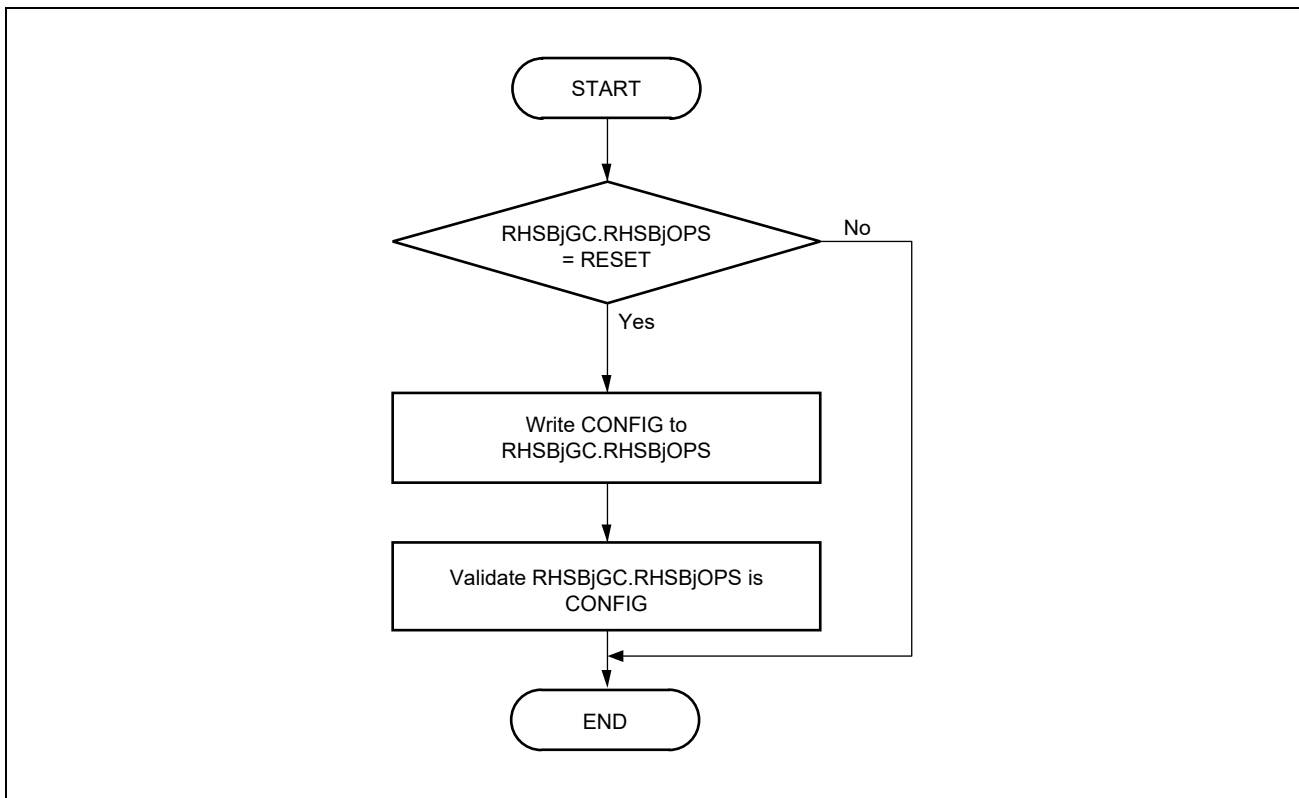


Figure 22.3 Software Flow to Leave RESET State

Note that the port function and direction registers of the MCU need to be configured before the RHSB module can perform active operation. To prevent invalid levels on the downstream chip select lines, it is recommended to define the correct active level for these lines before setting the MCU pin functions to the RHSB.

22.4.3 Deactivation of the RHSB Module (Entering RESET State)

This flow should be used to enter the RESET state. Once this flow is executed, the peripheral bus clock and the RHSB engine clock can be disabled and the functions of the MCU ports can be changed.

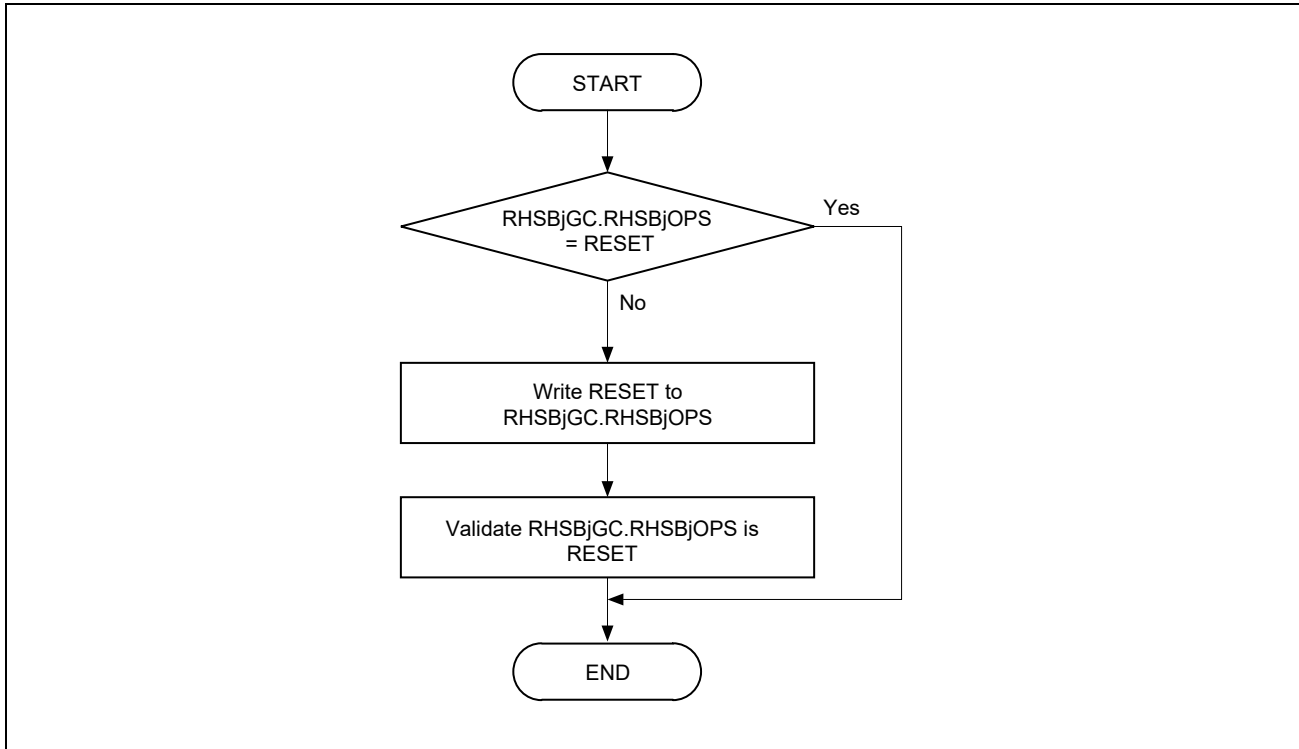


Figure 22.4 Software Flow to Enter RESET State

If the module enters the RESET state, ongoing transmission is aborted, ongoing reception is lost and the RHSB output lines are changed to their reset levels.

Note that when using the flow mentioned in **Figure 22.5**, it is possible to enter RESET state without influencing an ongoing transmission.

22.4.4 Changing between Operation States (ACTIVE, CONFIG, TEST)

Changing the operation state of the RHSB module is triggered by writing the target state to RHSBjGC.RHSBjOPS. The successful processing of the state transition is observable in RHSBjGC.RHSBjOPS.

22.4.5 Leaving ACTIVE State without Interrupting Ongoing Transmission

When the ACTIVE state is left, any ongoing transmissions are aborted. This may result in invalid reception on the downstream receiver side (slave).

Using this flow, interruption of an ongoing transmission can be prevented when changing the operation state to CONFIG or RESET.

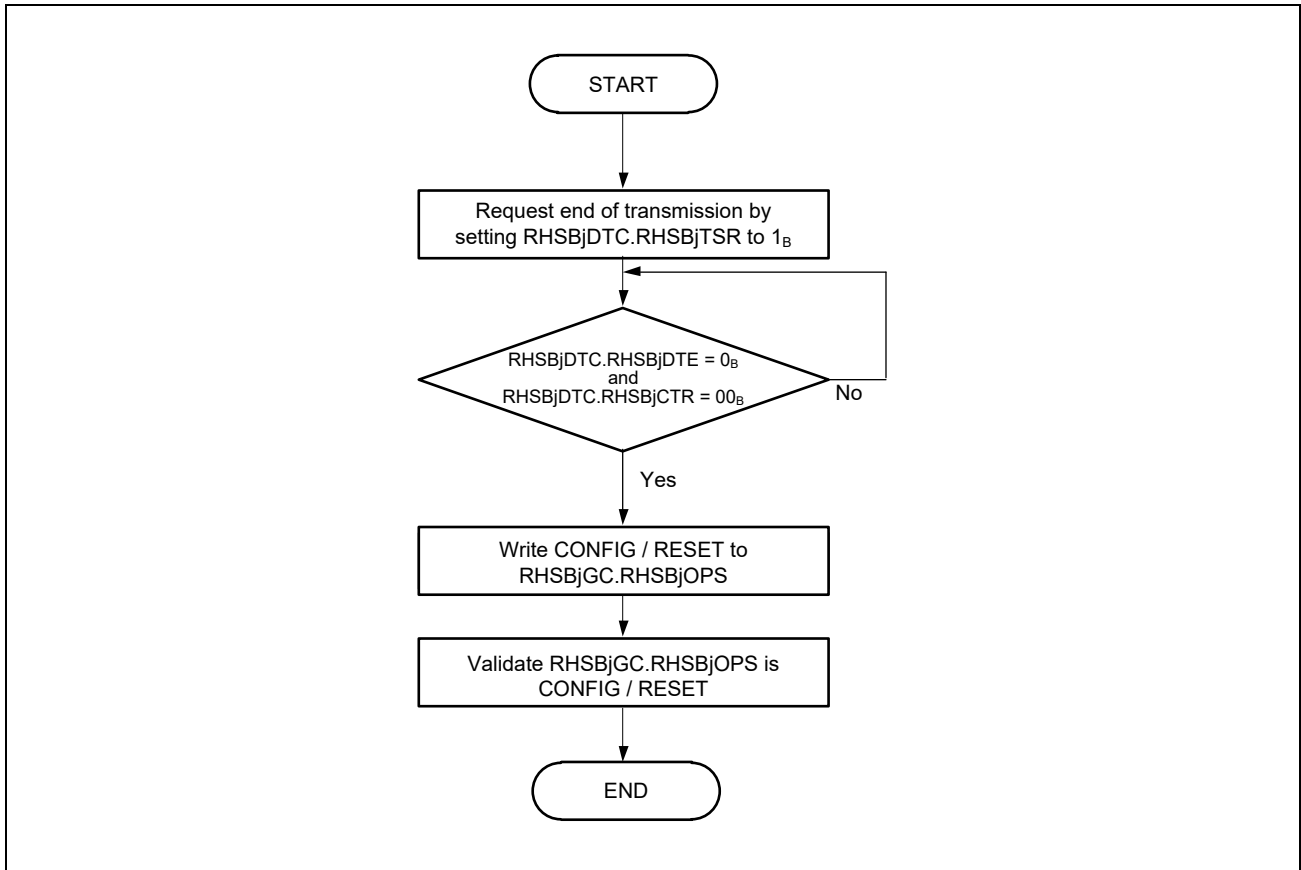


Figure 22.5 Software Flow to Leave ACTIVE/TEST State without Interruption of Ongoing Transmissions

The same flow can be used to leave the TEST state.

22.5 Operating Procedures

22.5.1 Downstream Communication

The RHSB module uses synchronous serial transmission for downstream communication.

In addition to the synchronous serial frame, the downstream communication defines a set of communication phases to serve up to two slaves by one frame and to define a repetition period for cyclic transmission.

There are different modes and configuration parameters available to schedule the downstream communication according to the application requirements. This downstream configuration is defined by the registers described in **Section 22.3.4, Downstream (Tx) Registers**.

The diagram in **Figure 22.6** illustrates the functional blocks of the downstream engine. Downstream communication is explained in detail in this section.

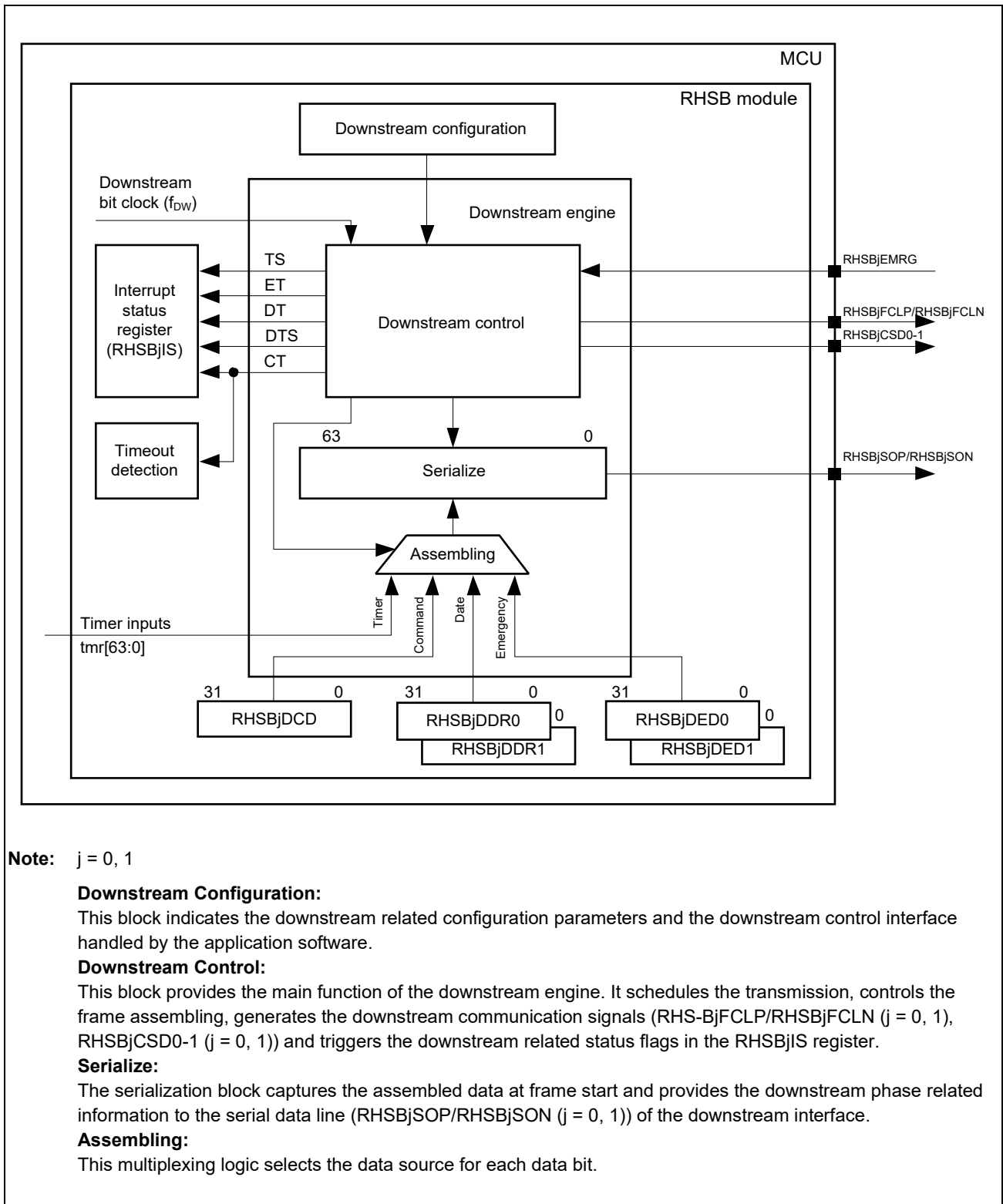


Figure 22.6 Block Diagram of Downstream Configuration

22.5.1.1 Downstream Communication Phases

The downstream communication is composed of different phases.

These phases are used by the RHSB module.

- Active phases:
 - Assertion phase:
The serial data line is invalid This optional phase allows the selected slave to get prepared for the content phase.
 - Content phase:
The serial data line is valid during this phase content is transmitted to the selected slave.
 - Deassertion phase:
The serial data line is invalid Optional phase after content is transmitted to the selected slave.
- Passive phases:
 - Frame passive phase:
This phase is part of each frame and defines a minimum time of inactivation before the next frame starts.
 - Inter-frame passive phase:
This phase describes the time, where no frame is transmitted (bus is idle).
An inter-frame passive phase occurs, depending on the downstream mode and the downstream configuration.

A frame consists of the active phases and the frame passive phase.

Figure 22.7 illustrates the phases of downstream communication. The frame shown in this figure addresses one slave (slave 0). The mode dependent clock during the passive phases is shown by dotted lines. All phases, including the optional ones, are shown in the ordering used for transmission.

Figure 22.8 illustrates the phases of downstream communication of a frame addressing two slaves (slave 0 and slave 1). The optional assertion and deassertion phases are individually configurable for each slave.

In **Figure 22.7** and **Figure 22.8**, the select signal of the slaves are configured as active high; the clock line phase is set to rising edge.

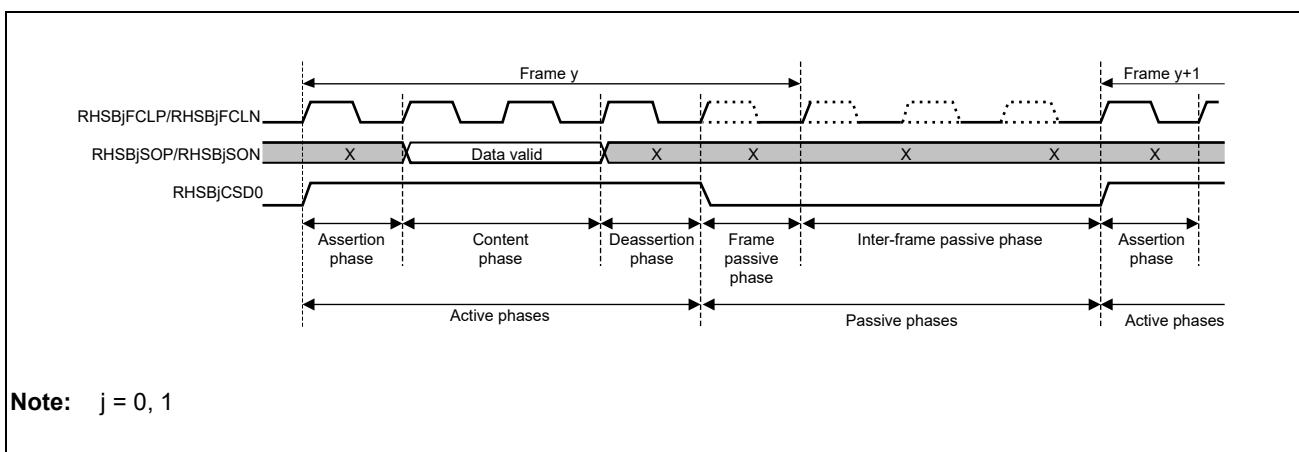


Figure 22.7 Downstream Communication Phases

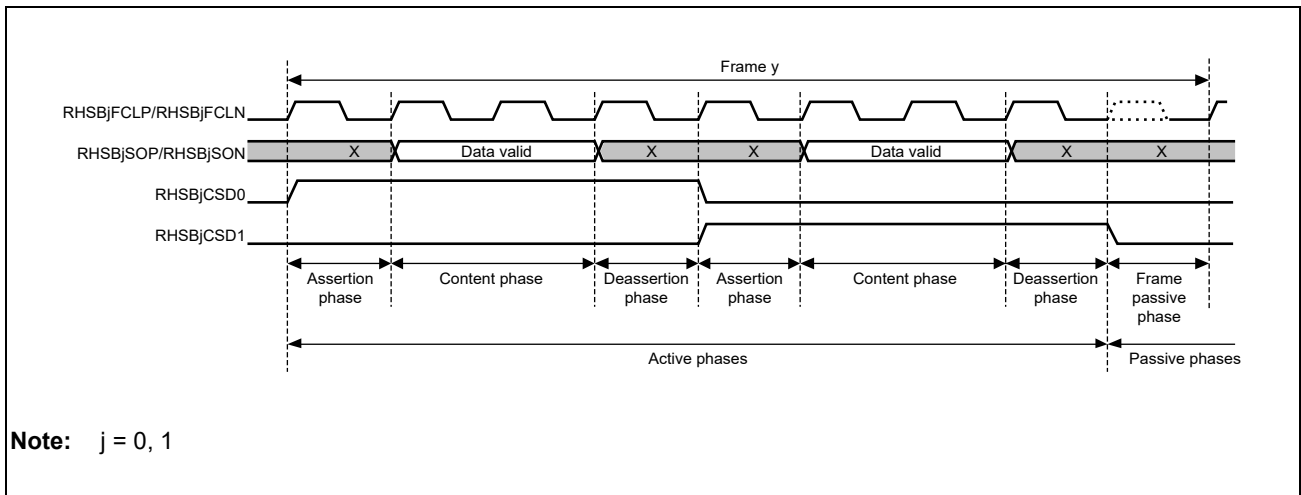


Figure 22.8 Downstream Communication Phases of a Frame Addressing Two Slaves

For evaluation purposes, the user can monitor the communication phases by reading RHBjMSR. Note that this function is not available for all downstream bit rates.

22.5.1.2 Frame Dependent Flagging

Figure 22.9 illustrates the RHSB internal processing flow of command and data frame transmission. The scheduling when a command or data frame is transmitted is mode dependent and explained in **Section 22.5.1.3, Downstream Modes**.

The flows described in **Figure 22.8** are used as macros in the mode description of **Section 22.5.1.3, Downstream Modes**.

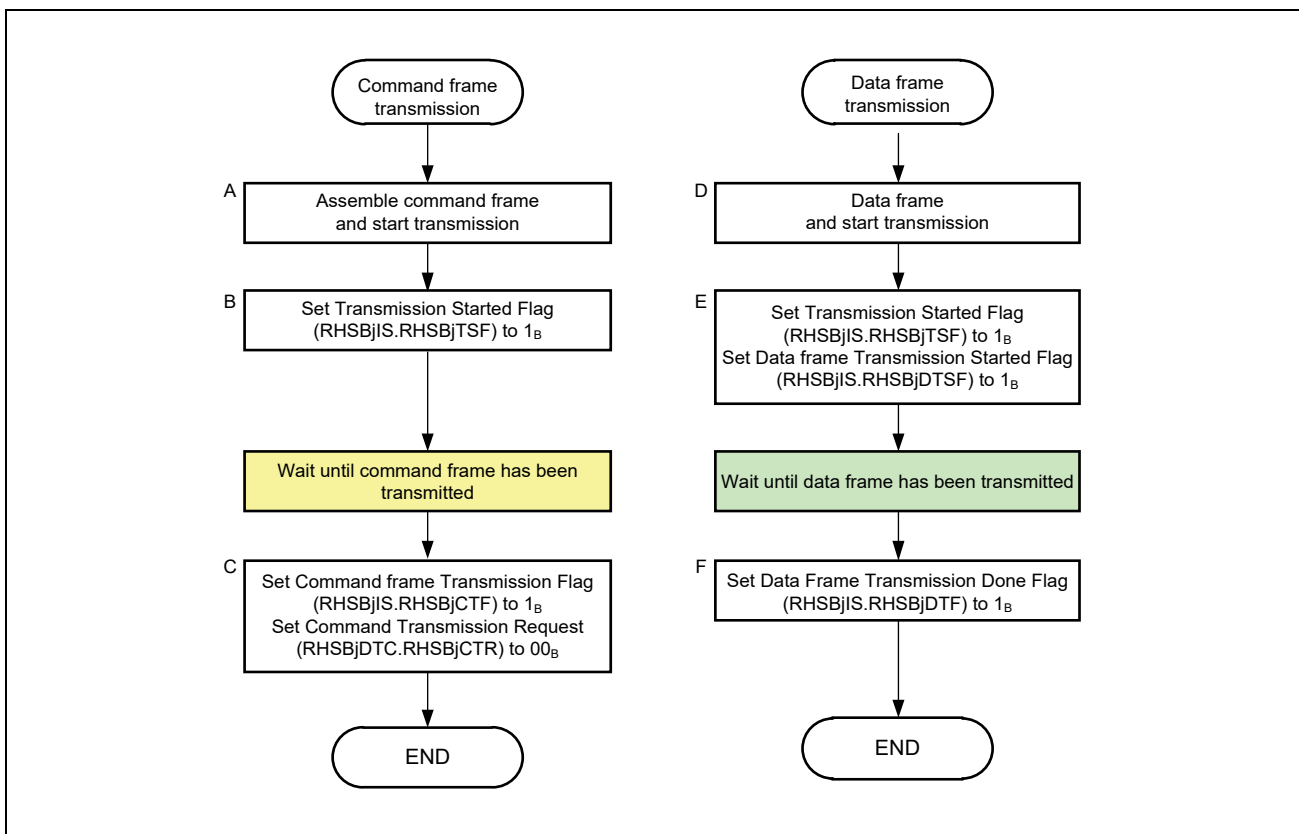


Figure 22.9 Internal Transmission Processing of Command and Data Frames

The flagging related to command and data frame transmission is illustrated in **Figure 22.10**. A gray background illustrates values that depend on previous user interaction but have no influence on the transmission.

In triggered mode, RHSBjIS.RHSBjDTSF is always 0_B immediately before a data frame transmission starts. In other modes, the value can be 0_B or 1_B.

The timing marks A to F refer to the marks in **Figure 22.9**.

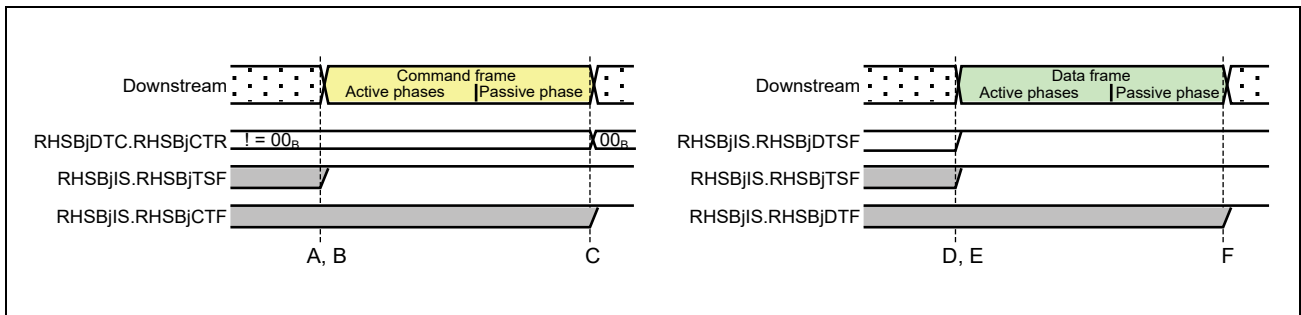


Figure 22.10 Command and Data Frame Related Flagging

When an emergency frame is transmitted, the RHBjIS.RHBjETF is rising at the same time as RHBjIS.RHBjDTF (timing F).

22.5.1.3 Downstream Modes

The RHSB module supports different downstream transmission modes that can be selected during CONFIG state.

These modes are available and can be configured in RHSBjDCR.RHSBjDMS:

- Triggered mode

The RHSB module starts downstream communication immediately on user request (software triggered).

- Single-period repetition mode

The RHSB module starts downstream communication in a defined repetition period. The cyclic transmission is independent from user interaction.

- Multi-period repetition mode

The RHSB module starts downstream communication in two defined repetition periods. The cyclic transmission is independent from user interaction.

In all modes, transmission of data frames is only possible when the data transmission is enabled by setting RHSBjDTC.RHSBjDTE to 1_B.

Except in triggered mode, it is recommended to initialize the downstream data registers with valid data before enabling data transmission because a data frame transmission is immediately started when RHSBjDTC.RHSBjDTE changes from 0_B to 1_B.

The RHSB module is able to transmit command frames independent from the downstream mode and regardless of whether data transmission is enabled.

(1) Triggered Mode

In this mode, transmission of data frames is triggered by the user. Each trigger starts the transmission of one data frame.

There is no repetition period defined for this mode, the transmission is controlled by the software execution timing.

These configuration parameters are ignored in triggered mode:

- Command frame insertion mode (RHSBjDCR.RHSBjCIM)
- Command frame transmission delay (RHSBjDCR.RHSBjCTD)
- Repetition period length (RHSBjDCR.RHSBjREP)

A data frame transmission is triggered when RHSBjIS.RHSBjDTSF is 0_B. This bit can be cleared by writing 0_B to RHSBjIS.RHSBjDTSF (keep data unchanged) or by updating the downstream data registers (RHSBjDDRi). See **Section 22.5.1.10, Data Update and Data Frame Transmission Request** for details about data update and data frame transmission request.

The flow shown in **Figure 22.11** assumes that the RHSB module is in triggered mode and that the data transmission is enabled.

The details about command and data frame transmission (colored ovals) are shown in **Figure 22.9**.

The flow is started when the following configuration is applied:

- RHSBjGC.RHSBjOPS is ACTIVE
- RHSBjDCR.RHSBjDMS is triggered
- RHSBjDTC.RHSBjDTE is 1_B

The flow is immediately aborted when the RHSB module leaves the ACTIVE state.

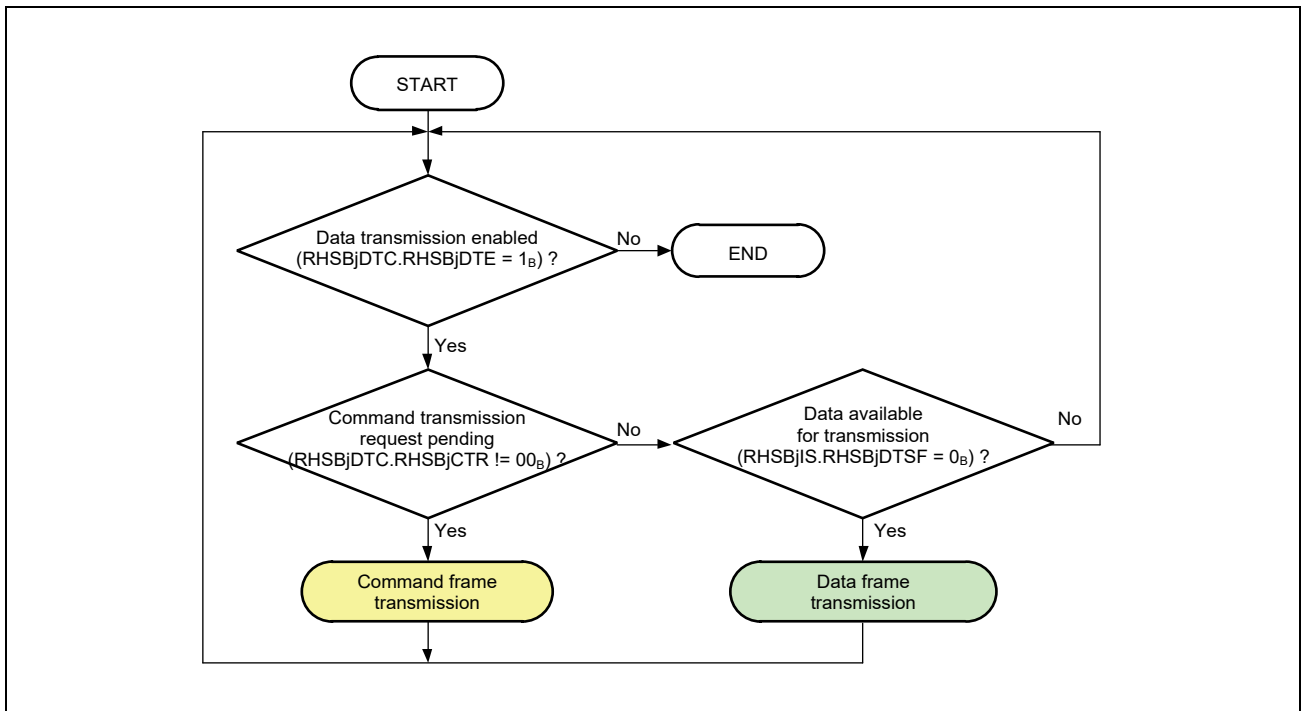


Figure 22.11 Internal Transmission Processing in Triggered Mode when Data Transmission is Enabled

Figure 22.12 illustrates the downstream communication in case of concurrent command and data frame transmission requests. If there is a pending request when a frame transmission ends, the next frame is transmitted immediately.

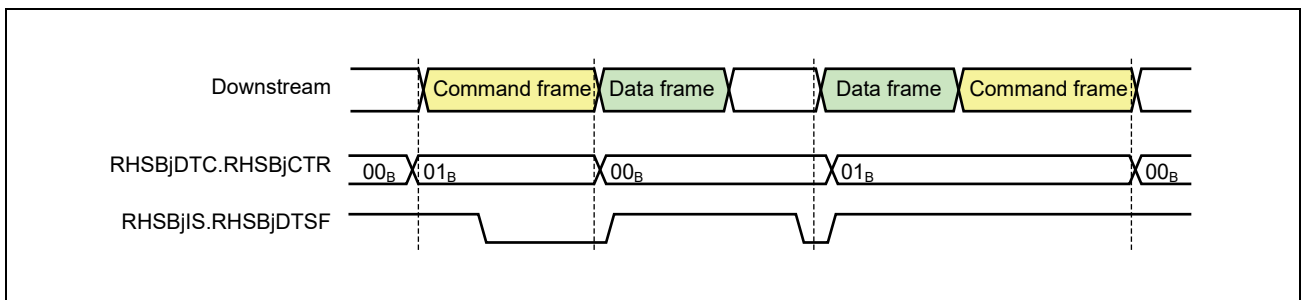


Figure 22.12 Example of Concurrent Command/Data Transmission Requests in Triggered Mode

(a) Command Frame Transmission in Triggered Mode

Command frame transmission has higher priority than data frame transmission as shown in the transmission processing figure. The user needs to schedule data and command frame transmission to be consistent with the requirements of the application.

When the data transmission is disabled (RHSBjDTC.RHSBjDTE = 0B) a command frame is immediately transmitted when the request is pending. This command frame transmission is not controlled by the internal transmission processing as shown in **Figure 22.11**.

(2) Repetition Mode (Single-Period Repetition, Multi-Period Repetition)

In this mode transmission of data frames is done periodically.

A periodical time tick defines the repetition period of the cyclic data transmission.

The fixed repetition period used in this mode is controlled by the RHSB module.

Using RHSBjDTC.RHSBjDTE and RHSBjDTC.RHSBjTSR, the user can start and stop the cyclic data transmission in repetition mode. The RHSB module ensures that disabling of data transmission will not interrupt an ongoing transmission.

In repetition mode, the trigger of data frame transmission is independent from RHSBjIS.RHSBjDTSF.

However the values from the downstream data registers (RHSBjDDRi) are used to assemble the transmission data.

(a) Repetition Period Definition

The repetition period is defined by RHSBjDCR.RHSBjREP. The minimum repetition period should be at least the data frame length.

When the repetition period is bigger than the data frame length, there is an inter-frame passive phase between two data frames as illustrated in **Figure 22.13**.

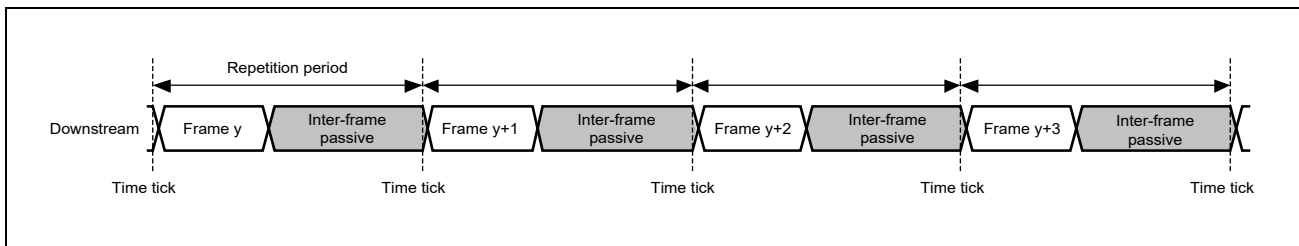


Figure 22.13 Transmission Timing in Repetition Mode

The length of a data frame can be calculated as the sum of these configuration parameters:

- Assertion phase length (RHSBjSDCi.RHSBjAPLn) of all addressed slaves
- Data frame selection bit (RHSBjSDCi.RHSBjCPSn) of all addressed slaves
- Deassertion phase length (RHSBjSDCi.RHSBjDPLn) of all addressed slaves
- Number of data bits (RHSBjDEC.RHSBjNDBm) of all used DFTEs (RHSBjDCR.RHSBjSLS and RHSBjDCR1.RHSBj.SLS1 (multi-period repetition mode))
- Data frame passive phase length (RHSBjDCR.RHSBjDFP)

The user should not configure a repetition period (RHSBjDCR.RHSBjREP) less than 8 peripheral bus clock cycles. As example: With a typical peripheral bus clock (CLK_LSB) frequency of 40 MHz, this results in a minimal repetition period of 0.2 μ s.

The transmission period count in single-period repetition mode holds “period 0” and that in multi-period repetition mode toggles between “period 0” and “period 1” repeatedly as illustrated in **Figure 22.13**.

The sequence length in single-period repetition mode is defined by RHSBjDCR.RHSBjSLS for period 0. The sequence length in multi-period repetition mode is defined by RHSBjDCR.RHSBjSLS for period 0 and RHSBjDCR1.RHSBjSLS1 for period 1.

(b) Command Frame Transmission

Command frame transmission has higher priority than data frame transmission. Depending on configuration parameter RHSBjDCR.RHSBjCTD, transmission of consecutive command frames can be prevented.

Different command frame insertion methods are available. They differ in the strategy for the case when a command frame is transmitted and how data transmission is affected by command frame insertion. All insertion methods (controlled by RHSBjDCR.RHSBjCIM) are explained in detail in **Section 22.5.1.3(3), Command Frame Insertion Methods in Single-Period Repetition Mode.**

(3) Command Frame Insertion Methods in Single-Period Repetition Mode

In single-period repetition mode, a request for command transmission comes asynchronously with respect to the repetition period. The RHSB module supports different strategies controlled by RHSBjDCR.RHSBjCIM, on how to insert command frames into the repetition schedule.

If data transmission is disabled (RHSBjDTC.RHSBjDTE is 0_B), a command frame is immediately transmitted when the request is pending. In addition, the restriction about consecutive command frames (RHSBjDCR.RHSBjCTD is 1_B) is ignored.

(a) Time-Slot Method

When this method is selected, the RHSB module processes command and data frames aligned to the time ticks. The command frames substitute data frames within the repetition period.

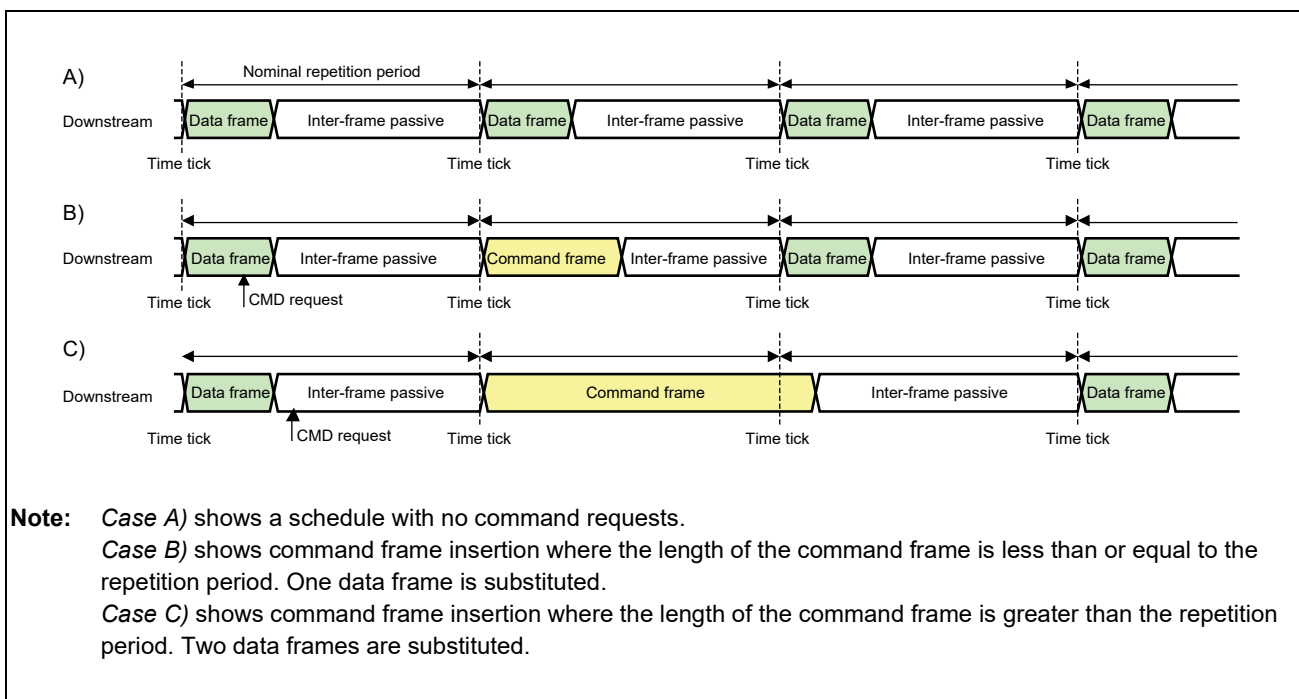


Figure 22.14 Examples of Command Insertion Using Time-Slot Method

(b) Immediate Method

When this method is selected, the RHSB module processes a command frame transmission request as soon as possible (immediately during inter-frame passive phase or when a data frame ends). The command frame transmission is not aligned to the repetition period.

If a data frame transmission cannot be started at the origin time tick because a command frame has broken the repetition period, the data transmission is processed as soon as possible. The repetition period is re-adjusted.

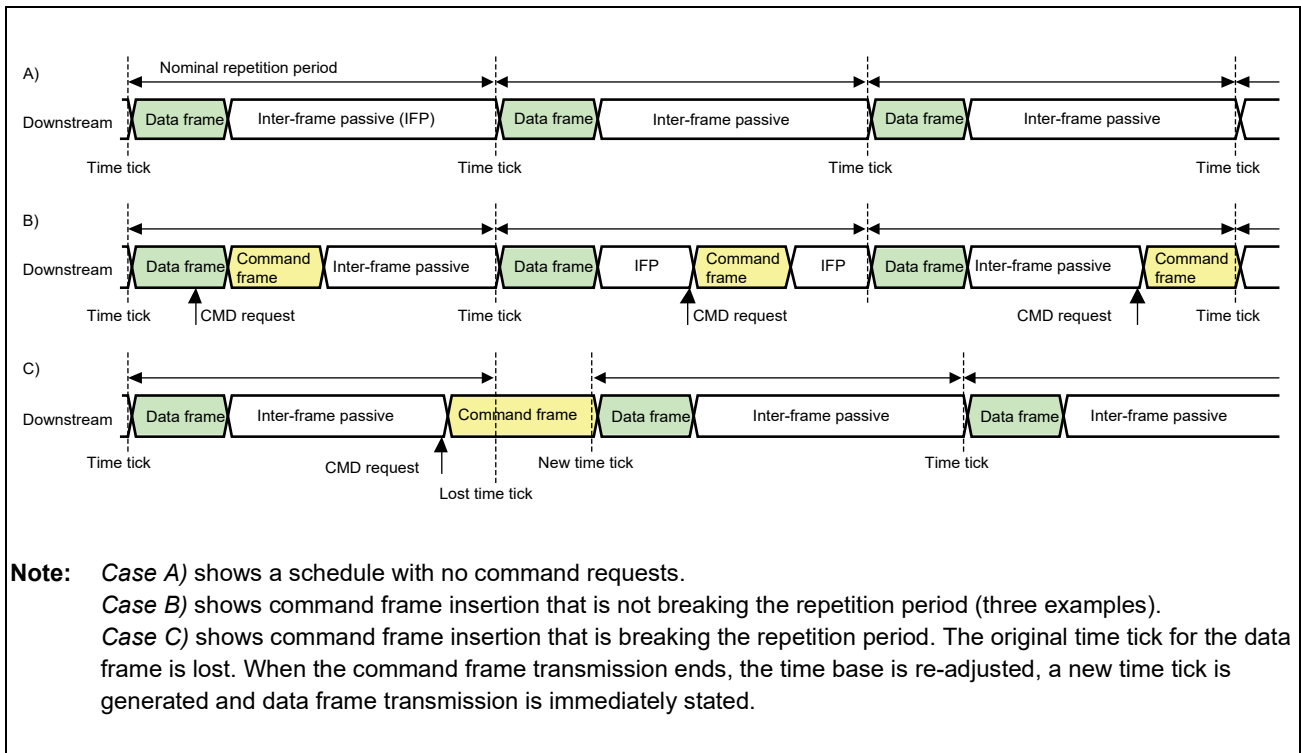


Figure 22.15 Examples of Command Insertion Using Immediate Method

It is possible to insert more than one command frame between two data frames if this is allowed from the configuration. Also in this case a re-adjust of the repetition period happens if a command frame breaks the repetition period.

(c) Best-Effort Method

When this method is selected, the RHSB module starts command frame transmission only

- At the end of a data frame transmission or
- When the command frame transmission will end within the inter-frame passive phase.

The repetition time grid is not adjusted when using the best-effort insertion method.

If the command frame is shorter than or equal to the inter-frame passive phase, command frame transmission has no influence on the data frame transmission (see case B) in **Figure 22.16**). If possible from configuration, more than one command frame can be transmitted during the repetition period.

If the command frame is longer than the inter-frame passive phase, command frame transmission influences the data frame transmission (see case C) in **Figure 22.16**). The data transmission is processed as soon as possible.

If a data frame transmission is influenced by another data frame (shifted by command frame transmission), the data frame is not processed (see case D) in **Figure 22.16**).

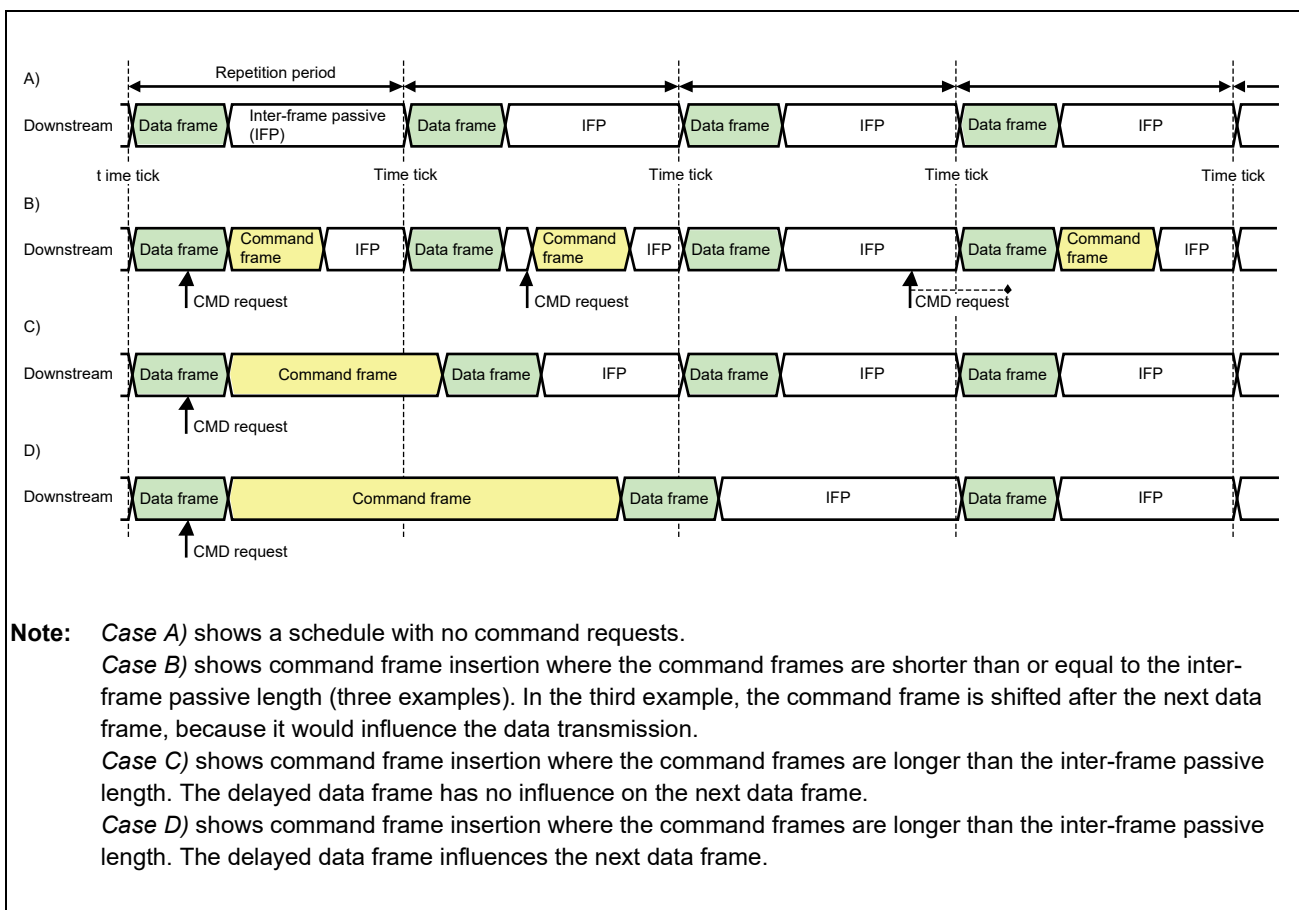


Figure 22.16 Examples of Command Insertion Using Best-Effort Method

22.5.1.4 Command Frame Insertion Methods in Multi-Period Repetition Mode

In Multi-Period repetition mode, a request for command transmission comes asynchronously with respect to the repetition period.

(1) Best-Effort Method

Multi-Period repetition mode supports only Best-effort method. When this method is selected, the RHSB module starts command frame transmission only

- At the end of a data frame transmission in period 1.

The repetition time grid is not adjusted when using the best-effort insertion method.

If the command frame is shorter than or equal to the inter-frame passive phase in period 1, command frame transmission has no influence on the data frame transmission (see cases B), C), D), E), F) and G) in **Figure 22.17**). If possible from configuration, more than one command frame can be transmitted during the repetition period.

If the command frame is longer than the inter-frame passive phase, command frame transmission influences the data frame transmission (see case H) in **Figure 22.17**). The transferred command and data are not guaranteed.

If a data frame transmission is influenced by another data frame (shifted by command frame transmission), the data frame is not processed (see case I) in **Figure 22.17**). The transferred command and data are not guaranteed.

The processing of the best-effort method is also mentioned in transmission processing flows as shown in **Figure 22.17**.

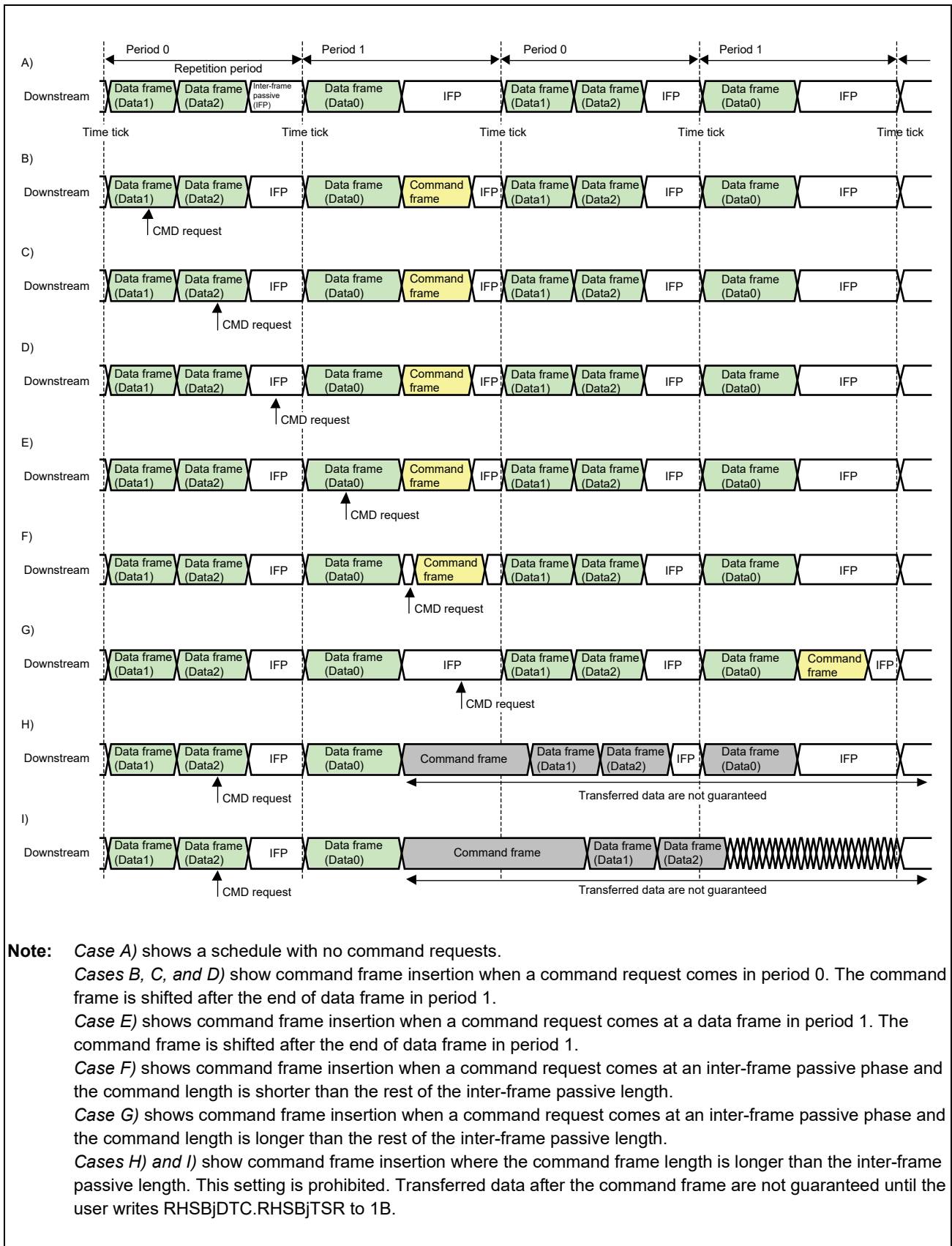


Figure 22.17 Examples of Command Insertion Using Best-Effort Method in Multi-Period Repetition Mode

22.5.1.5 Physical Frame Format

(1) Clock Line

The serial clock line (RHSBjFCLP/RHSBjFCLN ($j = 0, 1$)) provides the timing information from the MSC master to the MSC slave modules. The clock line and the clock phase are common for all slaves.

The user can configure by setting RHSBjDCR.RHSBjCLP whether the serial data line (RHSBjSOP/RHSBjSON ($j = 0, 1$)) and the chip select lines (RHSBjCSD0-1 ($j = 0, 1$)) are changing their values with rising edge or falling edge of the serial clock line (RHSBjFCLP/RHSBjFCLN ($j = 0, 1$)).

Note that RHSBjDCR.RHSBjCLP is directly influencing the level on the serial clock line (RHSBjFCLP/RHSBjFCLN ($j = 0, 1$)).

The user is able to configure the serial clock activity by RHSBjDCR.RHSBjCAC anytime. The RHSB module guarantees a glitch free clock signal on the serial clock line (RHSBjFCLP/RHSBjFCLN ($j = 0, 1$)).

The configuration given by RHSBjDCR.RHSBjCAC is not affecting the clock activity during the active phases; for proper downstream communication, the clock is required during active phases. RHSBjDCR.RHSBjCAC only affects clock activity during the passive phases; for downstream communication, the clock is not required during passive phases.

An active serial clock line may be required in some applications by a connected slave (e.g. if a slave uses the downstream clock to handle the upstream communication).

(2) Individual Slave Configuration

For each slave, the user can independently program in CONFIG state:

- The active level of the chip select line (RHSBjSDCi.RHSBjCSLPn)
- The active level of the serial data line (RHSBjSDCi.RHSBjSOLPn)
- The assertion phase length (RHSBjSDCi.RHSBjAPLn)
- The deassertion phase length (RHSBjSDCi.RHSBjDPLn)
- The presence of a selection bit in data frames (RHSBjSDCi.RHSBjCPSn)

Note that the active level configuration for the chip select lines is directly visible at the outputs of the RHSB module.

(3) Downstream Frame Encoding

The selection bit is 0_B for data frames and 1_B for command frames.

The serial data line (RHSBjSOP/RHSBjSON ($j = 0, 1$)) is specified as 'invalid' for the assertion and deassertion phases; the RHSB module is transmitting 1_B during these phases. Note that the level on the serial data line depends on the individual slave configuration (RHSBjSDCi.RHSBjSOLPn).

An enabled inversion of the serial data line happens as long as slave n is selected.

The inversion of the serial data line is only configurable for active phases; the RHSB module sets serial data line (RHSBjSOP/RHSBjSON ($j = 0, 1$)) to high level during passive phases independent from RHSBjSDCi.RHSBjSOLPn.

Figure 22.18 illustrates a data frame encoding in the case of two slaves with active high chip select lines. The frame transmits a data word of 3 bits (value is 03_H) to slave 0 and to slave 1. The configuration of both slaves is identical except that the data line for slave 1 is inverted.

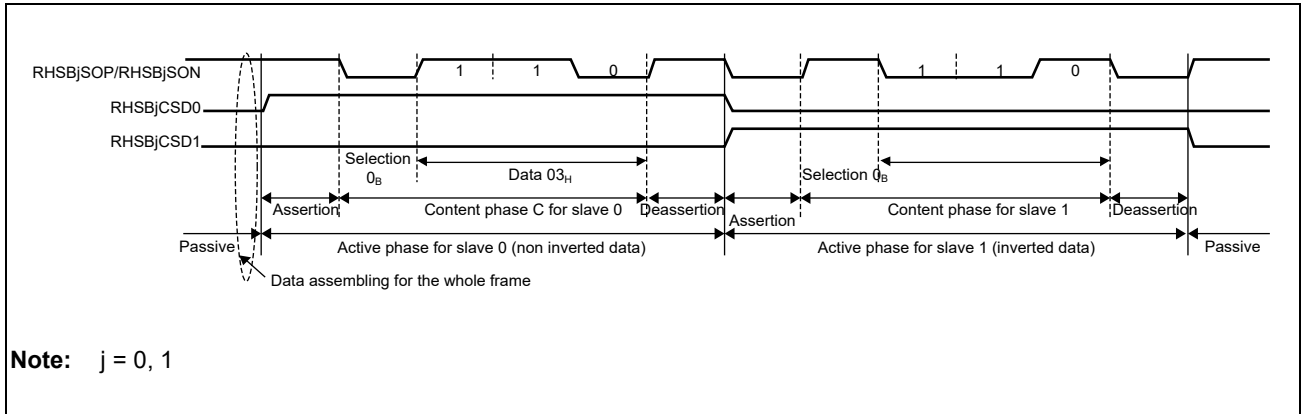


Figure 22.18 Frame Encoding Example

The RHSB module assembles the downstream frame directly before the transmission starts (as marked in **Figure 22.18**). This assembling timing is independent from the downstream mode (RHSBjDCR.RHSBjDMS) and the frame type (command/data frame).

22.5.1.6 Data Frame Assembling

A data frame is a stream of un-correlated bits sent to up to 2 slaves. There is an individual chip select line for each slave indicating which content is for which slave. Two chip select signals are not simultaneously active. Only one chip select signal can be activated at a time.

A data frame is composed by up to 4 downstream frame transmission elements (DFTEs).

The number of DFTEs used is defined by RHSBjDCR.RHSBjSLS.

For each DFTE, the user can configure:

- The slave assigned to (RHSBjDEC.RHSBjSSDm)
- Presence of an optional selection bit in each content phase (RHSBjSDCi.RHSBjCPSn)
- The number of used data bits (RHSBjDEC.RHSBjNDBm)

If a slave needs more than one DFTE, the user should assign consecutive DFTEs to the same slave. These DFTEs are combined into one content phase.

Note that it is possible to assign nonconsecutive DFTEs to the same slave (e.g. DFTE0 and DFTE2); in this case, each DFTE defines its own content phase.

The DFTEs are always placed in ascending order inside a data frame. Because the DFTE/slave assignment is freely configurable, the ordering of slave selection is not restricted by the fixed DFTE order.

(1) Data Source Configuration

For each bit of a DFTE the user can configure independently the data source by setting `RHSBjDEBAm.RHSBjDSSn` as:

- Timer input (tmr)
- Timer input inverted
- Downstream data register

The relation between DFTEs and the data sources is shown in **Table 22.31**. The diagram in **Figure 22.19** illustrates this relation for one bit and includes the emergency case.

For details about the mapping of timer inputs see **Section 22.6, Cross Bar (XBAR)**.

Table 22.31 Relation of DFTEs and Data Sources

	Timer Input	Data Input	Bit Assignment	Emergency Data	Emergency Enable
DFTE0[15:0]	tmr[15:0]	DDR0[31:16]	DEBA0[31:0]	DED0[31:16]	EBE0[31:16]
DFTE1[15:0]	tmr[31:16]	DDR0[15:0]	DEBA1[31:0]	DED0[15:0]	EBE0[15:0]
DFTE2[15:0]	tmr[47:32]	DDR1[31:16]	DEBA2[31:0]	DED1[31:16]	EBE1[31:16]
DFTE3[15:0]	tmr[63:48]	DDR1[15:0]	DEBA3[31:0]	DED1[15:0]	EBE1[15:0]

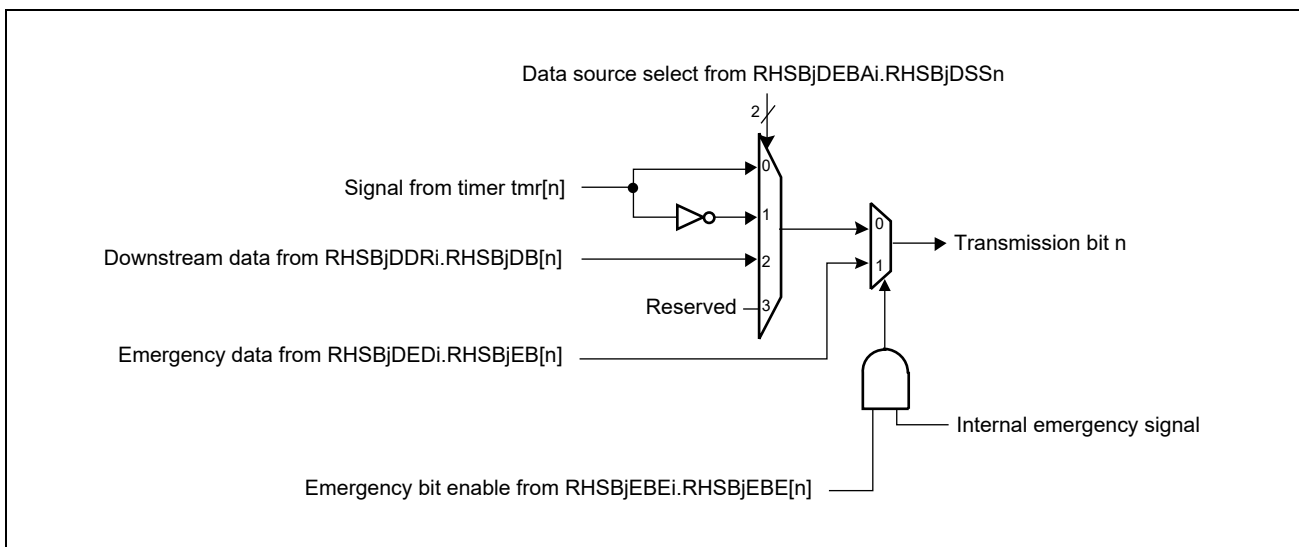


Figure 22.19 Data Source Selection for One Bit of a DFTE

The ‘transmission bit n’ signal in **Figure 22.19** indicates the value of bit n. This signal is used by the RHSB module to assemble the content of a data frame when starting the data frame transmission.

(2) Data Frame DFTE Mapping

Table 22.32 shows all possible configurations of data bits available for each slave depending on the number of used slaves. The slaves are named A to D to illustrate that there is no dependency between the DFTEs and slave number. If there is a range of assigned DFTEs given, these DFTEs have to be consecutive.

Table 22.32 Combinations of DFTEs and Data Bits Assignments to Slaves

Number of Slaves	Slave A Bits	Slave B Bits	Slave C Bits	Slave D Bits	DFTE Assignment
1 slave (A)	1 to 64	—	—	—	1 to 4 DFTEs to slave A
2 slaves (A, B)	1 to 48	1 to 16	—	—	1 to 3 DFTEs to slave A 1 DFTE to slave B
2 slaves (A, B)	1 to 32	1 to 32	—	—	1 to 2 DFTEs to slave A 1 to 2 DFTEs to slave B

Figure 22.20 shows an example of a downstream frame to be transmitted to slave A assembled from DFTE1 and DFTE2. The slave device A is configured in RHSBjSDCi register as: RHSBjAPLn = 1, RHSBjCSLPn = 0, RHSBjSOLPn = 0, RHSBjCPSn = 1, RHSBjDPLn = 1 (active high, assertion and deassertion phase of 1_D bit, selection bit is present). For DFTE1, 3 data bits are configured (RHSBjDEC.RHSBjNDB1 = 2), and for DFTE2, 13 data bits are configured (RHSBjDEC.RHSBjNDB2 = 12).

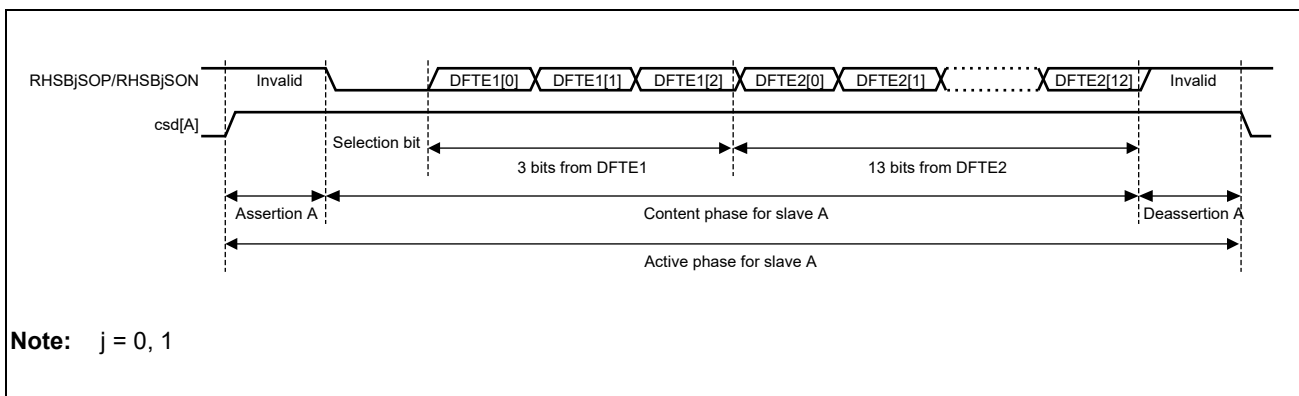


Figure 22.20 Example Bit Mapping when Combining Two DFTEs

22.5.1.7 Command Frame Assembling

The command frame is a stream of correlated bits sent to one slave. There is only one content phase available in a command frame.

The content phase of a command frame always contains a selection bit.

The number of data bits that are sent in the content phase from the RHSBjDCD register is configurable from 1 to 32 bits. This number can be changed individually for each command frame transmission by RHSBjDTC.RHSBjNCB.

The frame passive phase of command frames is fixed to 2D bits.

The length of assertion and deassertion phase depends on the slave device configuration in RHSBjSDCi register as used for data frames.

Figure 22.21 shows an example of a command frame with 5 bits (RHSBjDTC.RHSBjNCB = 4) to be transmitted to slave A. The slave device A is configured in RHSBjSDCi register as: RHSBjAPLn = 1, RHSBjCSLPn = 0, RHSBjSOLPn = 0, RHSBjDPLn = 1 (active high, assertion and deassertion phase of 1D bit).

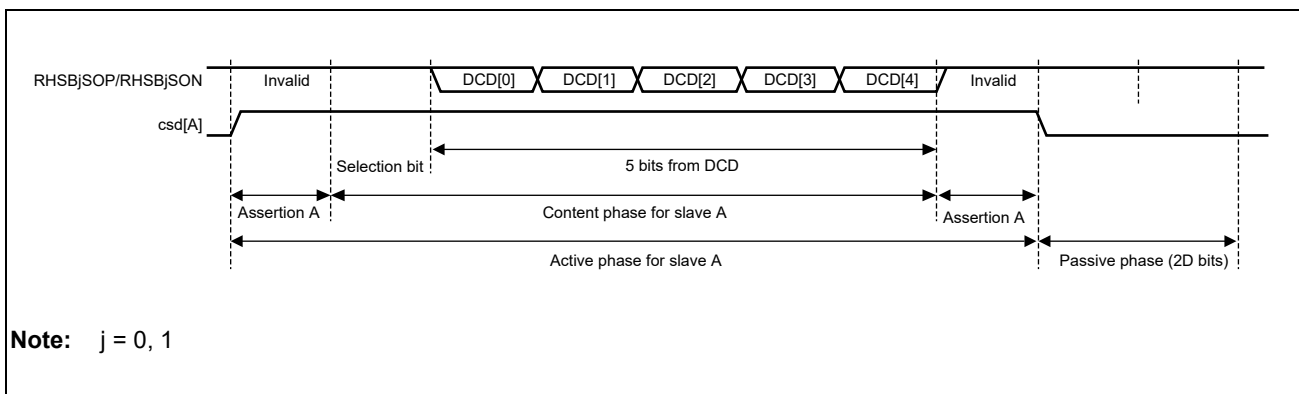


Figure 22.21 Command Frame Bit Mapping

22.5.1.8 Emergency Function

The emergency function of the RHSB modules supports an automatic transmission of pre-defined data to slaves in case of emergency. The emergency function controls only the content of data frames; the scheduling of data transmission is not affected.

The external emergency signal (RHSBjEMRG ($j = 0, 1$)) which is available as primary input pin of the MCU, is evaluated to derive the emergency condition.

The emergency function can be globally enabled/disabled using RHSBjDCR.RHSBjEE. Additionally, the user can individually select via RHSBjEBEi.RHSBjEBE for each bit of each DFTE if there should be any action taken in an emergency case.

When the RHSB module detects the emergency condition, the data source of the selected DFTE bits is forced to the corresponding value defined in the downstream emergency data registers (RHSBjDEDi). This selection is illustrated in **Figure 22.17**.

To use the emergency function, the user should:

- Configure the active level of the emergency signal (RHSBjDCR.RHSBjEIP).
- Configure the mode indicating how the emergency condition is derived from the emergency signal (RHSBjDCR.RHSBjEIM).
- Enable the emergency function and define the usage of automatic stop capability (RHSBjDCR.RHSBjEE).
- Select the action in an emergency case for the individual DFTE bits (RHSBjEBEi.RHSBjEBE).
- Define the emergency data for the individual DFTE bits (RHSBjDEDi.RHSBjEB).

Figure 22.22 illustrates how the emergency condition (internal emergency signal) is derived from the level of the external emergency signal (RHSBjEMRG ($j = 0, 1$)). The RHSB module requires stable levels on the external emergency signal that are longer than three peripheral bus clock cycles (CLK_LSB).

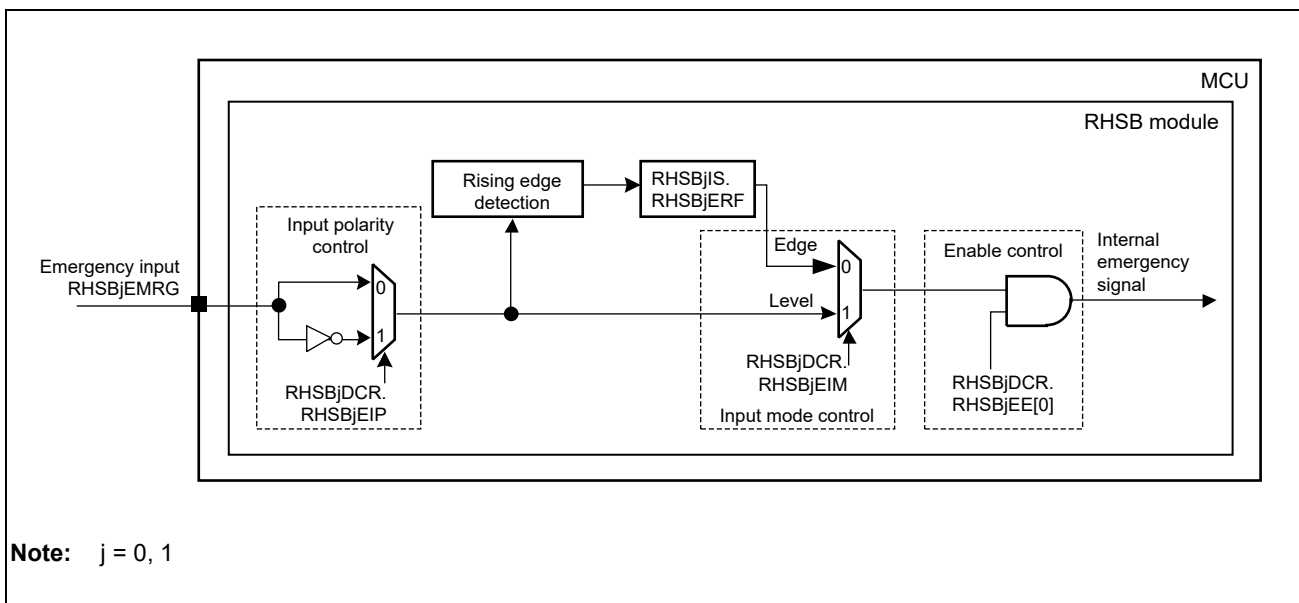


Figure 22.22 Detection of Emergency Condition

The RHSB module uses the internal emergency signal when assembling a data frame. To transmit an emergency frame the internal emergency signal has to be active at assembling time (when the data frame starts).

Even if the emergency function is not used (RHSBjDCR.RHSBjEE is 00_B), the user can read RHSBjIS.RHSBjERF to get informed about the emergency case detection.

(1) Edge Sensitive Input Mode

The emergency input mode is edge sensitive when RHSBjDCR.RHSBjEIM is 0_B.

When the RHSB module detects an edge where the external emergency signal goes active, the RHSBjIS.RHSBjERF flag is set to 1_B.

This flag needs to be cleared by the user.

When using the edge sensitive input mode, the user must ensure that the RHSBjIS.RHSBjERF flag is 1_B until an emergency frame is transmitted. The RHSBjIS.RHSBjETF flag indicates that the emergency frame has been transmitted.

Note that an edge will be detected when the ACTIVE state is entered while there is an active level at the external emergency signal.

(2) Level Sensitive Input Mode

The emergency input mode is level sensitive when RHSBjDCR.RHSBjEIM is 1_B.

When using the level sensitive input mode, the user must ensure that the external emergency signal is on the active level for one repetition period or longer in single-period repetition mode or for two repetition periods or longer in multi-period repetition mode. If shorter, the RHSB module may not detect emergencies.

(3) Automatic Stop Function

When the emergency function is enabled by RHSBjDCR.RHSBjEE, the user defines if the data transmission should be automatically stopped after the transmission of the first emergency frame.

When RHSBjDCR.RHSBjEE is set to 01_B (emergency condition detection enabled, no stop), the automatic stop function is inactive. The RHSB module transmits emergency frames as long as the emergency condition (see **Figure 22.22**) is true.

When the emergency condition changes to false, the RHSB module continues with transmission of normal data frames.

When RHSBjDCR.RHSBjEE is set to 11_B (emergency condition detection and automatic stop enabled), the RHSB module transmits exactly one emergency frame in single-period repetition mode or the RHSB module transmits one or two emergency frames in multi-period repetition mode when the emergency condition is true.

After transmitting the emergency frame to all slaves, the data transmission is automatically stopped (RHSBjDTC.RHSBjDTE changes to 0_B) after period 0 as illustrated in **Figure 22.23**. Independent from the downstream mode (RHSBjDCR.RHSBjDMS), no data transmission is possible until the user enables this by setting DTC.DTE to 1_B. After setting DTC.DTE to 1_B, the RHSB module restarts transmission of DFTE0 data from period 0.

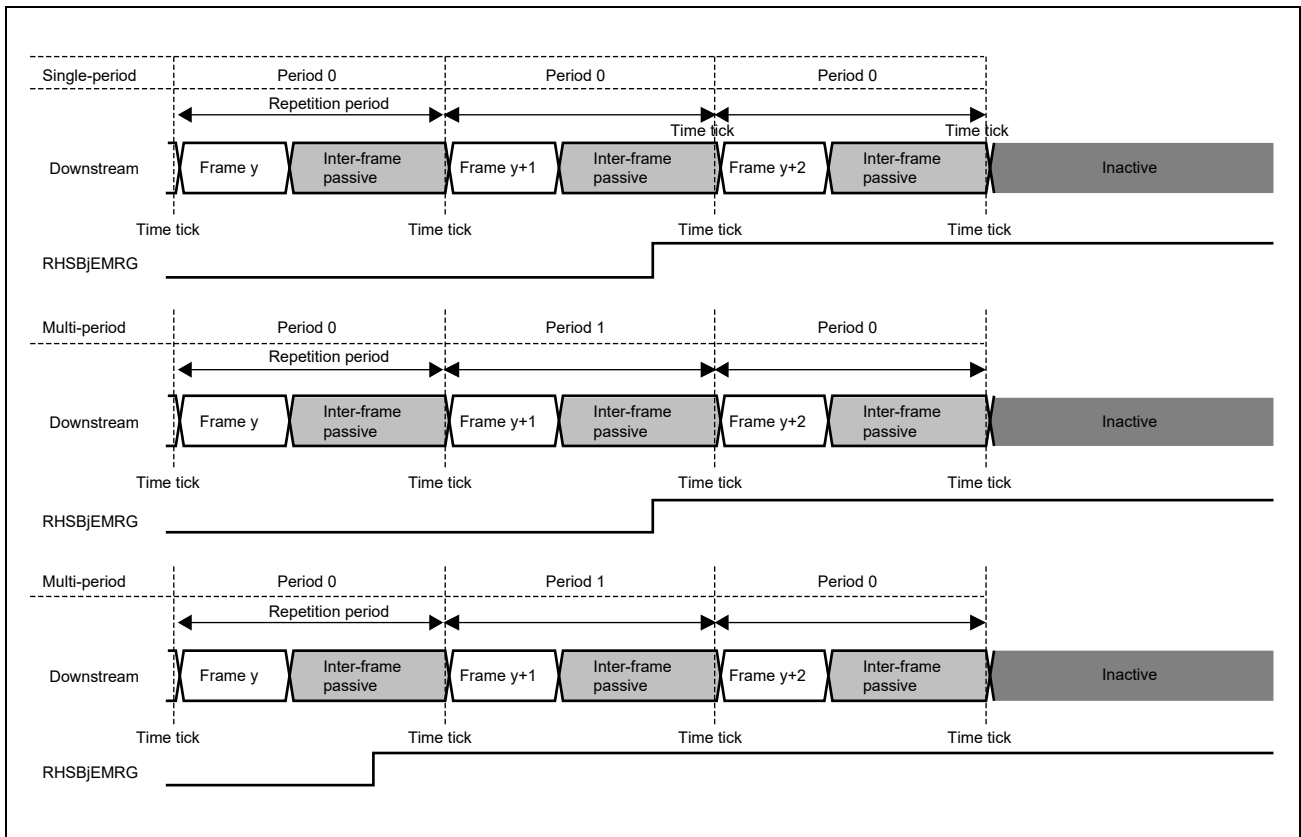


Figure 22.23 Automatic Stop Function in Repetition Mode

22.5.1.9 Downstream Bit Rates

The downstream bit rate (f_{DW}) is derived from the RHSB engine clock source frequency (f_{PE}).

With the divider value x defined by `RHSBjDCR.RHSBjDBR`, the downstream bit rate can be calculated as a function of the RHSB engine clock source frequency:

$$f_{DW} = f_{PE} / x$$

With f_{PE} of 80 MHz, these downstream bit rates can be achieved.

Table 22.33 Possible Downstream Bit Rates with an RHSB Engine Clock Source of 80 MHz

RHSBjDCR. RHSBjDBR	1	3	4	7	8	9	10	11	12
x	2	4	5	8	16	32	64	128	256
Bit rate	40 Mbit/s	20 Mbit/s	16 Mbit/s	10 Mbit/s	5 Mbit/s	2.5 Mbit/s	1.25 Mbit/s	625 kbit/s	312.5 kbit/s

With f_{PE} of 160 MHz these downstream bit rates can be achieved.

Table 22.34 Possible Downstream Bit Rates with an RHSB Engine Clock Source of 160 MHz

RHSBjDCR. RHSBjDBR	1	3	4	7	8	9	10	11	12
x	2	4	5	8	16	32	64	128	256
Bit rate	80 Mbit/s	40 Mbit/s	32 Mbit/s	20 Mbit/s	10 Mbit/s	5 Mbit/s	2.5 Mbit/s	1.25 Mbit/s	625 kbit/s

When the divider value x is odd, the duty cycle of the clock signals on the serial clock line (RHSBjFCLP/RHSBjFCLN) will not be 50%. In this case, the first phase of the clock signal becomes 1 clock cycle longer than that of f_{PE} .

Example: When x is 5 and RHSBjDCR.RHSBjCLP is 0_B , the widths at high and low level of the clock signal are 3 cycles of f_{PE} and 2 cycles of f_{PE} , respectively. When x is 5 and RHSBjDCR.RHSBjCLP is 1_B , the widths at low and high level of the clock signal are 3 cycles of f_{PE} and 2 cycles of f_{PE} , respectively.

22.5.1.10 Data Update and Data Frame Transmission Request

By setting RHSBjIS.RHSBjDTSF to 0_B , the user acknowledges the data transmission start event and informs the RHSB module that new data is available.

RHSBjIS.RHSBjDTSF can be set to 0_B by two strategies:

- Writing the RHSBjIS.RHSBjDTSF flag directly to 0_B
 - Data configured in RHSBjDDRi is used for the next transmission
- Writing to RHSBjDDRi register (by the user or by DMA)
 - New data written to RHSBjDDRi is used for the next transmission

The write DDRi method allows an optimized transmission flow because it automatically sets RHSBjIS.RHSBjDTSF to 0_B and hence starts the transmission.

When a write to an RHSBjDDRi triggers a transmission depends on the configuration parameter RHSBjDCR.RHSBjSLS (number of used DFTEs).

- RHSBjDCR.RHSBjSLS is 00_B (1 DFTE is used): Writing to RHSBjDDR0[31:16] (DFTE0)
- RHSBjDCR.RHSBjSLS is 01_B (2 DFTEs are used): Writing to RHSBjDDR0[15:0] (DFTE1)
- RHSBjDCR.RHSBjSLS is 10_B (3 DFTEs are used): Writing to RHSBjDDR1[31:16] (DFTE2)
- RHSBjDCR.RHSBjSLS is 11_B (4 DFTEs are used): Writing to RHSBjDDR1[15:0] (DFTE3)

Even if RHSBjDCR1.RHSBjSLS1 (number of used DFTEs for period 1) is available in multi-period repetition mode, when a write to an RHSBjDDRi triggers a transmission depends on the configuration parameter RHSBjDCR.RHSBjSLS (number of used DFTEs). If RHSBjDCR1.RHSBjSLS1 is smaller than RHSBjDCR.RHSBjSLS, the user should write dummy data to RHSBjDDRi defined by RHSBjDCR.RHSBjSLS.

- RHSBjDCR.RHSBjSLS is 00_B (1 DFTEs is used): Writing to RHSBjDDR0[31:16] (DFTE0)
- RHSBjDCR.RHSBjSLS is 01_B (2 DFTEs are used): Writing to RHSBjDDR0[15:0] (DFTE1)
- RHSBjDCR.RHSBjSLS is 10_B (3 DFTEs are used): Writing to RHSBjDDR1[31:16] (DFTE2)
- RHSBjDCR.RHSBjSLS is 11_B (4 DFTEs are used): Writing to RHSBjDDR1[15:0] (DFTE3)

The data frame transmission started flag (RHSBjIS.RHSBjDTSF) is set to 1_B when the assembling has been performed at the start of the data frame transmission.

RHSBjIS.RHSBjDTSF equal to 1_B indicates that a data frame transmission has been started and so the downstream data registers can be written with new transmit data.

In addition, the RHSBjIS.RHSBjDTSF is set to 1_B when data transmission is initially enabled by RHSBjDTC.RHSBjDTE to prevent a data frame transmission without explicit software trigger.

Updating RHSBjDDRi while a data frame is transmitted does not affect the ongoing transmission but it requires an additional data transmission.

As there is write protection of the downstream data registers (RHSBjDDRi) while there is new data pending for transmission, it is guaranteed that written downstream data will be transmitted at least one time.

(1) Scheduling of Downstream Data Update

The user can update the downstream data by using the following as a trigger.

- Data Frame Transmission Started
New data can be supplied in a timely manner. In this case, RHSBjIS.RHSBjDTSF is used as a trigger.
- Data Frame Transmission Done
New data is supplied upon completion of data frame transmission. In this case, RHSBjIS.RHSBjDTF is used as a trigger.

Write operations to RHSBjDDRi, RHSBjIS.RHSBjDTSF and RHSBjIS.RHSBjDTF are cleared. It is recommendable to use only one of these write operations as a trigger of downstream data update.

22.5.1.11 Command Frame Transmission Request

The transmission of command frames is always triggered when the user writes to RHSBjDTC.RHSBjCTR. On each request, the transmission of one command frame is started. The method for how command frames are inserted into the downstream communication depends on the selected mode and a set of configuration parameters; for details, see the detailed mode description.

When RHSBjDTC.RHSBjCTR is set to a value not equal to 00_B, a command frame transmission is scheduled. After completion of the command frame transmission, RHSBjDTC.RHSBjCTR is automatically set to 00_B and the user is able to set a new request.

22.5.2 Upstream Communication

The RHSB module uses asynchronous serial reception for upstream communication.

There are two upstream channels available to connect up to two slaves with the RHSB module. Only one channel can be selected as the active channel at one time. Any activity on the channel that is not selected is ignored.

It is possible to receive consecutive upstream frames sent on the same serial input line.

Reception of upstream communication is only supported when the user has enabled this function by setting RHSBjUCR.RHSBjUE to 1_B in CONFIG state.

The diagram in **Figure 22.24** illustrates the functional blocks of the upstream engine. The detailed upstream communication is explained in this section.

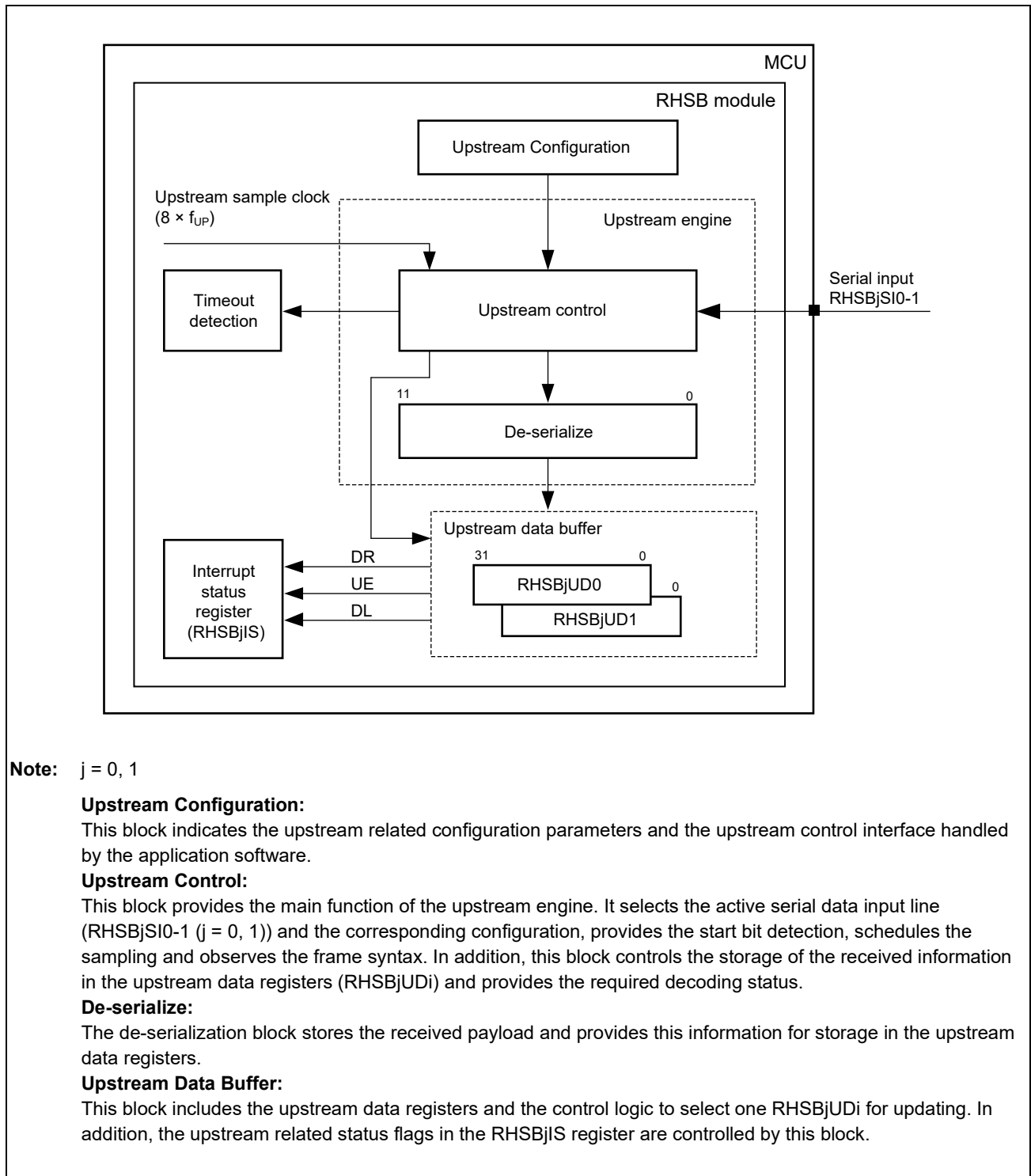


Figure 22.24 Block Diagram of Upstream

22.5.2.1 Upstream Modes

The RHSB module has two upstream modes with different methods for how the slaves are connected to the serial input lines. These upstream modes can be configured by RHSBjUCR.RHSBjUMS:

- Dedicated input mode
 - Each slave is connected to an individual serial input line.
 - The active serial input line is selected by RHSBjUCS.RHSBjACC.
- Shared input mode
 - All slaves are connected on the serial input line 0.
 - The active serial input line is always line 0 independently of the value in RHSBjUCS.RHSBjACC.

Note that when using the Shared input mode, the unused serial input lines are not required to be configured as RHSB inputs in the MCU port function registers.

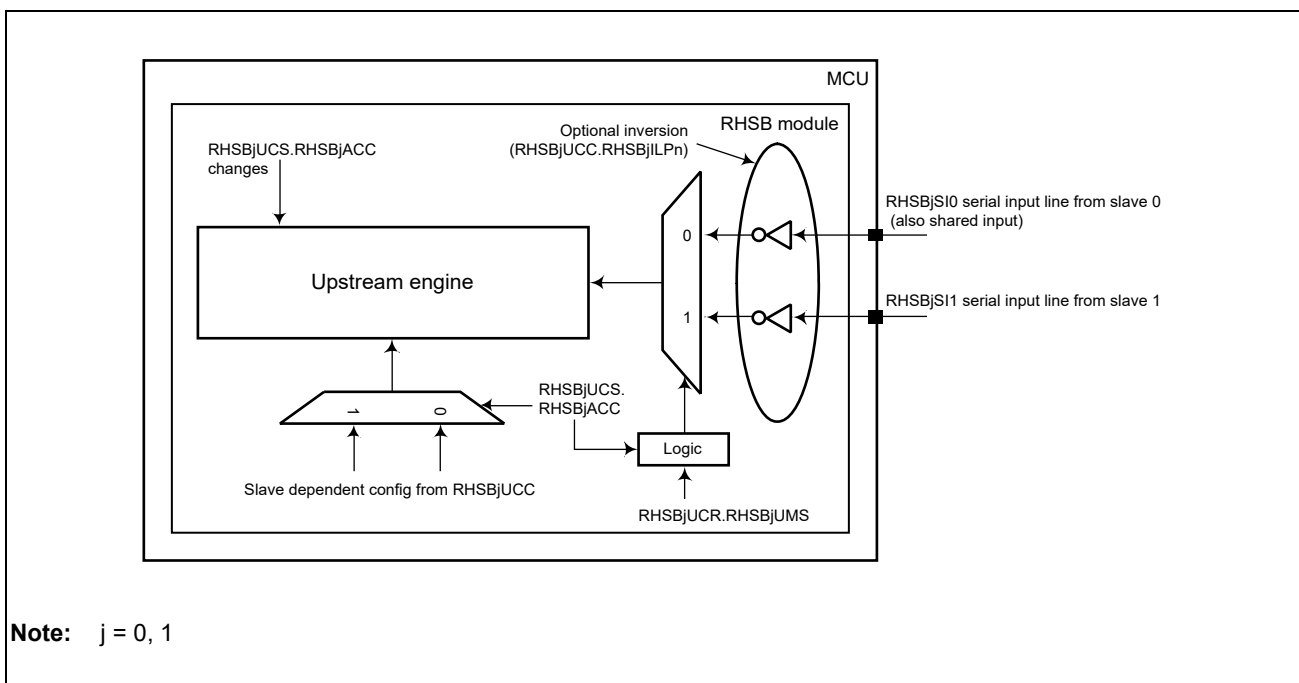


Figure 22.25 Slave Connection and Selection for Upstream Communication

Figure 22.25 illustrates how the serial input lines are connected to the upstream engine. The shown 'Logic' ensures that in shared mode, input line 0 is always used, independent from the value of RHSBjUCS.RHSBjACC. The shown 'RHSBjUCS.RHSBjACC changes' information is used by the upstream engine to abort an ongoing reception when the selected channel changes.

22.5.2.2 Individual Slave Configuration

The RHSB module supports for each slave an individual upstream configuration. The usage of different channel configurations is independent from the upstream mode (RHSBjUCR.RHSBjUMS). For each channel, the user can program via the RHSBjUCC register:

- The frame type (8 or 12 bit format)
- The number of stop bits (2 or 3)
- The parity type (even or odd)
- The upstream bit rate (see **Section 22.5.2.5, Upstream Bit Rates** for details)

The serial input line polarity is independent from RHSBjUCS.RHSBjACC. The inversion function configured by RHSBjUCC.RHSBjILPn is fixed and assigned to a serial input line n. Thus, RHSBjUCC.RHSBjILP0 is used for all two channels in shared input mode.

The configuration used by the upstream engine to decode upstream frames is defined by RHSBjUCS.RHSBjACC.

There are two ways to change the active upstream configuration:

- Direct request for configuration change by writing to RHSBjUCS.RHSBjACC.
- Request of a command transmission with upstream data request by writing 11_B to RHSBjDTC.RHSBjCTR.

22.5.2.3 Frame Format Types

The RHSB module supports two frame types for upstream communication that only differ in the number of data bits (8 or 12 bit). Other configuration options are not influenced by the frame type.

When the parity type is even, the number of 1_B bits inside the parity covered area is even (0, 2, 4, 6, 8, 10 and 12).

When the parity type is odd, the number of 1_B bits inside the parity covered area is odd (1, 3, 5, 7, 9, 11 and 13).

Figure 22.26 and **Figure 22.27** show an upstream frame with non-inverted data line.

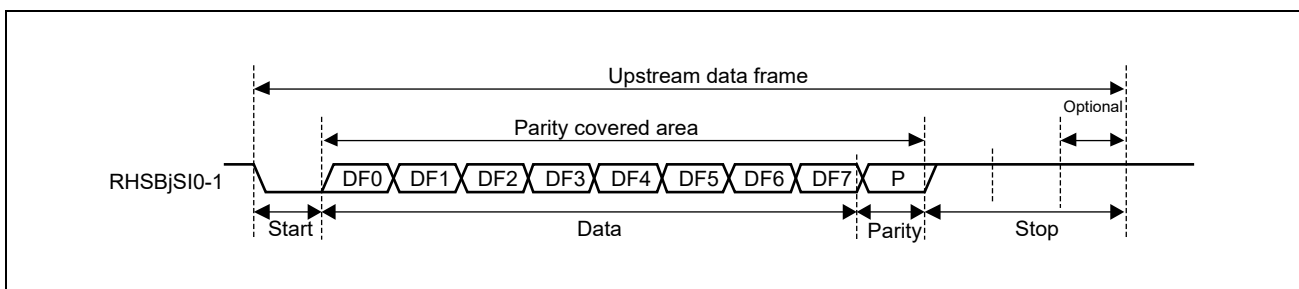


Figure 22.26 Upstream Frame Format with 8 Data Bits (12- or 13-Bit Frame)

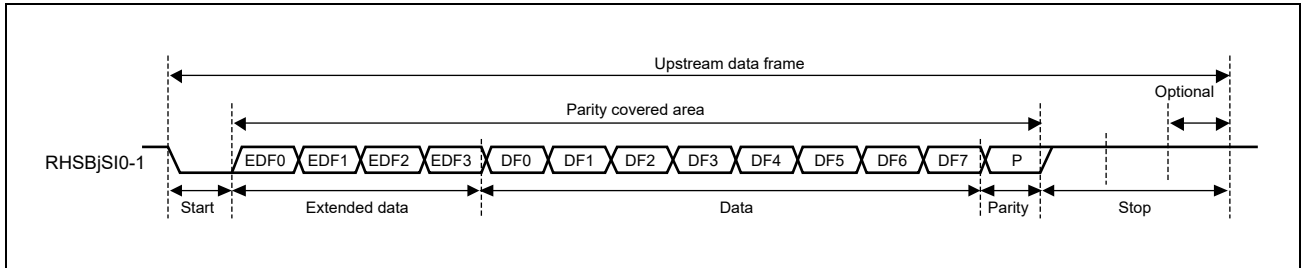


Figure 22.27 Upstream Frame Format with 12 Data Bits (16- or 17-Bit Frame)

22.5.2.4 Frame Storing

The RHSB module offers two upstream data registers (RHSBjUDI) where the upstream engine stores its decoding results. There are two methods selectable in RHSBjUCR.RHSBjFSM defining how the RHSBjUDI is selected for updating.

Note that the decoding of an ongoing reception is aborted when RHSBjUCS.RHSBjACC changes. In this case, no decoding result is stored.

(1) Normal Storage Method

This method is used when RHSBjUCR.RHSBjFSM is 0_B.

The storage strategy is identical for frames of 8-bit and 12-bit format.

The active channel configuration (RHSBjUCS.RHSBjACC) defines the upstream data register to store the decoding results. The target can be identified by these relations:

- When RHSBjUCS.RHSBjACC is 00_B, RHSBjUD0 is updated
- When RHSBjUCS.RHSBjACC is 01_B, RHSBjUD1 is updated

(2) Addressed Storage Method

This method is used when RHSBjUCR.RHSBjFSM is 1_B.

When the frame type is configured as 8-bit format, the target is always RHSBjUD0.

When the frame type is configured as 12-bit format, the target depends on the received extended data field (EDF, see **Figure 22.25**). The target can be identified by these relations:

- When RHSBjEDF[3:2] is 00_B, RHSBjUD0 is updated
- When RHSBjEDF[3:2] is 01_B, RHSBjUD1 is updated

Note also in case of a faulty reception (parity or stop bit error detected), the received extended data field is used to identify the target for updating the error flags.

22.5.2.5 Upstream Bit Rates

The upstream bit rate (f_{UP}) is derived from the downstream bit rate (f_{DW}).

The bit rate is configured for each channel individually by $RHSBjUCC.RHSBjUBRn$.

With ($x = RHSBjUCC.RHSBjUBRn$), the upstream bit rate used by channel n can be calculated as a function of the downstream bit rate such that:

$$f_{UP} = f_{DW} / 2^x$$

Because the RHSB module samples each upstream bit 8 times, the maximal upstream bit rate is 1/8 of the RHSB engine clock source frequency ($f_{PE}/8$). However, set this RHSB module to 16-frequency division at 5 MHz or lower.

22.5.2.6 Update of Decoding Status

The upstream engine updates one of the $RHSBjUDI$ registers after the decoding of an upstream frame has been completed (after the last stop bit).

When the decoder has detected a parity or stop bit error, the related error flags ($RHSBjFERR$ or $RHSBjPERR$ or both) are set in the corresponding $RHSBjUDI$ register.

When the decoder has not detected parity nor stop bit errors, this frame is judged as a valid frame and these parts of the corresponding $RHSBjUDI$ are updated:

- New data flag ($RHSBjND$) is set to 1_B
- Data field ($RHSBjDF$) is updated
- Extended data field ($RHSBjEDF$) is updated in case of 12-bit frames
- Data lost flag ($RHSBjDL$) is set to 1_B if the new data flag was already 1_B

22.5.3 Timeout Detection

The concept of timeout detection realized by the RHSB module is linked to the concept of “command transmission with data request”. The concept assumes this flow:

Step 1: Master requests data from slave x by sending a command frame.

Step 2: Slave x answers the master by sending the requested data on the upstream channel.

From an application point of view the master expects to receive an answer from slave x within a certain period (timeout period). This expectation is processed by the timeout detection function.

To use the timeout detection function the user should enable the upstream reception (RHSBjUCR.RHSBjUE is 1_B) and should enable the timeout function (RHSBjUCR.RHSBjTOE is 1_B).

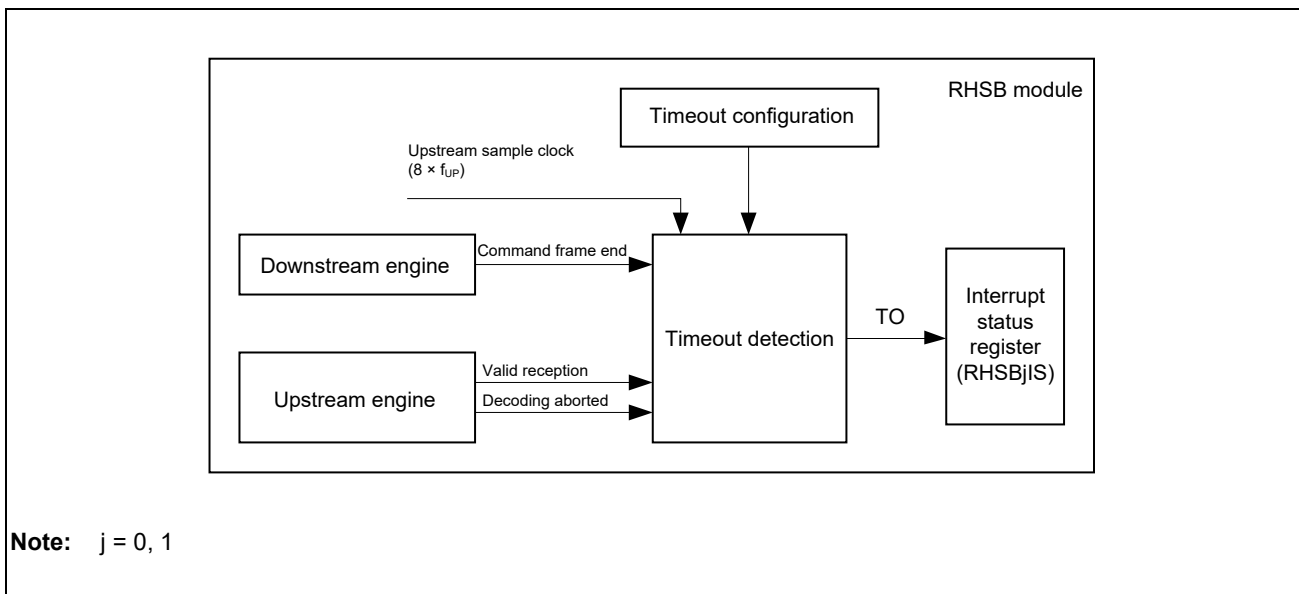


Figure 22.28 Block Diagram of Timeout

The diagram in **Figure 22.28** illustrates the scope of timeout detection block. It is controlled by the downstream and upstream engine. The timeout detection controls RHSBjIS.RHSBjTOF. The detection of a timeout has no influence on an ongoing reception.

Note that the timeout detection cannot distinguish between the reception due to the request and other receptions. For example, if the shared input mode configuration is used and two slaves are transmitting upstream frames during the timeout period. The user must ensure that in fault free operation, only one upstream reception can happen during the timeout period.

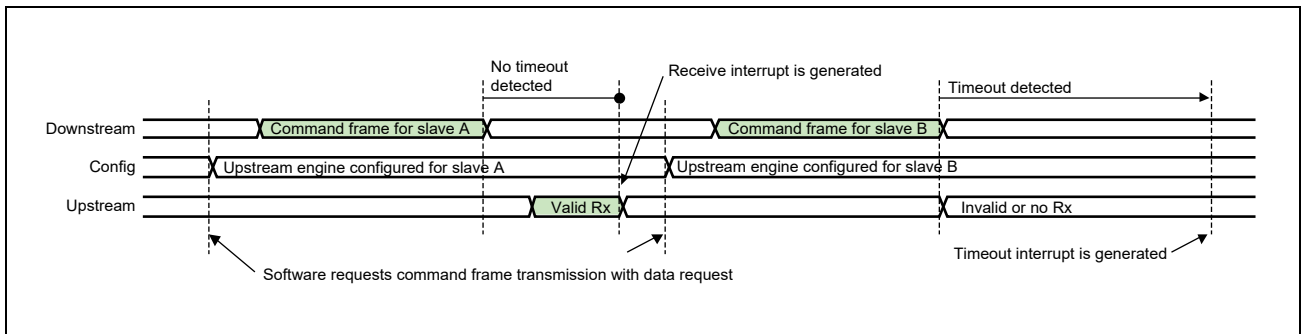


Figure 22.29 Example of Timeout Detection

The example shown in **Figure 22.29** illustrates the typical timing of the timeout detection.

In the first case, the user issues a command frame transmission with data request for slave A. The downstream configuration is updated, and then a command frame is scheduled for transmission. After the end of transmission, the timeout period starts. The valid response of the slave is decoded before the timeout period ends.

In the second case, the user issues a command frame transmission with data request for slave B. In this case, the timeout period ends before a valid response was decoded.

22.5.3.1 Command Transmission with Remote Data Request

To perform a command transmission with data request, the user must follow this flow:

- Writing the command data to the RHSBjDCD register
- Define the target slave by writing to RHSBjDTC.RHSBjSSCF
- Request command transmission with data request by writing 11_B to RHSBjDTC.RHSBjCTR

Writing of the command to RHSBjDRC.RHSBjCTR generates triggers from the RHSB module to carry out these actions:

- Update of the active slave configuration (RHSBjUCS.RHSBjACC) with the target slave number (RHSBjDTC.RHSBjSSCF)
- Abort an ongoing reception if the active slave changes
- Transmit the command frame on the downstream channel
- Start timeout counter when the command transmission has been completed

When there is a valid reception on the upstream channel, the RHSB module judges the remote data request as handled and the timeout counter is stopped.

When there is no valid reception on the upstream channel before the timeout counter elapses, the remote data request is judged as failed and the timeout is flagged by setting RHSBjUDi.RHSBjTO and RHSBjIS.RHSBjTOF to 1_B.

22.5.3.2 Timeout Detection Details

The timeout counter uses the nominal upstream bit time for timeout detection. The timeout counter is not influenced by possible resynchronization effects during start bit decoding.

The timeout counter is initialized to the value $(\text{RHSBjUCR.RHSBjRTO} + 1) \times 8$ (U bit), when the command frame transmission has been completed (end of frame passive phase).

The timeout counter is decremented with every sample period (U bit/8) used by the decoder according to the active slave configuration (RHSBjUCC.RHSBjUBRn).

The timeout counter stops when:

- There is a valid reception decoded on the upstream channel or
- The timeout counter elapse or
- The decoding is aborted (e.g. by changing the RHSBjUCS.RHSBjACC)

Note that there is no synchronization between the downstream frame passive phase end and the upstream sample points. Thus, the first timeout counter decrement can happen between 1_D bit and 1 U bit/8.

22.5.4 Test Mode Operation

The RHSB module is able to support key-on tests when RHSBjGC.RHSBjOPS is TEST.

The aim of the test mode is to transmit data on the downstream channel and receive the same data as an upstream frame. Because the upstream engine can only decode UART-like frames, such pattern needs to be emulated within the SPI-like downstream frame.

Since the downstream and upstream paths in the RHSB module are independent, the test mode allows good test coverage of all RHSB flags.

While the test mode is active, the serial input lines are disconnected from the upstream engine and the outputs are driven with fixed levels:

- The serial output line (RHSBjSOP/RHSBjSON) is driven with a high level
- The serial clock line (RHSBjFCLP/RHSBjFCLN) is driven with a level depending on the clock line phase (low level when RHSBjDCR.RHSBjCLP is 0_B)
- The chip select lines (RHSBjCSD0-1) are driven with a level depending on the chip select line polarity (low level when RHSBjSDCi.RHSBjCSLPn is 0_B)

Figure 22.30 illustrates the details about the position of the loop-back multiplexer. Only the inversion of the downstream data (RHSBjSDCi.RHSBjSOLPn) is influencing the test mode; all other inversions are out of scope.

The loop-back multiplexer connects the internal serial data (RHSBjSOP/RHSBjSON) signal to one upstream channel. The channel is selected by an active internal chip select. Thus, all two upstream channels can be checked.

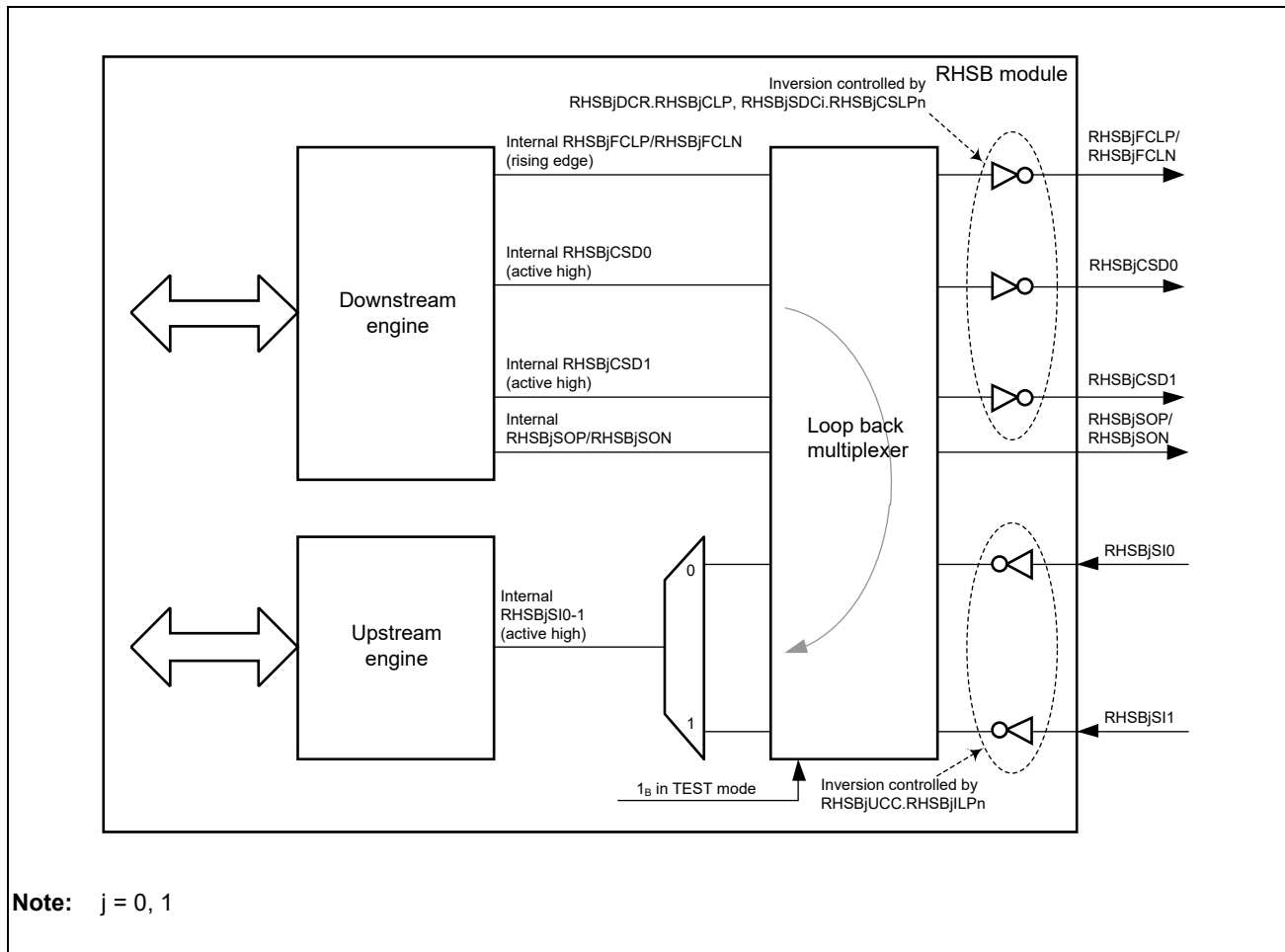


Figure 22.30 Loop-Back Connection in Test Mode

The test mode should be carried out under the following conditions:

- The downstream bit rate divider should be $f_{DW} = f_{PE}/8$ (this results in the maximum bit rate of 10 MBit/s when using 80 MHz RHSB engine clock source frequency). The user should set RHSBjDCR.RHSBjDBR to 7_D .
- The upstream bit rate should be equal to the downstream bit rate ($f_{UP} = f_{DW}$). The user should set RHSBjUCC.RHSBjUBRn to 0_D for all two upstream channels.

22.5.4.1 Test Mode Data Generation

Figure 22.31 shows an example of how the bits of a command frame are received by the upstream channel. The example uses non inverted downstream data without assertion/deassertion phases and assumes a 12-bit format with two stop bits.

On the downstream data line, a command frame is transmitted for slave 1, and the receiver has to be connected to the same slave by setting RHSBjUCS.RHSBjACC to 01_B.

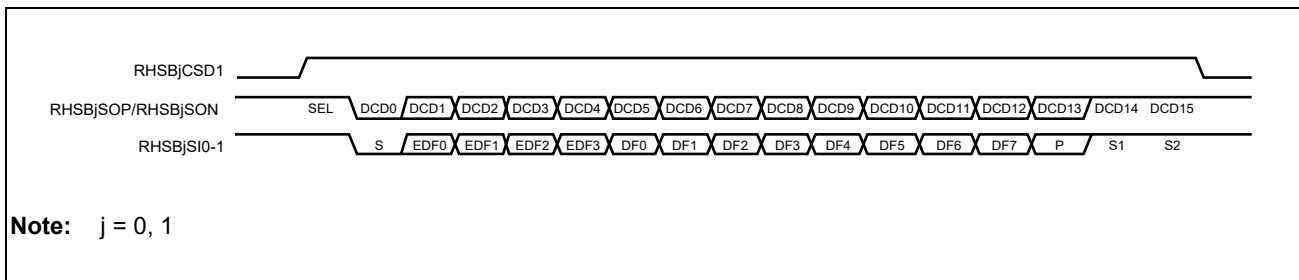


Figure 22.31 Example of How a Command Frame is Received by the Upstream Engine

By writing a pattern into the RHSBjDCD register, the user is able to check the parity logic and the stop bit error detection. In addition, it is possible to check all 32 bits of the RHSBjDCD register, the command length configuration, the assertion/deassertion phase lengths and the data inversion.

Similar checks are also possible with data frames. Note that only the data for the select slave is received, even if multiple slaves are addressed in the data frame.

22.5.5 Interrupts

The RHSB module has 9 sources that can be used for interrupt request generation.

The interrupt status flags are cleared by the user when writing the flags to 0_B. Writing 1_B has no effect. For some interrupt flags, there are additional functional clear conditions defined to allow an optimized software flow. These interrupt flags are marked by * in the interrupt source lists in **Section 22.5.5.1, Downstream Related Interrupts (Transmission)** and **Section 22.5.5.2, Upstream Related Interrupts (Reception)**.

When the RHSB module is in the RESET state or CONFIG state, an interrupt request is not generated.

22.5.5.1 Downstream Related Interrupts (Transmission)

- Interrupt line 0
 - Data frame transmission started (RHSBjDTSF) *
 - Data frame transmission done (RHSBjDTF) *
- Interrupt line 1
 - Command frame transmission done (RHSBjCTF) *
 - Transmission started (RHSBjTSF)
- Interrupt line 2
 - Emergency frame transmission done (RHSBjETF)

22.5.5.2 Upstream Related Interrupts (Reception)

- Interrupt line 3
 - Data received (RHSBjDRF) *
- Interrupt line 4
 - Upstream error (RHSBjUEF) *
 - Timeout detected (RHSBjTOF)
 - Data lost (RHSBjDLF) *

By reading the RHSBjUSS register, the user can get informed about the complete upstream status.

22.5.5.3 Interrupt Request Details

There is an individual interrupt status flag in the RHSBjIS register (RHSBjIS.RHSBjxF) and an individual interrupt enable bit in the RHSBjIC register (RHSBjIC.RHSBjxIE) for each interrupt source (x).

The function of the interrupt status is independent from the interrupt enable. The interrupt enable controls only if an interrupt request is made to the MCU's interrupt controller.

Figure 22.32 illustrates how an interrupt request is generated and the realization of the interrupt grouping.

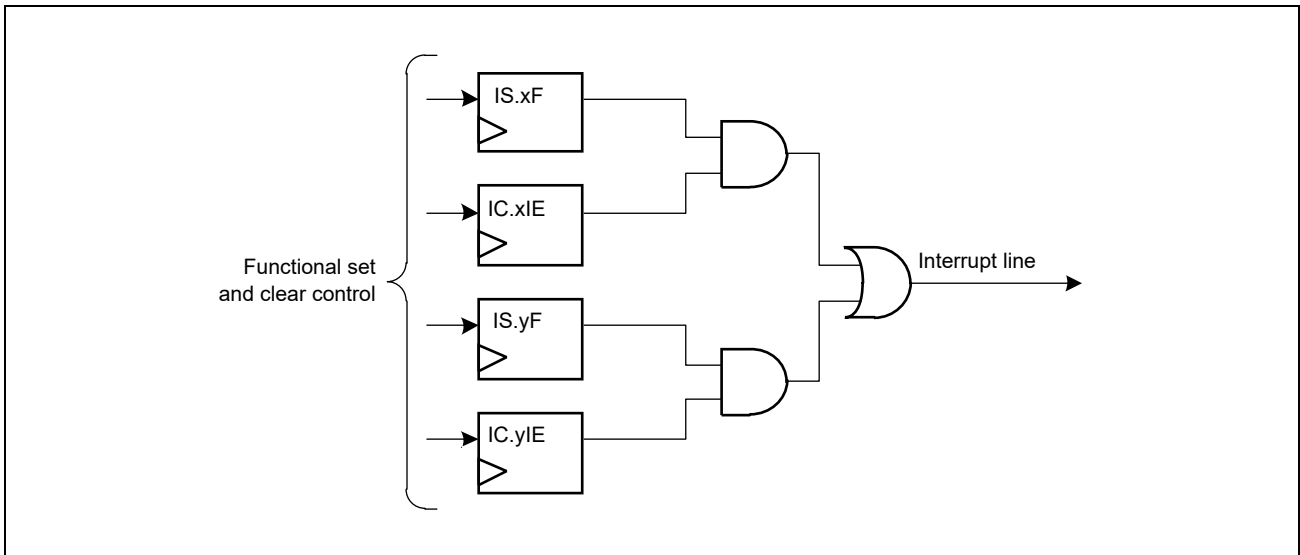


Figure 22.32 Interrupt Grouping (Example of Two Interrupts x and y)

22.5.6 DMA Capability

22.5.6.1 DMA Usage for Downstream Data Transmission

The downstream data DMA function is intended to serve applications where pre-calculated data tables are stored in the memory (RAM or FLASH) defining the pattern the RHSB module has to transfer to slaves.

A downstream data DMA transfer is started when RHSBjIS.RHSBjDTSF is 1_B and DMA is enabled by RHSBjGC.RHSBjDDE.

The user should not use RHSBjIS.RHSBjDTSF as interrupt source (RHSBjIC.RHSBjDTSIE should be set to 0_B) when using downstream data DMA. In addition, the user should not write to the downstream data registers (RHSBjDDRi) nor set RHSBjIS.RHSBjDTSF to 0_B.

In **Figure 22.33**, case 1) shows normal DMA transfer of downstream data for transmission. In case 2), the DMAC is not able to supply downstream data within the repetition cycle, the DMAC retransmits the last data that it wrote. Accordingly, define a repetition cycle that is longer than the time the DMAC will take to supply the downstream data.

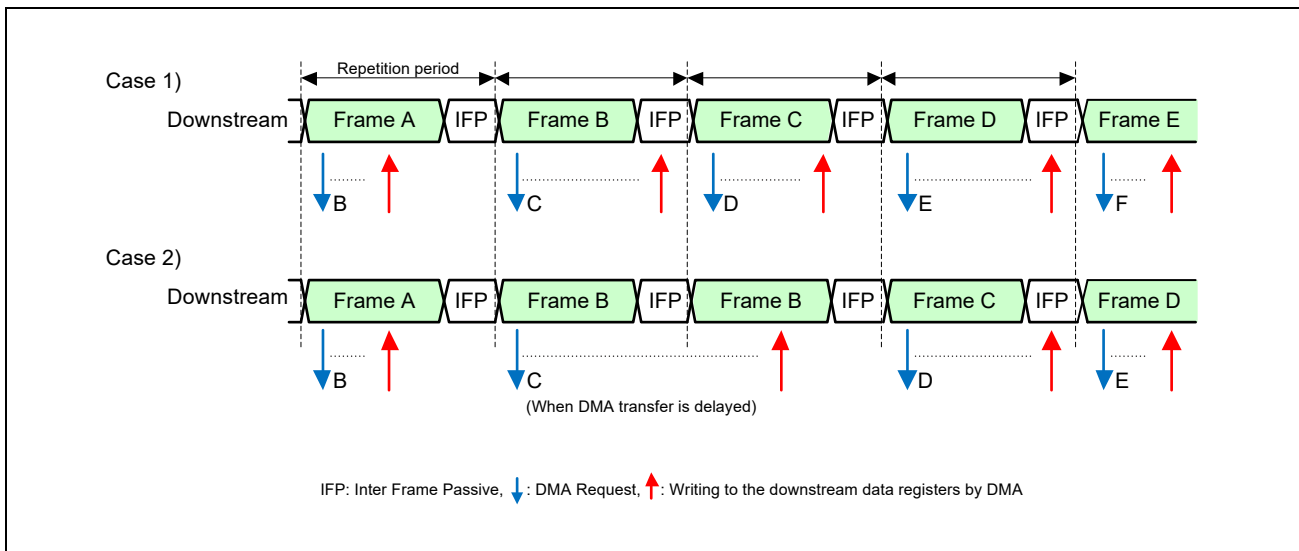


Figure 22.33 Downstream Data DMA Timing

(1) Configuring and Enabling Flow to Use Downstream Data DMA

These configurations should be done during CONFIG state:

- Set RHSBjDCR.RHSBjDMS to 01_B (triggered mode)
- Set RHSBjIC.RHSBjDTSIE to 0_B
- Set all other configuration parameters according to the application requirements
- Configure the DMA controller to serve the required data to the RHSB module
 - Target in the RHSB module is the RHSBjDDR0 register
 - Number of transferred data words depends on RHSBjDCR.RHSBjSLS

These actions can be done in ACTIVE state according to the application requirements:

- Write initial data to RHSBjDDRi registers
 - The first DMA transfer after setting RHSBjDTC.RHSBjDTE to 1_B is requested after a data frame with this initial data has been transmitted
- Enable or disable the downstream data transmission by setting RHSBjDTC.RHSBjDTE or RHSBjDTC.RHSBjTSR to 1_B.
- Enable or disable the downstream data DMA function (RHSBjGC.RHSBjDDE)

While the downstream data DMA is enabled, RHSBjIS.RHSBjDTSF equal to 1_B indicates a pending DMA transfer.

(2) Data Layout in Memory for Downstream Data DMA

The layout of the data to be transferred from the memory to the RHSBjDDRi registers of the RHSB module depends on the data frame configuration in RHSBjDCR.RHSBjSLS.

Because the DMA request (RHSBjIS.RHSBjDTSF) is cleared by writing to the downstream data registers (RHSBjDDRi), the user must ensure that the transferred amount of data matches the required data for transmission (see **Section 22.5.1.10, Data Update and Data Frame Transmission Request** for details about the request mapping). The RHSB module allows writing more than the required data to the RHSBjDDRi registers. Therefore, there is no functional restriction when the DMA controller supports only 32 bit transfers.

Note that when using 16-bit transfers, the user must ensure that the upper part (bits [31:16]) of the downstream data registers are written first.

22.5.6.2 DMA Usage for Downstream Command Transmission

The downstream command DMA function is intended to serve applications where command data tables are stored in the memory (RAM or FLASH) the RHSB module has to transfer to slaves.

A downstream command DMA transfer is started when RHSBjIS.RHSBjCTF is 1_B and DMA is enabled by RHSBjGC.RHSBjDCDE.

The user should not use RHSBjIS.RHSBjCTF as interrupt source (RHSBjIC.RHSBjCTIE should be set to 0_B) when using downstream command DMA. In addition, the user should not write to the downstream command data register (RHSBjDCD) nor downstream transmission control register (RHSBjDTC), nor set RHSBjIS.RHSBjCTF to 0_B.

(1) Configuring and Enabling Flow to Use Downstream Command DMA

These configurations should be done during CONFIG state:

- Set RHSBjIC.RHSBjCTIE to 0_B
- Set all other configuration parameters according to the application requirements
- Configure the DMA controller to serve the required data to the RHSB module
 - Targets in the RHSB module are the RHSBjDCD register and RHSBjDTC register
 - Number of transferred command bits depends on RHSBjDTC.RHSBjNCB

These actions can be done in ACTIVE state according to the application requirements:

- Write command data to RHSBjDCD register and configuration parameters to RHSBjDTC register
- Enable or disable the downstream command DMA function (RHSBjGC.RHSBjDCDE)

While the downstream command DMA is enabled, RHSBjIS.RHSBjCTF equal to 1_B indicates a pending DMA transfer.

(2) Command Data Layout in Memory for Downstream Command DMA

The layout of the command data to be transferred from the memory to the RHSBjDCD register of the RHSB module depends on the command frame configuration in RHSBjDTC.RHSBjNCB.

Because the DMA request (RHSBjIS.RHSBjCTF) is cleared by writing to the command transmission request (RHSBjDTC.RHSBjCTR), the user must ensure that the transferred amount of command data fits to the required command for transmission. The RHSB module allows writing more than the required data to the RHSBjDCD register. Therefore, there is no functional restriction when the DMA controller supports only 32 bit transfers.

Note that when using 16 bit transfers the user must ensure that the upper part (bits [31:16]) of the downstream command data register is written first.

22.5.6.3 DMA Usage for Upstream Data Reception

The upstream DMA function is intended to store a set of upstream frames into the memory.

An upstream DMA transfer is started when RHSBjIS.RHSBjDRF is 1_B and DMA is enabled by RHSBjGC.RHSBjUDE.

The user should not use RHSBjIS.RHSBjDRF as interrupt source (RHSBjIC.RHSBjDRIE should be set to 0_B) when using upstream DMA. In addition, the user should not read from the upstream data read registers (RHSBjUDR) nor set RHSBjUDi.RHSBjND to 0_B nor set RHSBjIS.RHSBjDRF to 0_B.

When the DMA controller is not able to process the new received data in a timely manner, a received message might be lost because it is no longer available in the RHSBjUDR (even if still available in the related RHSBjUDi). The software gets informed about the overrun by the RHSBjUDR.RHSBjNDS bits and about data lost by RHSBjUDi.RHSBjDL.

(1) Configuring and Enabling Flow to Use Upstream DMA

These configurations should be done during CONFIG state:

- Set RHSBjUCR.RHSBjUE to 1_B
- Set RHSBjIC.RHSBjDRIE to 0_B
- Set all other configuration parameters according to the application requirements
- Configure the DMA controller to serve the required data to the RHSB module
 - Source in the RHSB module is RHSBjUDR
 - Number of transferred data (8, 16, 32 bits) depends on the application requirements

These actions can be done in ACTIVE state according to the application requirements:

- Request data from the slaves (e.g. by command frame)
- Enable or disable the upstream DMA function (RHSBjGC.RHSBjUDE)
- Evaluate the upstream error interrupts to handle upstream problems

While the upstream DMA is enabled, RHSBjIS.RHSBjDRF equal to 1_B indicates a pending DMA transfer.

(2) Data Layout in Memory for Upstream DMA

The DMA transfer should only use the RHSBjUDR as source for the DMA transfer. Whether all 32 bits of the RHSBjUDR are transferred to the memory or only a part (8 or 16 bits) depends on the application requirements.

When the DMA controller reads from the RHSBjUDR, the related new data flag in an RHSBjUDi register is set to 0_B. In addition, RHSBjIS.RHSBjDRF is set to 0_B. Note that the error flags in the RHSBjUDi registers are not affected by DMA transfer.

22.6 Cross Bar (XBAR)

22.6.1 Overview

XBAR is configured of the following functions.

- RHSB XBAR

The RHSB XBAR selects signals from the timer and GPIO with the multiplexer and outputs arbitrary signals to the RHSB. The number of units of RHSB is two. Each RHSB has one XBAR. Each XBAR has four 16-bit sub XBARs. The output of signals to the RHSB is specified by setting the RHSBGjCRkH, RHSBGjCRkL and RHSBGjCROk registers.

22.6.1.1 Number of Units

This microcontroller has the following number of units of the Renesas high speed bus Cross bar (RHSB XBAR).

Table 22.35 Number of Units

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Units	2	2
Unit Name	RHSB XBARj (j = 0, 1)	RHSB XBARj (j = 0, 1)

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Units	2	2
Unit Name	RHSB XBARj (j = 0, 1)	RHSB XBARj (j = 0, 1)

22.6.1.2 Register Base Addresses

The RHSB XBAR base addresses are listed in the following table.

The RHSB XBAR register addresses are given as offsets from the base address in general.

Table 22.36 Register Base Addresses

RHSB XBARj	RHSBj_Base	Bus Group
<RHSB_XBAR0_base>	FFED D000 _H	Peripheral Group 2L
<RHSB_XBAR1_base>	FFED D800 _H	Peripheral Group 2L

22.6.1.3 Clock Supply

The RHSB XBAR clock supply is shown in the following table.

Table 22.37 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
RHSB XBARj	PCLK	CLK_LSB

Note: j = 0, 1

22.6.1.4 Reset Source

The RHSB XBAR reset sources are listed in the following table. RHSB XBAR is initialized by these reset sources.

Table 22.38 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
RHSB XBAR0	All registers	✓	✓	✓	✓	—	—
RHSB XBAR1	All registers	✓	✓	✓	✓	—	—

22.6.2 Module Configuration

22.6.2.1 RHSB XBAR Configuration

When ATU-V is used, the RHSB cross bar is selection logic that connects ATU- V/GPIO and RHSB. The RHSB cross bar has four 16-bit sub XBARs for each RHSB channel.

See **Table 30.13 Block Configuration** about usable channels.

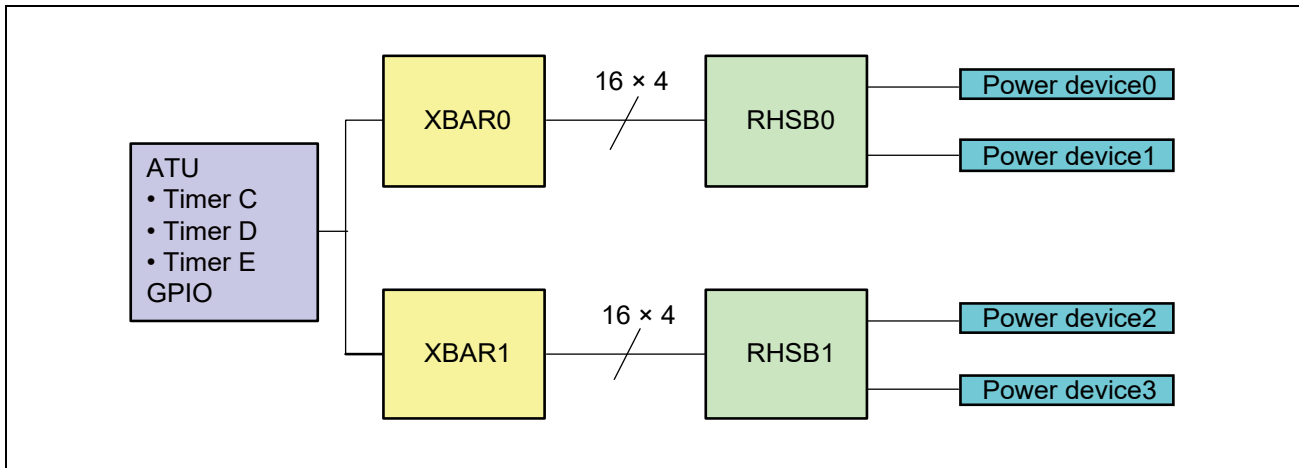


Figure 22.34 RHSB XBAR Configuration when ATU-V is Used

When GTM is used, the RHSB cross bar is selection logic that connects GTM/GPIO and RHSB. The RHSB cross bar has four 16-bit sub XBARs for each RHSB channel.

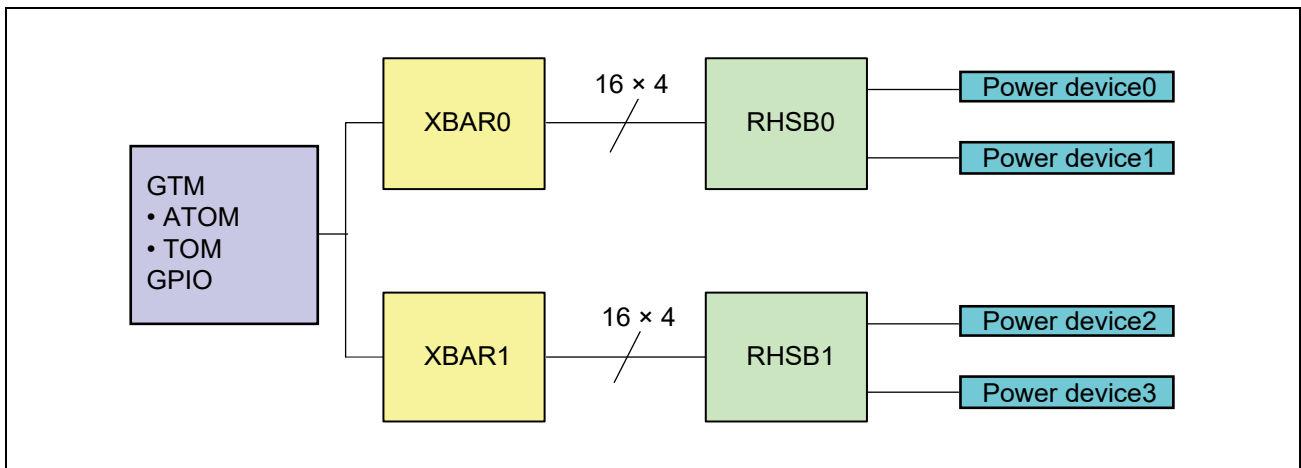


Figure 22.35 RHSB XBAR Configuration when GTM is Used

The RHSB cross bar has four 16-bit sub XBARs in the 1st stage. The timer signals can be selected by setting the RHSBGjCRkH and RHSBGjCRkL registers in the 16-bit sub XBAR. The GPIO signals or the timer signals, which are selected in the 1st stage, are selected by setting the RHSBGjCROk registers in the 2nd stage.

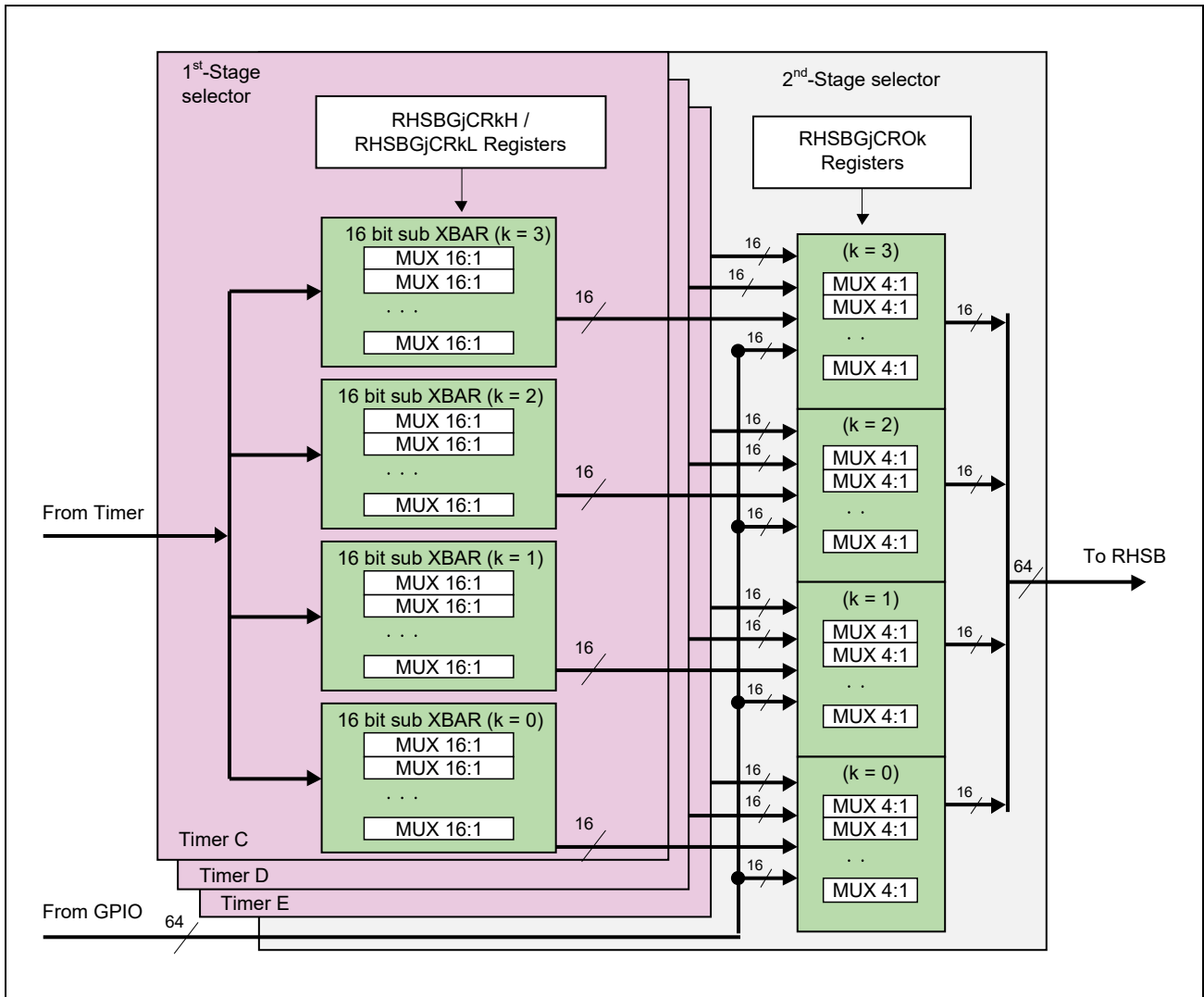


Figure 22.36 RHSB XBAR Block Diagram when ATU-V is Used

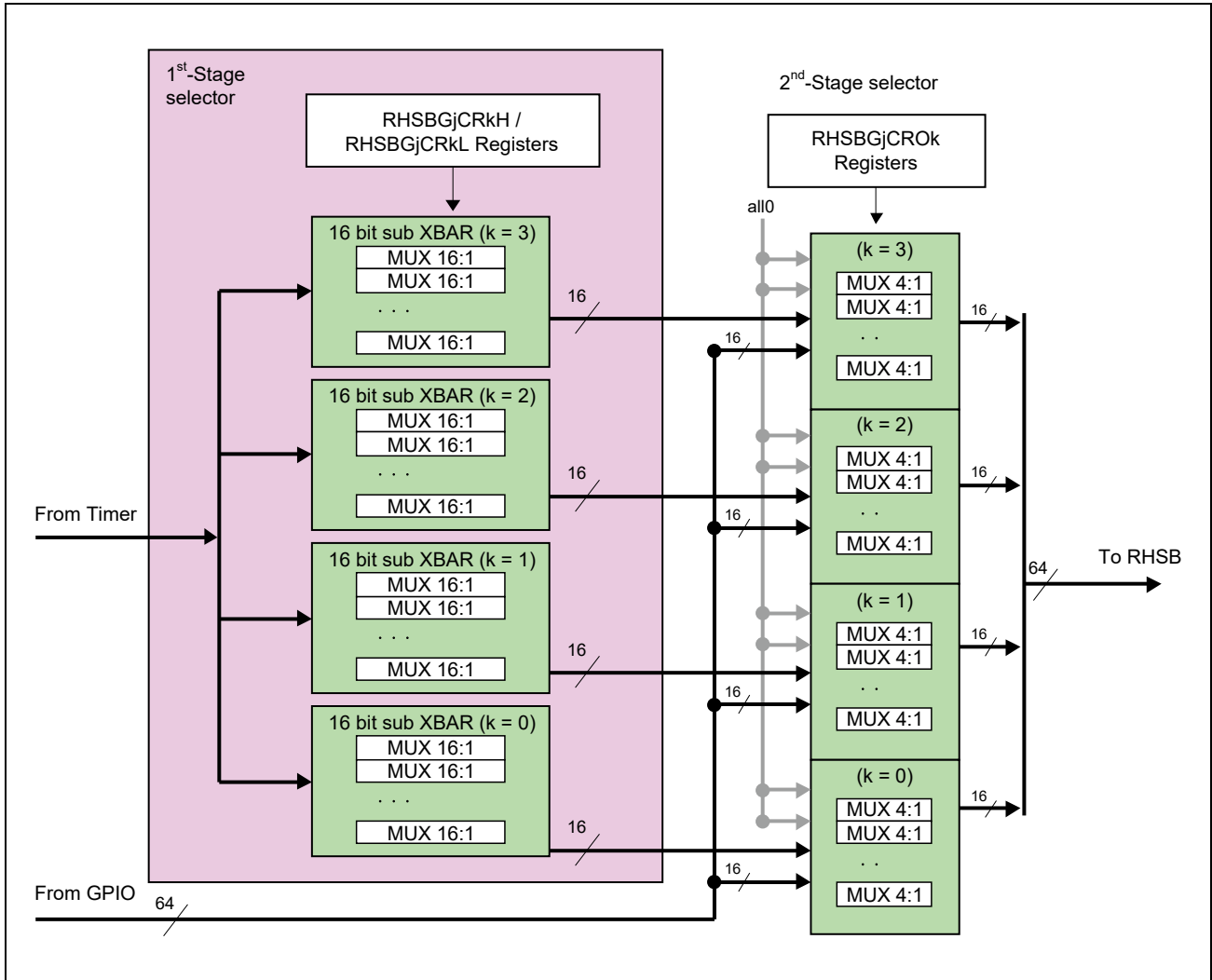


Figure 22.37 RHSB XBAR Block diagram when GTM is Used

22.6.3 Registers

22.6.3.1 List of Registers

The following tables show the address allocation of the RHSB XBAR registers.

Table 22.39 RHSB XBAR Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection
RHSB XBARj	Microsecond Bus Control Register H	RHSBGjCR0H	<RHSB_XBARj_base> + 00 _H	8, 16, 32	—
RHSB XBARj	Microsecond Bus Control Register L	RHSBGjCR0L	<RHSB_XBARj_base> + 04 _H	8, 16, 32	—
RHSB XBARj	Microsecond Bus Control Register H	RHSBGjCR1H	<RHSB_XBARj_base> + 08 _H	8, 16, 32	—
RHSB XBARj	Microsecond Bus Control Register L	RHSBGjCR1L	<RHSB_XBARj_base> + 0C _H	8, 16, 32	—
RHSB XBARj	Microsecond Bus Control Register H	RHSBGjCR2H	<RHSB_XBARj_base> + 10 _H	8, 16, 32	—
RHSB XBARj	Microsecond Bus Control Register L	RHSBGjCR2L	<RHSB_XBARj_base> + 14 _H	8, 16, 32	—
RHSB XBARj	Microsecond Bus Control Register H	RHSBGjCR3H	<RHSB_XBARj_base> + 18 _H	8, 16, 32	—
RHSB XBARj	Microsecond Bus Control Register L	RHSBGjCR3L	<RHSB_XBARj_base> + 1C _H	8, 16, 32	—
RHSB XBARj	Microsecond Bus Control Register Out	RHSBGjCRO0	<RHSB_XBARj_base> + 20 _H	8, 16, 32	—
RHSB XBARj	Microsecond Bus Control Register Out	RHSBGjCRO1	<RHSB_XBARj_base> + 24 _H	8, 16, 32	—
RHSB XBARj	Microsecond Bus Control Register Out	RHSBGjCRO2	<RHSB_XBARj_base> + 28 _H	8, 16, 32	—
RHSB XBARj	Microsecond Bus Control Register Out	RHSBGjCRO3	<RHSB_XBARj_base> + 2C _H	8, 16, 32	—

Note: j = 0, 1.

When ATU-V is used, see **Section 22.6.3.2, RHSBGjCRkH — Microsecond Bus Control Register H** to **Section 22.6.3.4, RHSBGjCROk — Microsecond Bus Control Register Out**.

When GTM is used, see **Section 22.6.3.5, RHSBGjCRkH — Microsecond Bus Control Register H** to **Section 22.6.3.7, RHSBGjCROk — Microsecond Bus Control Register Out**.

22.6.3.2 RHSBGjCRkH — Microsecond Bus Control Register H (ATU-V)

RHSBGjCRkH is a register that selects the upper bits of signals to be output to the RHSB from signals received from the ATU and GPIO. This register supports ATU timer C, timer D, timer E and GPIO signals.

Do not change the setting of the bits in this register while any of the ATU timer C, timer D, timer E, or GPIO are working.

j = 0, 1.

k = 0 to 3.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MjMDk15				MjMDk14				MjMDk13				MjMDk12			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MjMDk11				MjMDk10				MjMDk9				MjMDk8			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.40 RHSBGjCRkH Register (ATU-V) Contents (1/8)

Bit Position	Bit Name	Function
31 to 28	MjMDk15	<p>These bits select the signal of the fifteenth bit.</p> <p>0000: ATU_C sub_0 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 01)./ ATU_D sub_0 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 10)./ ATU_E sub_0 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 11).</p> <p>0001: ATU_C sub_1 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 01)./ ATU_D sub_1 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 10)./ ATU_E sub_1 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 11).</p> <p>0010: ATU_C sub_2 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 01)./ ATU_D sub_2 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 10)./ ATU_E sub_2 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 11).</p> <p>0011: ATU_C sub_3 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 01)./ ATU_D sub_3 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 10)./ ATU_E sub_3 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 11).</p> <p>0100: ATU_C sub_4 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 01)./ ATU_D sub_4 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 10)./ ATU_E sub_4 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 11).</p> <p>0101: ATU_C sub_5 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 01)./ ATU_D sub_5 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 10)./ ATU_E sub_5 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 11).</p> <p>0110: ATU_C sub_6 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 01)./ ATU_D sub_6 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 10)./ ATU_E sub_6 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 11).</p> <p>0111: ATU_C sub_7 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 01)./ ATU_D sub_7 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 10)./ ATU_E sub_7 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 11).</p> <p>1000: ATU_C sub_8 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 01)./ ATU_D sub_8 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 10)./ ATU_E sub_8 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 11).</p> <p>1001: ATU_C sub_9 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK15 = 11).</p> <p>1010: ATU_C sub_10 ch_3 is selected (RHSBGjCROk.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK15 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Table 22.40 RHSBGjCRkH Register (ATU-V) Contents (2/8)

Bit Position	Bit Name	Function
27 to 24	MjMDk14	<p>These bits select the signal of the fourteenth bit.</p> <p>0000: ATU_C sub_0 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 01)./ ATU_D sub_0 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 10)./ ATU_E sub_0 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 11).</p> <p>0001: ATU_C sub_1 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 01)./ ATU_D sub_1 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 10)./ ATU_E sub_1 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 11).</p> <p>0010: ATU_C sub_2 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 01)./ ATU_D sub_2 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 10)./ ATU_E sub_2 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 11).</p> <p>0011: ATU_C sub_3 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 01)./ ATU_D sub_3 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 10)./ ATU_E sub_3 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 11).</p> <p>0100: ATU_C sub_4 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 01)./ ATU_D sub_4 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 10)./ ATU_E sub_4 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 11).</p> <p>0101: ATU_C sub_5 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 01)./ ATU_D sub_5 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 10)./ ATU_E sub_5 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 11).</p> <p>0110: ATU_C sub_6 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 01)./ ATU_D sub_6 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 10)./ ATU_E sub_6 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 11).</p> <p>0111: ATU_C sub_7 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 01)./ ATU_D sub_7 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 10)./ ATU_E sub_7 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 11).</p> <p>1000: ATU_C sub_8 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 01)./ ATU_D sub_8 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 10)./ ATU_E sub_8 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 11).</p> <p>1001: ATU_C sub_9 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK14 = 11).</p> <p>1010: ATU_C sub_10 ch_2 is selected (RHSBGjCROk.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK14 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Table 22.40 RHSBGjCRkH Register (ATU-V) Contents (3/8)

Bit Position	Bit Name	Function
23 to 20	MjMDk13	<p>These bits select the signal of the thirteenth bit.</p> <p>0000: ATU_C sub_0 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 01)./ ATU_D sub_0 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 10)./ ATU_E sub_0 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 11).</p> <p>0001: ATU_C sub_1 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 01)./ ATU_D sub_1 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 10)./ ATU_E sub_1 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 11).</p> <p>0010: ATU_C sub_2 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 01)./ ATU_D sub_2 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 10)./ ATU_E sub_2 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 11).</p> <p>0011: ATU_C sub_3 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 01)./ ATU_D sub_3 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 10)./ ATU_E sub_3 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 11).</p> <p>0100: ATU_C sub_4 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 01)./ ATU_D sub_4 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 10)./ ATU_E sub_4 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 11).</p> <p>0101: ATU_C sub_5 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 01)./ ATU_D sub_5 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 10)./ ATU_E sub_5 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 11).</p> <p>0110: ATU_C sub_6 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 01)./ ATU_D sub_6 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 10)./ ATU_E sub_6 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 11).</p> <p>0111: ATU_C sub_7 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 01)./ ATU_D sub_7 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 10)./ ATU_E sub_7 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 11).</p> <p>1000: ATU_C sub_8 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 01)./ ATU_D sub_8 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 10)./ ATU_E sub_8 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 11).</p> <p>1001: ATU_C sub_9 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK13 = 11).</p> <p>1010: ATU_C sub_10 ch_1 is selected (RHSBGjCROk.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK13 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Table 22.40 RHSBGjCRkH Register (ATU-V) Contents (4/8)

Bit Position	Bit Name	Function
19 to 16	MjMDk12	<p>These bits select the signal of the twelfth bit.</p> <p>0000: ATU_C sub_0 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 01)./ ATU_D sub_0 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 10)./ ATU_E sub_0 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 11).</p> <p>0001: ATU_C sub_1 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 01)./ ATU_D sub_1 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 10)./ ATU_E sub_1 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 11).</p> <p>0010: ATU_C sub_2 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 01)./ ATU_D sub_2 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 10)./ ATU_E sub_2 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 11).</p> <p>0011: ATU_C sub_3 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 01)./ ATU_D sub_3 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 10)./ ATU_E sub_3 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 11).</p> <p>0100: ATU_C sub_4 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 01)./ ATU_D sub_4 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 10)./ ATU_E sub_4 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 11).</p> <p>0101: ATU_C sub_5 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 01)./ ATU_D sub_5 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 10)./ ATU_E sub_5 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 11).</p> <p>0110: ATU_C sub_6 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 01)./ ATU_D sub_6 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 10)./ ATU_E sub_6 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 11).</p> <p>0111: ATU_C sub_7 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 01)./ ATU_D sub_7 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 10)./ ATU_E sub_7 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 11).</p> <p>1000: ATU_C sub_8 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 01)./ ATU_D sub_8 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 10)./ ATU_E sub_8 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 11).</p> <p>1001: ATU_C sub_9 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK12 = 11).</p> <p>1010: ATU_C sub_10 ch_0 is selected (RHSBGjCROk.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK12 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Table 22.40 RHSBGjCRkH Register (ATU-V) Contents (5/8)

Bit Position	Bit Name	Function
15 to 12	MjMDk11	<p>These bits select the signal of the eleventh bit.</p> <p>0000: ATU_C sub_0 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 01)./ ATU_D sub_0 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 10)./ ATU_E sub_0 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 11).</p> <p>0001: ATU_C sub_1 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 01)./ ATU_D sub_1 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 10)./ ATU_E sub_1 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 11).</p> <p>0010: ATU_C sub_2 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 01)./ ATU_D sub_2 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 10)./ ATU_E sub_2 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 11).</p> <p>0011: ATU_C sub_3 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 01)./ ATU_D sub_3 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 10)./ ATU_E sub_3 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 11).</p> <p>0100: ATU_C sub_4 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 01)./ ATU_D sub_4 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 10)./ ATU_E sub_4 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 11).</p> <p>0101: ATU_C sub_5 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 01)./ ATU_D sub_5 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 10)./ ATU_E sub_5 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 11).</p> <p>0110: ATU_C sub_6 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 01)./ ATU_D sub_6 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 10)./ ATU_E sub_6 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 11).</p> <p>0111: ATU_C sub_7 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 01)./ ATU_D sub_7 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 10)./ ATU_E sub_7 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 11).</p> <p>1000: ATU_C sub_8 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 01)./ ATU_D sub_8 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 10)./ ATU_E sub_8 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 11).</p> <p>1001: ATU_C sub_9 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK11 = 11).</p> <p>1010: ATU_C sub_10 ch_3 is selected (RHSBGjCROk.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK11 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Table 22.40 RHSBGjCRkH Register (ATU-V) Contents (6/8)

Bit Position	Bit Name	Function
11 to 8	MjMDk10	<p>These bits select the signal of the tenth bit.</p> <p>0000: ATU_C sub_0 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 01)./ ATU_D sub_0 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 10)./ ATU_E sub_0 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 11).</p> <p>0001: ATU_C sub_1 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 01)./ ATU_D sub_1 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 10)./ ATU_E sub_1 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 11).</p> <p>0010: ATU_C sub_2 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 01)./ ATU_D sub_2 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 10)./ ATU_E sub_2 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 11).</p> <p>0011: ATU_C sub_3 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 01)./ ATU_D sub_3 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 10)./ ATU_E sub_3 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 11).</p> <p>0100: ATU_C sub_4 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 01)./ ATU_D sub_4 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 10)./ ATU_E sub_4 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 11).</p> <p>0101: ATU_C sub_5 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 01)./ ATU_D sub_5 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 10)./ ATU_E sub_5 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 11).</p> <p>0110: ATU_C sub_6 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 01)./ ATU_D sub_6 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 10)./ ATU_E sub_6 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 11).</p> <p>0111: ATU_C sub_7 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 01)./ ATU_D sub_7 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 10)./ ATU_E sub_7 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 11).</p> <p>1000: ATU_C sub_8 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 01)./ ATU_D sub_8 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 10)./ ATU_E sub_8 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 11).</p> <p>1001: ATU_C sub_9 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK10 = 11).</p> <p>1010: ATU_C sub_10 ch_2 is selected (RHSBGjCROk.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK10 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Table 22.40 RHSBGjCRkH Register (ATU-V) Contents (7/8)

Bit Position	Bit Name	Function
7 to 4	MjMDk9	<p>These bits select the signal of the ninth bit.</p> <p>0000: ATU_C sub_0 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 01)./ ATU_D sub_0 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 10)./ ATU_E sub_0 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 11).</p> <p>0001: ATU_C sub_1 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 01)./ ATU_D sub_1 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 10)./ ATU_E sub_1 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 11).</p> <p>0010: ATU_C sub_2 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 01)./ ATU_D sub_2 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 10)./ ATU_E sub_2 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 11).</p> <p>0011: ATU_C sub_3 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 01)./ ATU_D sub_3 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 10)./ ATU_E sub_3 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 11).</p> <p>0100: ATU_C sub_4 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 01)./ ATU_D sub_4 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 10)./ ATU_E sub_4 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 11).</p> <p>0101: ATU_C sub_5 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 01)./ ATU_D sub_5 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 10)./ ATU_E sub_5 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 11).</p> <p>0110: ATU_C sub_6 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 01)./ ATU_D sub_6 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 10)./ ATU_E sub_6 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 11).</p> <p>0111: ATU_C sub_7 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 01)./ ATU_D sub_7 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 10)./ ATU_E sub_7 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 11).</p> <p>1000: ATU_C sub_8 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 01)./ ATU_D sub_8 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 10)./ ATU_E sub_8 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 11).</p> <p>1001: ATU_C sub_9 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk9 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk9 = 11).</p> <p>1010: ATU_C sub_10 ch_1 is selected (RHSBGjCROk.MjMDOk9 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk9 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk9 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Table 22.40 RHSBGjCRkH Register (ATU-V) Contents (8/8)

Bit Position	Bit Name	Function
3 to 0	MjMDk8	<p>These bits select the signal of the eighth bit.</p> <p>0000: ATU_C sub_0 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 01)./ ATU_D sub_0 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 10)./ ATU_E sub_0 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 11).</p> <p>0001: ATU_C sub_1 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 01)./ ATU_D sub_1 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 10)./ ATU_E sub_1 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 11).</p> <p>0010: ATU_C sub_2 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 01)./ ATU_D sub_2 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 10)./ ATU_E sub_2 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 11).</p> <p>0011: ATU_C sub_3 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 01)./ ATU_D sub_3 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 10)./ ATU_E sub_3 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 11).</p> <p>0100: ATU_C sub_4 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 01)./ ATU_D sub_4 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 10)./ ATU_E sub_4 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 11).</p> <p>0101: ATU_C sub_5 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 01)./ ATU_D sub_5 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 10)./ ATU_E sub_5 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 11).</p> <p>0110: ATU_C sub_6 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 01)./ ATU_D sub_6 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 10)./ ATU_E sub_6 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 11).</p> <p>0111: ATU_C sub_7 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 01)./ ATU_D sub_7 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 10)./ ATU_E sub_7 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 11).</p> <p>1000: ATU_C sub_8 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 01)./ ATU_D sub_8 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 10)./ ATU_E sub_8 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 11).</p> <p>1001: ATU_C sub_9 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk8 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk8 = 11).</p> <p>1010: ATU_C sub_10 ch_0 is selected (RHSBGjCROk.MjMDOk8 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk8 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk8 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Note 1. If this is selected, the low level is output.

22.6.3.3 RHSBGjCRkL — Microsecond Bus Control Register L (ATU-V)

RHSBGjCRkL is a register that selects the lower bits of the signals to be output to the RHSB from signals received from the ATU and GPIO. This register supports ATU timer C, timer D, timer E and GPIO signals.

Do not change the settings of the bits in this register while any of the ATU timer C, timer D, timer E, and GPIO are working.

RHSBGjCRkL can be read and written in byte units.

j = 0, 1.

k = 0 to 3.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MjMDk7				MjMDk6				MjMDk5				MjMDk4			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MjMDk3				MjMDk2				MjMDk1				MjMDk0			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.41 RHSBGjCRkL Register (ATU-V) Contents (1/8)

Bit Position	Bit Name	Function
31 to 28	MjMDk7	<p>These bits select the signal of the seventh bit.</p> <p>0000: ATU_C sub_0 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 01)./ ATU_D sub_0 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 10)./ ATU_E sub_0 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 11).</p> <p>0001: ATU_C sub_1 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 01)./ ATU_D sub_1 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 10)./ ATU_E sub_1 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 11).</p> <p>0010: ATU_C sub_2 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 01)./ ATU_D sub_2 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 10)./ ATU_E sub_2 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 11).</p> <p>0011: ATU_C sub_3 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 01)./ ATU_D sub_3 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 10)./ ATU_E sub_3 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 11).</p> <p>0100: ATU_C sub_4 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 01)./ ATU_D sub_4 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 10)./ ATU_E sub_4 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 11).</p> <p>0101: ATU_C sub_5 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 01)./ ATU_D sub_5 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 10)./ ATU_E sub_5 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 11).</p> <p>0110: ATU_C sub_6 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 01)./ ATU_D sub_6 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 10)./ ATU_E sub_6 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 11).</p> <p>0111: ATU_C sub_7 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 01)./ ATU_D sub_7 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 10)./ ATU_E sub_7 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 11).</p> <p>1000: ATU_C sub_8 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 01)./ ATU_D sub_8 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 10)./ ATU_E sub_8 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 11).</p> <p>1001: ATU_C sub_9 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK7 = 11).</p> <p>1010: ATU_C sub_10 ch_3 is selected (RHSBGjCROk.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK7 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Table 22.41 RHSBGjCRkL Register (ATU-V) Contents (2/8)

Bit Position	Bit Name	Function
27 to 24	MjMDk6	<p>These bits select the signal of the sixth bit.</p> <p>0000: ATU_C sub_0 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 01)./ ATU_D sub_0 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 10)./ ATU_E sub_0 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 11).</p> <p>0001: ATU_C sub_1 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 01)./ ATU_D sub_1 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 10)./ ATU_E sub_1 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 11).</p> <p>0010: ATU_C sub_2 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 01)./ ATU_D sub_2 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 10)./ ATU_E sub_2 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 11).</p> <p>0011: ATU_C sub_3 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 01)./ ATU_D sub_3 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 10)./ ATU_E sub_3 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 11).</p> <p>0100: ATU_C sub_4 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 01)./ ATU_D sub_4 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 10)./ ATU_E sub_4 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 11).</p> <p>0101: ATU_C sub_5 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 01)./ ATU_D sub_5 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 10)./ ATU_E sub_5 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 11).</p> <p>0110: ATU_C sub_6 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 01)./ ATU_D sub_6 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 10)./ ATU_E sub_6 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 11).</p> <p>0111: ATU_C sub_7 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 01)./ ATU_D sub_7 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 10)./ ATU_E sub_7 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 11).</p> <p>1000: ATU_C sub_8 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 01)./ ATU_D sub_8 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 10)./ ATU_E sub_8 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 11).</p> <p>1001: ATU_C sub_9 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk6 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk6 = 11).</p> <p>1010: ATU_C sub_10 ch_2 is selected (RHSBGjCROk.MjMDOk6 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk6 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk6 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Table 22.41 RHBGjCRkL Register (ATU-V) Contents (3/8)

Bit Position	Bit Name	Function
23 to 20	MjMDk5	<p>These bits select the signal of the fifth bit.</p> <p>0000: ATU_C sub_0 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 01)./ ATU_D sub_0 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 10)./ ATU_E sub_0 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 11).</p> <p>0001: ATU_C sub_1 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 01)./ ATU_D sub_1 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 10)./ ATU_E sub_1 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 11).</p> <p>0010: ATU_C sub_2 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 01)./ ATU_D sub_2 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 10)./ ATU_E sub_2 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 11).</p> <p>0011: ATU_C sub_3 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 01)./ ATU_D sub_3 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 10)./ ATU_E sub_3 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 11).</p> <p>0100: ATU_C sub_4 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 01)./ ATU_D sub_4 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 10)./ ATU_E sub_4 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 11).</p> <p>0101: ATU_C sub_5 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 01)./ ATU_D sub_5 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 10)./ ATU_E sub_5 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 11).</p> <p>0110: ATU_C sub_6 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 01)./ ATU_D sub_6 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 10)./ ATU_E sub_6 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 11).</p> <p>0111: ATU_C sub_7 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 01)./ ATU_D sub_7 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 10)./ ATU_E sub_7 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 11).</p> <p>1000: ATU_C sub_8 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 01)./ ATU_D sub_8 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 10)./ ATU_E sub_8 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 11).</p> <p>1001: ATU_C sub_9 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk5 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk5 = 11).</p> <p>1010: ATU_C sub_10 ch_1 is selected (RHSBGjCROk.MjMDOk5 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk5 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk5 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Table 22.41 RHBGjCRkL Register (ATU-V) Contents (4/8)

Bit Position	Bit Name	Function
19 to 16	MjMDk4	<p>These bits select the signal of the fourth bit.</p> <p>0000: ATU_C sub_0 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 01)./ ATU_D sub_0 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 10)./ ATU_E sub_0 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 11).</p> <p>0001: ATU_C sub_1 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 01)./ ATU_D sub_1 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 10)./ ATU_E sub_1 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 11).</p> <p>0010: ATU_C sub_2 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 01)./ ATU_D sub_2 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 10)./ ATU_E sub_2 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 11).</p> <p>0011: ATU_C sub_3 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 01)./ ATU_D sub_3 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 10)./ ATU_E sub_3 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 11).</p> <p>0100: ATU_C sub_4 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 01)./ ATU_D sub_4 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 10)./ ATU_E sub_4 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 11).</p> <p>0101: ATU_C sub_5 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 01)./ ATU_D sub_5 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 10)./ ATU_E sub_5 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 11).</p> <p>0110: ATU_C sub_6 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 01)./ ATU_D sub_6 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 10)./ ATU_E sub_6 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 11).</p> <p>0111: ATU_C sub_7 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 01)./ ATU_D sub_7 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 10)./ ATU_E sub_7 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 11).</p> <p>1000: ATU_C sub_8 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 01)./ ATU_D sub_8 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 10)./ ATU_E sub_8 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 11).</p> <p>1001: ATU_C sub_9 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk4 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk4 = 11).</p> <p>1010: ATU_C sub_10 ch_0 is selected (RHSBGjCROk.MjMDOk4 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk4 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOk4 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Table 22.41 RHSBGjCRkL Register (ATU-V) Contents (5/8)

Bit Position	Bit Name	Function
15 to 12	MjMDk3	<p>These bits select the signal of the third bit.</p> <p>0000: ATU_C sub_0 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 01)./ ATU_D sub_0 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 10)./ ATU_E sub_0 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 11).</p> <p>0001: ATU_C sub_1 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 01)./ ATU_D sub_1 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 10)./ ATU_E sub_1 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 11).</p> <p>0010: ATU_C sub_2 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 01)./ ATU_D sub_2 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 10)./ ATU_E sub_2 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 11).</p> <p>0011: ATU_C sub_3 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 01)./ ATU_D sub_3 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 10)./ ATU_E sub_3 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 11).</p> <p>0100: ATU_C sub_4 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 01)./ ATU_D sub_4 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 10)./ ATU_E sub_4 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 11).</p> <p>0101: ATU_C sub_5 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 01)./ ATU_D sub_5 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 10)./ ATU_E sub_5 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 11).</p> <p>0110: ATU_C sub_6 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 01)./ ATU_D sub_6 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 10)./ ATU_E sub_6 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 11).</p> <p>0111: ATU_C sub_7 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 01)./ ATU_D sub_7 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 10)./ ATU_E sub_7 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 11).</p> <p>1000: ATU_C sub_8 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 01)./ ATU_D sub_8 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 10)./ ATU_E sub_8 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 11).</p> <p>1001: ATU_C sub_9 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK3 = 11).</p> <p>1010: ATU_C sub_10 ch_3 is selected (RHSBGjCROk.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK3 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Table 22.41 RHBGjCRkL Register (ATU-V) Contents (6/8)

Bit Position	Bit Name	Function
11 to 8	MjMDk2	<p>These bits select the signal of the second bit.</p> <p>0000: ATU_C sub_0 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 01)./ ATU_D sub_0 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 10)./ ATU_E sub_0 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 11).</p> <p>0001: ATU_C sub_1 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 01)./ ATU_D sub_1 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 10)./ ATU_E sub_1 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 11).</p> <p>0010: ATU_C sub_2 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 01)./ ATU_D sub_2 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 10)./ ATU_E sub_2 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 11).</p> <p>0011: ATU_C sub_3 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 01)./ ATU_D sub_3 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 10)./ ATU_E sub_3 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 11).</p> <p>0100: ATU_C sub_4 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 01)./ ATU_D sub_4 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 10)./ ATU_E sub_4 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 11).</p> <p>0101: ATU_C sub_5 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 01)./ ATU_D sub_5 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 10)./ ATU_E sub_5 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 11).</p> <p>0110: ATU_C sub_6 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 01)./ ATU_D sub_6 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 10)./ ATU_E sub_6 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 11).</p> <p>0111: ATU_C sub_7 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 01)./ ATU_D sub_7 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 10)./ ATU_E sub_7 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 11).</p> <p>1000: ATU_C sub_8 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 01)./ ATU_D sub_8 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 10)./ ATU_E sub_8 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 11).</p> <p>1001: ATU_C sub_9 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK2 = 11).</p> <p>1010: ATU_C sub_10 ch_2 is selected (RHSBGjCROk.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK2 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Table 22.41 RHSBGjCRkL Register (ATU-V) Contents (7/8)

Bit Position	Bit Name	Function
7 to 4	MjMDk1	<p>These bits select the signal of the first bit.</p> <p>0000: ATU_C sub_0 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 01)./ ATU_D sub_0 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 10)./ ATU_E sub_0 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 11).</p> <p>0001: ATU_C sub_1 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 01)./ ATU_D sub_1 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 10)./ ATU_E sub_1 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 11).</p> <p>0010: ATU_C sub_2 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 01)./ ATU_D sub_2 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 10)./ ATU_E sub_2 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 11).</p> <p>0011: ATU_C sub_3 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 01)./ ATU_D sub_3 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 10)./ ATU_E sub_3 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 11).</p> <p>0100: ATU_C sub_4 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 01)./ ATU_D sub_4 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 10)./ ATU_E sub_4 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 11).</p> <p>0101: ATU_C sub_5 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 01)./ ATU_D sub_5 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 10)./ ATU_E sub_5 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 11).</p> <p>0110: ATU_C sub_6 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 01)./ ATU_D sub_6 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 10)./ ATU_E sub_6 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 11).</p> <p>0111: ATU_C sub_7 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 01)./ ATU_D sub_7 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 10)./ ATU_E sub_7 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 11).</p> <p>1000: ATU_C sub_8 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 01)./ ATU_D sub_8 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 10)./ ATU_E sub_8 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 11).</p> <p>1001: ATU_C sub_9 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK1 = 11).</p> <p>1010: ATU_C sub_10 ch_1 is selected (RHSBGjCROk.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK1 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Table 22.41 RHBGjCRkL Register (ATU-V) Contents (8/8)

Bit Position	Bit Name	Function
3 to 0	MjMDk0	<p>These bits select the signal of the zeroth bit.</p> <p>0000: ATU_C sub_0 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 01)./ ATU_D sub_0 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 10)./ ATU_E sub_0 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 11).</p> <p>0001: ATU_C sub_1 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 01)./ ATU_D sub_1 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 10)./ ATU_E sub_1 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 11).</p> <p>0010: ATU_C sub_2 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 01)./ ATU_D sub_2 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 10)./ ATU_E sub_2 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 11).</p> <p>0011: ATU_C sub_3 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 01)./ ATU_D sub_3 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 10)./ ATU_E sub_3 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 11).</p> <p>0100: ATU_C sub_4 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 01)./ ATU_D sub_4 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 10)./ ATU_E sub_4 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 11).</p> <p>0101: ATU_C sub_5 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 01)./ ATU_D sub_5 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 10)./ ATU_E sub_5 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 11).</p> <p>0110: ATU_C sub_6 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 01)./ ATU_D sub_6 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 10)./ ATU_E sub_6 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 11).</p> <p>0111: ATU_C sub_7 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 01)./ ATU_D sub_7 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 10)./ ATU_E sub_7 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 11).</p> <p>1000: ATU_C sub_8 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 01)./ ATU_D sub_8 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 10)./ ATU_E sub_8 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 11).</p> <p>1001: ATU_C sub_9 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK0 = 11).</p> <p>1010: ATU_C sub_10 ch_0 is selected (RHSBGjCROk.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROk.MjMDOK0 = 11).</p> <p>1011: Setting prohibited.*1</p> <p>1100: Setting prohibited.*1</p> <p>1101: Setting prohibited.*1</p> <p>1110: Setting prohibited.*1</p> <p>1111: Setting prohibited.*1</p>

Note 1. If this is selected, the low level is output.

22.6.3.4 RHSBGjCROk — Microsecond Bus Control Register Out (ATU-V)

RHSBGjCROk is a register that selects signals to be output to the RHSB from signals received from the ATU and GPIO. This register supports ATU timer C, timer D, timer E and GPIO signals.

Do not change the settings of the bits in this register while any of the ATU timer C, timer D, timer E or GPIO are working.

$j = 0, 1.$

$k = 0$ to 3.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MjMDOK15		MjMDOK14		MjMDOK13		MjMDOK12		MjMDOK11		MjMDOK10		MjMDOK9		MjMDOK8	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MjMDOK7		MjMDOK6		MjMDOK5		MjMDOK4		MjMDOK3		MjMDOK2		MjMDOK1		MjMDOK0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.42 RHSBGjCROk Register (ATU-V) Contents (1/4)

Bit Position	Bit Name	Function
31 to 30	MjMDOK15	These bits select the signal of the fifteenth bit. 00: P50_[15] is selected (k = 0)/ P51_[15] is selected (k = 1)/ P52_[15] is selected (k = 2)/ P53_[15] is selected (k = 3). 01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected
29 to 28	MjMDOK14	These bits select the signal of the fourteenth bit. 00: P50_[14] is selected (k = 0)/ P51_[14] is selected (k = 1)/ P52_[14] is selected (k = 2)/ P53_[14] is selected (k = 3). 01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected
27 to 26	MjMDOK13	These bits select the signal of the thirteenth bit. 00: P50_[13] is selected (k = 0)/ P51_[13] is selected (k = 1)/ P52_[13] is selected (k = 2)/ P53_[13] is selected (k = 3). 01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected

Table 22.42 RHBGjCROk Register (ATU-V) Contents (2/4)

Bit Position	Bit Name	Function
25 to 24	MjMDOK12	<p>These bits select the signal of the twelfth bit.</p> <p>00: P50_[12] is selected (k = 0)/ P51_[12] is selected (k = 1)/ P52_[12] is selected (k = 2)/ P53_[12] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected</p>
23 to 22	MjMDOK11	<p>These bits select the signal of the eleventh bit.</p> <p>00: P50_[11] is selected (k = 0)/ P51_[11] is selected (k = 1)/ P52_[11] is selected (k = 2)/ P53_[11] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected</p>
21 to 20	MjMDOK10	<p>These bits select the signal of the tenth bit.</p> <p>00: P50_[10] is selected (k = 0)/ P51_[10] is selected (k = 1)/ P52_[10] is selected (k = 2)/ P53_[10] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected</p>
19 to 18	MjMDOK9	<p>These bits select the signal of the ninth bit.</p> <p>00: P50_[9] is selected (k = 0)/ P51_[9] is selected (k = 1)/ P52_[9] is selected (k = 2)/ P53_[9] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected</p>
17 to 16	MjMDOK8	<p>These bits select the signal of the eighth bit.</p> <p>00: P50_[8] is selected (k = 0)/ P51_[8] is selected (k = 1)/ P52_[8] is selected (k = 2)/ P53_[8] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected</p>
15 to 14	MjMDOK7	<p>These bits select the signal of the seventh bit.</p> <p>00: P50_[7] is selected (k = 0)/ P51_[7] is selected (k = 1)/ P52_[7] is selected (k = 2)/ P53_[7] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected</p>

Table 22.42 RHSBGjCROk Register (ATU-V) Contents (3/4)

Bit Position	Bit Name	Function
13 to 12	MjMDOk6	<p>These bits select the signal of the sixth bit.</p> <p>00: P50_[6] is selected (k = 0)/ P51_[6] is selected (k = 1)/ P52_[6] is selected (k = 2)/ P53_[6] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected</p>
11 to 10	MjMDOk5	<p>These bits select the signal of the fifth bit.</p> <p>00: P50_[5] is selected (k = 0)/ P51_[5] is selected (k = 1)/ P52_[5] is selected (k = 2)/ P53_[5] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected</p>
9 to 8	MjMDOk4	<p>These bits select the signal of the fourth bit.</p> <p>00: P50_[4] is selected (k = 0)/ P51_[4] is selected (k = 1)/ P52_[4] is selected (k = 2)/ P53_[4] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected</p>
7 to 6	MjMDOk3	<p>These bits select the signal of the third bit.</p> <p>00: P50_[3] is selected (k = 0)/ P51_[3] is selected (k = 1)/ P52_[3] is selected (k = 2)/ P53_[3] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected</p>
5 to 4	MjMDOk2	<p>These bits select the signal of the second bit.</p> <p>00: P50_[2] is selected (k = 0)/ P51_[2] is selected (k = 1)/ P52_[2] is selected (k = 2)/ P53_[2] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected</p>
3 to 2	MjMDOk1	<p>These bits select the signal of the first bit.</p> <p>00: P50_[1] is selected (k = 0)/ P51_[1] is selected (k = 1)/ P52_[1] is selected (k = 2)/ P53_[1] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected</p>

Table 22.42 RHBGjCROk Register (ATU-V) Contents (4/4)

Bit Position	Bit Name	Function
1 to 0	MjMDOk0	These bits select the signal of the zeroth bit. 00: P50_[0] is selected (k = 0)/ P51_[0] is selected (k = 1)/ P52_[0] is selected (k = 2)/ P53_[0] is selected (k = 3). 01: Select output from Table 1-1 is selected 10: Select output from Table 1-2 is selected 11: Select output from Table 1-3 is selected

22.6.3.5 RHSBjCRkH — Microsecond Bus Control Register H (GTM)

RHSBjCRkH is a register that selects the upper bits of signals to be output to the RHSB from signals received from the GTM and GPIO. This register supports GTM TOM, GTM ATOM and GPIO signals.

Do not change the settings of the bits in this register while any of GTM TOM, GTM ATOM or GPIO are working.

$j = 0, 1.$

$k = 0$ to $3.$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MjMDk15				MjMDk14				MjMDk13				MjMDk12			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MjMDk11				MjMDk10				MjMDk9				MjMDk8			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.43 RHSBGjCRkH Register (GTM) Contents (1/8)

Bit Position	Bit Name	Function
31 to 28	MjMDk15	<p>These bits select the signal of the fifteenth bit.</p> <p>0000: GTM TOM sub_0 ch_7 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p> <p>0001: GTM TOM sub_0 ch_15 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p> <p>0010: GTM TOM sub_1 ch_7 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p> <p>0011: GTM TOM sub_1 ch_15 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p> <p>0100: GTM TOM sub_2 ch_7 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p> <p>0101: GTM TOM sub_2 ch_15 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p> <p>0110: GTM TOM sub_3 ch_7 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p> <p>0111: GTM TOM sub_3 ch_15 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p> <p>1000: GTM ATOM sub_0 ch_7 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p> <p>1001: GTM ATOM sub_1 ch_7 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p> <p>1010: GTM ATOM sub_2 ch_7 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p> <p>1011: GTM ATOM sub_3 ch_7 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p> <p>1100: GTM ATOM sub_4 ch_7 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p> <p>1101: GTM ATOM sub_5 ch_7 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p> <p>1110: GTM ATOM sub_6 ch_7 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p> <p>1111: GTM ATOM sub_7 ch_7 is selected (RHSBGjCROK.MjMDOK15 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK15 = 11).</p>

Table 22.43 RHSBGjCRkH Register (GTM) Contents (2/8)

Bit Position	Bit Name	Function
27 to 24	MjMDk14	<p>These bits select the signal of the fourteenth bit.</p> <p>0000: GTM TOM sub_0 ch_6 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p> <p>0001: GTM TOM sub_0 ch_14 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p> <p>0010: GTM TOM sub_1 ch_6 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p> <p>0011: GTM TOM sub_1 ch_14 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p> <p>0100: GTM TOM sub_2 ch_6 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p> <p>0101: GTM TOM sub_2 ch_14 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p> <p>0110: GTM TOM sub_3 ch_6 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p> <p>0111: GTM TOM sub_3 ch_14 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p> <p>1000: GTM ATOM sub_0 ch_6 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p> <p>1001: GTM ATOM sub_1 ch_6 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p> <p>1010: GTM ATOM sub_2 ch_6 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p> <p>1011: GTM ATOM sub_3 ch_6 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p> <p>1100: GTM ATOM sub_4 ch_6 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p> <p>1101: GTM ATOM sub_5 ch_6 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p> <p>1110: GTM ATOM sub_6 ch_6 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p> <p>1111: GTM ATOM sub_7 ch_6 is selected (RHSBGjCROK.MjMDOK14 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK14 = 11).</p>

Table 22.43 RHSBGjCRkH Register (GTM) Contents (3/8)

Bit Position	Bit Name	Function
23 to 20	MjMDk13	<p>These bits select the signal of the thirteenth bit.</p> <p>0000: GTM TOM sub_0 ch_5 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p> <p>0001: GTM TOM sub_0 ch_13 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p> <p>0010: GTM TOM sub_1 ch_5 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p> <p>0011: GTM TOM sub_1 ch_13 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p> <p>0100: GTM TOM sub_2 ch_5 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p> <p>0101: GTM TOM sub_2 ch_13 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p> <p>0110: GTM TOM sub_3 ch_5 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p> <p>0111: GTM TOM sub_3 ch_13 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p> <p>1000: GTM ATOM sub_0 ch_5 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p> <p>1001: GTM ATOM sub_1 ch_5 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p> <p>1010: GTM ATOM sub_2 ch_5 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p> <p>1011: GTM ATOM sub_3 ch_5 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p> <p>1100: GTM ATOM sub_4 ch_5 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p> <p>1101: GTM ATOM sub_5 ch_5 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p> <p>1110: GTM ATOM sub_6 ch_5 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p> <p>1111: GTM ATOM sub_7 ch_5 is selected (RHSBGjCROK.MjMDOK13 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK13 = 11).</p>

Table 22.43 RHSBGjCRkH Register (GTM) Contents (4/8)

Bit Position	Bit Name	Function
19 to 16	MjMDk12	<p>These bits select the signal of the twelfth bit.</p> <p>0000: GTM TOM sub_0 ch_4 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p> <p>0001: GTM TOM sub_0 ch_12 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p> <p>0010: GTM TOM sub_1 ch_4 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p> <p>0011: GTM TOM sub_1 ch_12 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p> <p>0100: GTM TOM sub_2 ch_4 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p> <p>0101: GTM TOM sub_2 ch_12 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p> <p>0110: GTM TOM sub_3 ch_4 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p> <p>0111: GTM TOM sub_3 ch_12 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p> <p>1000: GTM ATOM sub_0 ch_4 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p> <p>1001: GTM ATOM sub_1 ch_4 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p> <p>1010: GTM ATOM sub_2 ch_4 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p> <p>1011: GTM ATOM sub_3 ch_4 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p> <p>1100: GTM ATOM sub_4 ch_4 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p> <p>1101: GTM ATOM sub_5 ch_4 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p> <p>1110: GTM ATOM sub_6 ch_4 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p> <p>1111: GTM ATOM sub_7 ch_4 is selected (RHSBGjCROK.MjMDOK12 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK12 = 11).</p>

Table 22.43 RHSBGjCRkH Register (GTM) Contents (5/8)

Bit Position	Bit Name	Function
15 to 12	MjMDk11	<p>These bits select the signal of the eleventh bit.</p> <p>0000: GTM TOM sub_0 ch_3 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p> <p>0001: GTM TOM sub_0 ch_11 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p> <p>0010: GTM TOM sub_1 ch_3 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p> <p>0011: GTM TOM sub_1 ch_11 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p> <p>0100: GTM TOM sub_2 ch_3 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p> <p>0101: GTM TOM sub_2 ch_11 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p> <p>0110: GTM TOM sub_3 ch_3 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p> <p>0111: GTM TOM sub_3 ch_11 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p> <p>1000: GTM ATOM sub_0 ch_3 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p> <p>1001: GTM ATOM sub_1 ch_3 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p> <p>1010: GTM ATOM sub_2 ch_3 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p> <p>1011: GTM ATOM sub_3 ch_3 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p> <p>1100: GTM ATOM sub_4 ch_3 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p> <p>1101: GTM ATOM sub_5 ch_3 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p> <p>1110: GTM ATOM sub_6 ch_3 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p> <p>1111: GTM ATOM sub_7 ch_3 is selected (RHSBGjCROK.MjMDOK11 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK11 = 11).</p>

Table 22.43 RHSBGjCRkH Register (GTM) Contents (6/8)

Bit Position	Bit Name	Function
11 to 8	MjMDk10	<p>These bits select the signal of the tenth bit.</p> <p>0000: GTM TOM sub_0 ch_2 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p> <p>0001: GTM TOM sub_0 ch_10 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p> <p>0010: GTM TOM sub_1 ch_2 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p> <p>0011: GTM TOM sub_1 ch_10 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p> <p>0100: GTM TOM sub_2 ch_2 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p> <p>0101: GTM TOM sub_2 ch_10 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p> <p>0110: GTM TOM sub_3 ch_2 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p> <p>0111: GTM TOM sub_3 ch_10 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p> <p>1000: GTM ATOM sub_0 ch_2 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p> <p>1001: GTM ATOM sub_1 ch_2 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p> <p>1010: GTM ATOM sub_2 ch_2 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p> <p>1011: GTM ATOM sub_3 ch_2 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p> <p>1100: GTM ATOM sub_4 ch_2 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p> <p>1101: GTM ATOM sub_5 ch_2 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p> <p>1110: GTM ATOM sub_6 ch_2 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p> <p>1111: GTM ATOM sub_7 ch_2 is selected (RHSBGjCROK.MjMDOK10 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK10 = 11).</p>

Table 22.43 RHSBGjCRkH Register (GTM) Contents (7/8)

Bit Position	Bit Name	Function
7 to 4	MjMDk9	<p>These bits select the signal of the ninth bit.</p> <p>0000: GTM TOM sub_0 ch_1 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p> <p>0001: GTM TOM sub_0 ch_9 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p> <p>0010: GTM TOM sub_1 ch_1 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p> <p>0011: GTM TOM sub_1 ch_9 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p> <p>0100: GTM TOM sub_2 ch_1 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p> <p>0101: GTM TOM sub_2 ch_9 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p> <p>0110: GTM TOM sub_3 ch_1 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p> <p>0111: GTM TOM sub_3 ch_9 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p> <p>1000: GTM ATOM sub_0 ch_1 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p> <p>1001: GTM ATOM sub_1 ch_1 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p> <p>1010: GTM ATOM sub_2 ch_1 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p> <p>1011: GTM ATOM sub_3 ch_1 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p> <p>1100: GTM ATOM sub_4 ch_1 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p> <p>1101: GTM ATOM sub_5 ch_1 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p> <p>1110: GTM ATOM sub_6 ch_1 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p> <p>1111: GTM ATOM sub_7 ch_1 is selected (RHSBGjCROK.MjMDOK9 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK9 = 11).</p>

Table 22.43 RHSBGjCRkH Register (GTM) Contents (8/8)

Bit Position	Bit Name	Function
3 to 0	MjMDk8	<p>These bits select the signal of the eighth bit.</p> <p>0000: GTM TOM sub_0 ch_0 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p> <p>0001: GTM TOM sub_0 ch_8 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p> <p>0010: GTM TOM sub_1 ch_0 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p> <p>0011: GTM TOM sub_1 ch_8 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p> <p>0100: GTM TOM sub_2 ch_0 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p> <p>0101: GTM TOM sub_2 ch_8 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p> <p>0110: GTM TOM sub_3 ch_0 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p> <p>0111: GTM TOM sub_3 ch_8 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p> <p>1000: GTM ATOM sub_0 ch_0 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p> <p>1001: GTM ATOM sub_1 ch_0 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p> <p>1010: GTM ATOM sub_2 ch_0 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p> <p>1011: GTM ATOM sub_3 ch_0 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p> <p>1100: GTM ATOM sub_4 ch_0 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p> <p>1101: GTM ATOM sub_5 ch_0 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p> <p>1110: GTM ATOM sub_6 ch_0 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p> <p>1111: GTM ATOM sub_7 ch_0 is selected (RHSBGjCROK.MjMDOK8 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK8 = 11).</p>

Note 1. If this is selected, the low level is output.

22.6.3.6 RHSBjCRkL — Microsecond Bus Control Register L (GTM)

RHSBjCRkL is a register that selects the lower bits of signals to be output to the RHSB from signals received from the GTM and GPIO. This register supports GTM TOM, GTM ATOM and GPIO signals.

Do not change the setting of the bits in this register while any of GTM TOM, GTM ATOM or GPIO are working.

$j = 0, 1.$

$k = 0$ to $3.$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MjMDk7				MjMDk6				MjMDk5				MjMDk4			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MjMDk3				MjMDk2				MjMDk1				MjMDk0			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.44 RHSBGjCRkL Register (GTM) Contents (1/8)

Bit Position	Bit Name	Function
31 to 28	MjMDk7	<p>These bits select the signal of the seventh bit.</p> <p>0000: GTM TOM sub_0 ch_7 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p> <p>0001: GTM TOM sub_0 ch_15 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p> <p>0010: GTM TOM sub_1 ch_7 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p> <p>0011: GTM TOM sub_1 ch_15 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p> <p>0100: GTM TOM sub_2 ch_7 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p> <p>0101: GTM TOM sub_2 ch_15 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p> <p>0110: GTM TOM sub_3 ch_7 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p> <p>0111: GTM TOM sub_3 ch_15 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p> <p>1000: GTM ATOM sub_0 ch_7 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p> <p>1001: GTM ATOM sub_1 ch_7 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p> <p>1010: GTM ATOM sub_2 ch_7 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p> <p>1011: GTM ATOM sub_3 ch_7 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p> <p>1100: GTM ATOM sub_4 ch_7 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p> <p>1101: GTM ATOM sub_5 ch_7 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p> <p>1110: GTM ATOM sub_6 ch_7 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p> <p>1111: GTM ATOM sub_7 ch_7 is selected (RHSBGjCROK.MjMDOK7 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK7 = 11).</p>

Table 22.44 RHSBGjCRkL Register (GTM) Contents (2/8)

Bit Position	Bit Name	Function
27 to 24	MjMDk6	<p>These bits select the signal of the sixth bit.</p> <p>0000: GTM TOM sub_0 ch_6 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p> <p>0001: GTM TOM sub_0 ch_14 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p> <p>0010: GTM TOM sub_1 ch_6 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p> <p>0011: GTM TOM sub_1 ch_14 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p> <p>0100: GTM TOM sub_2 ch_6 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p> <p>0101: GTM TOM sub_2 ch_14 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p> <p>0110: GTM TOM sub_3 ch_6 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p> <p>0111: GTM TOM sub_3 ch_14 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p> <p>1000: GTM ATOM sub_0 ch_6 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p> <p>1001: GTM ATOM sub_1 ch_6 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p> <p>1010: GTM ATOM sub_2 ch_6 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p> <p>1011: GTM ATOM sub_3 ch_6 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p> <p>1100: GTM ATOM sub_4 ch_6 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p> <p>1101: GTM ATOM sub_5 ch_6 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p> <p>1110: GTM ATOM sub_6 ch_6 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p> <p>1111: GTM ATOM sub_7 ch_6 is selected (RHSBGjCROK.MjMDOK6 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK6 = 11).</p>

Table 22.44 RHSBGjCRkL Register (GTM) Contents (3/8)

Bit Position	Bit Name	Function
23 to 20	MjMDk5	<p>These bits select the signal of the fifth bit.</p> <p>0000: GTM TOM sub_0 ch_5 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p> <p>0001: GTM TOM sub_0 ch_13 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p> <p>0010: GTM TOM sub_1 ch_5 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p> <p>0011: GTM TOM sub_1 ch_13 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p> <p>0100: GTM TOM sub_2 ch_5 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p> <p>0101: GTM TOM sub_2 ch_13 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p> <p>0110: GTM TOM sub_3 ch_5 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p> <p>0111: GTM TOM sub_3 ch_13 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p> <p>1000: GTM ATOM sub_0 ch_5 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p> <p>1001: GTM ATOM sub_1 ch_5 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p> <p>1010: GTM ATOM sub_2 ch_5 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p> <p>1011: GTM ATOM sub_3 ch_5 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p> <p>1100: GTM ATOM sub_4 ch_5 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p> <p>1101: GTM ATOM sub_5 ch_5 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p> <p>1110: GTM ATOM sub_6 ch_5 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p> <p>1111: GTM ATOM sub_7 ch_5 is selected (RHSBGjCROK.MjMDOK5 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK5 = 11).</p>

Table 22.44 RHSBGjCRkL Register (GTM) Contents (4/8)

Bit Position	Bit Name	Function
19 to 16	MjMDk4	<p>These bits select the signal of the fourth bit.</p> <p>0000: GTM TOM sub_0 ch_4 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p> <p>0001: GTM TOM sub_0 ch_12 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p> <p>0010: GTM TOM sub_1 ch_4 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p> <p>0011: GTM TOM sub_1 ch_12 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p> <p>0100: GTM TOM sub_2 ch_4 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p> <p>0101: GTM TOM sub_2 ch_12 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p> <p>0110: GTM TOM sub_3 ch_4 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p> <p>0111: GTM TOM sub_3 ch_12 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p> <p>1000: GTM ATOM sub_0 ch_4 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p> <p>1001: GTM ATOM sub_1 ch_4 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p> <p>1010: GTM ATOM sub_2 ch_4 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p> <p>1011: GTM ATOM sub_3 ch_4 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p> <p>1100: GTM ATOM sub_4 ch_4 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p> <p>1101: GTM ATOM sub_5 ch_4 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p> <p>1110: GTM ATOM sub_6 ch_4 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p> <p>1111: GTM ATOM sub_7 ch_4 is selected (RHSBGjCROK.MjMDOK4 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK4 = 11).</p>

Table 22.44 RHSBGjCRkL Register (GTM) Contents (5/8)

Bit Position	Bit Name	Function
15 to 12	MjMDk3	<p>These bits select the signal of the third bit.</p> <p>0000: GTM TOM sub_0 ch_3 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p> <p>0001: GTM TOM sub_0 ch_11 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p> <p>0010: GTM TOM sub_1 ch_3 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p> <p>0011: GTM TOM sub_1 ch_11 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p> <p>0100: GTM TOM sub_2 ch_3 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p> <p>0101: GTM TOM sub_2 ch_11 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p> <p>0110: GTM TOM sub_3 ch_3 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p> <p>0111: GTM TOM sub_3 ch_11 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p> <p>1000: GTM ATOM sub_0 ch_3 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p> <p>1001: GTM ATOM sub_1 ch_3 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p> <p>1010: GTM ATOM sub_2 ch_3 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p> <p>1011: GTM ATOM sub_3 ch_3 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p> <p>1100: GTM ATOM sub_4 ch_3 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p> <p>1101: GTM ATOM sub_5 ch_3 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p> <p>1110: GTM ATOM sub_6 ch_3 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p> <p>1111: GTM ATOM sub_7 ch_3 is selected (RHSBGjCROK.MjMDOK3 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK3 = 11).</p>

Table 22.44 RHSBGjCRkL Register (GTM) Contents (6/8)

Bit Position	Bit Name	Function
11 to 8	MjMDk2	<p>These bits select the signal of the second bit.</p> <p>0000: GTM TOM sub_0 ch_2 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p> <p>0001: GTM TOM sub_0 ch_10 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p> <p>0010: GTM TOM sub_1 ch_2 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p> <p>0011: GTM TOM sub_1 ch_10 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p> <p>0100: GTM TOM sub_2 ch_2 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p> <p>0101: GTM TOM sub_2 ch_10 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p> <p>0110: GTM TOM sub_3 ch_2 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p> <p>0111: GTM TOM sub_3 ch_10 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p> <p>1000: GTM ATOM sub_0 ch_2 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p> <p>1001: GTM ATOM sub_1 ch_2 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p> <p>1010: GTM ATOM sub_2 ch_2 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p> <p>1011: GTM ATOM sub_3 ch_2 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p> <p>1100: GTM ATOM sub_4 ch_2 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p> <p>1101: GTM ATOM sub_5 ch_2 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p> <p>1110: GTM ATOM sub_6 ch_2 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p> <p>1111: GTM ATOM sub_7 ch_2 is selected (RHSBGjCROK.MjMDOK2 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK2 = 11).</p>

Table 22.44 RHSBGjCRkL Register (GTM) Contents (7/8)

Bit Position	Bit Name	Function
7 to 4	MjMDk1	<p>These bits select the signal of the first bit.</p> <p>0000: GTM TOM sub_0 ch_1 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p> <p>0001: GTM TOM sub_0 ch_9 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p> <p>0010: GTM TOM sub_1 ch_1 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p> <p>0011: GTM TOM sub_1 ch_9 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p> <p>0100: GTM TOM sub_2 ch_1 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p> <p>0101: GTM TOM sub_2 ch_9 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p> <p>0110: GTM TOM sub_3 ch_1 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p> <p>0111: GTM TOM sub_3 ch_9 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p> <p>1000: GTM ATOM sub_0 ch_1 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p> <p>1001: GTM ATOM sub_1 ch_1 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p> <p>1010: GTM ATOM sub_2 ch_1 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p> <p>1011: GTM ATOM sub_3 ch_1 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p> <p>1100: GTM ATOM sub_4 ch_1 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p> <p>1101: GTM ATOM sub_5 ch_1 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p> <p>1110: GTM ATOM sub_6 ch_1 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p> <p>1111: GTM ATOM sub_7 ch_1 is selected (RHSBGjCROK.MjMDOK1 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK1 = 11).</p>

Table 22.44 RHSBGjCRkL Register (GTM) Contents (8/8)

Bit Position	Bit Name	Function
3 to 0	MjMDk0	<p>These bits select the signal of the zeroth bit.</p> <p>0000: GTM TOM sub_0 ch_0 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p> <p>0001: GTM TOM sub_0 ch_8 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p> <p>0010: GTM TOM sub_1 ch_0 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p> <p>0011: GTM TOM sub_1 ch_8 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p> <p>0100: GTM TOM sub_2 ch_0 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p> <p>0101: GTM TOM sub_2 ch_8 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p> <p>0110: GTM TOM sub_3 ch_0 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p> <p>0111: GTM TOM sub_3 ch_8 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p> <p>1000: GTM ATOM sub_0 ch_0 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p> <p>1001: GTM ATOM sub_1 ch_0 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p> <p>1010: GTM ATOM sub_2 ch_0 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p> <p>1011: GTM ATOM sub_3 ch_0 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p> <p>1100: GTM ATOM sub_4 ch_0 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p> <p>1101: GTM ATOM sub_5 ch_0 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p> <p>1110: GTM ATOM sub_6 ch_0 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p> <p>1111: GTM ATOM sub_7 ch_0 is selected (RHSBGjCROK.MjMDOK0 = 01)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 10)./ This bit is prohibited*1 (RHSBGjCROK.MjMDOK0 = 11).</p>

Note 1. If this is selected, the low level is output.

22.6.3.7 RHSBGjCROk — Microsecond Bus Control Register Out (GTM)

RHSBGjCROk is a register that selects the signals to be output to the RHSB from signals received from the GTM and GPIO. This register supports GTM TOM, GTM ATOM and GPIO signals.

Do not change the settings of the bits in this register while any of GTM TOM, GTM ATOM or GPIO are working.

$j = 0, 1.$

$k = 0$ to 3.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MjMDOk15		MjMDOk14		MjMDOk13		MjMDOk12		MjMDOk11		MjMDOk10		MjMDOk9		MjMDOk8	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MjMDOk7		MjMDOk6		MjMDOk5		MjMDOk4		MjMDOk3		MjMDOk2		MjMDOk1		MjMDOk0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.45 RHSBGjCROk Register (GTM) Contents (1/4)

Bit Position	Bit Name	Function
31 to 30	MjMDOk15	<p>These bits select the signal of the fifteenth bit.</p> <p>00: P50_[15] is selected (k = 0)/ P51_[15] is selected (k = 1)/ P52_[15] is selected (k = 2)/ P53_[15] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected</p> <p>10: Setting prohibited.*1</p> <p>11: Setting prohibited.*1</p>
29 to 28	MjMDOk14	<p>These bits select the signal of the fourteenth bit.</p> <p>00: P50_[14] is selected (k = 0)/ P51_[14] is selected (k = 1)/ P52_[14] is selected (k = 2)/ P53_[14] is selected (k = 3).</p> <p>01 : Select output from Table 1-1 is selected</p> <p>10 : Setting prohibited.*1</p> <p>11 : Setting prohibited.*1</p>
27 to 26	MjMDOk13	<p>These bits select the signal of the thirteenth bit.</p> <p>00: P50_[13] is selected (k = 0)/ P51_[13] is selected (k = 1)/ P52_[13] is selected (k = 2)/ P53_[13] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected</p> <p>10: Setting prohibited.*1</p> <p>11: Setting prohibited.*1</p>

Table 22.45 RHBGjCROk Register (GTM) Contents (2/4)

Bit Position	Bit Name	Function
25 to 24	MjMDOK12	<p>These bits select the signal of the twelfth bit.</p> <p>00: P50_[12] is selected (k = 0)/ P51_[12] is selected (k = 1)/ P52_[12] is selected (k = 2)/ P53_[12] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected</p> <p>10: Setting prohibited.*1</p> <p>11: Setting prohibited.*1</p>
23 to 22	MjMDOK11	<p>These bits select the signal of the eleventh bit.</p> <p>00: P50_[11] is selected (k = 0)/ P51_[11] is selected (k = 1)/ P52_[11] is selected (k = 2)/ P53_[11] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected</p> <p>10: Setting prohibited.*1</p> <p>11: Setting prohibited.*1</p>
21 to 20	MjMDOK10	<p>These bits select the signal of the tenth bit.</p> <p>00: P50_[10] is selected (k = 0)/ P51_[10] is selected (k = 1)/ P52_[10] is selected (k = 2)/ P53_[10] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected</p> <p>10: Setting prohibited.*1</p> <p>11: Setting prohibited.*1</p>
19 to 18	MjMDOK9	<p>These bits select the signal of the ninth bit.</p> <p>00: P50_[9] is selected (k = 0)/ P51_[9] is selected (k = 1)/ P52_[9] is selected (k = 2)/ P53_[9] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected</p> <p>10: Setting prohibited.*1</p> <p>11: Setting prohibited.*1</p>
17 to 16	MjMDOK8	<p>These bits select the signal of the eighth bit.</p> <p>00: P50_[8] is selected (k = 0)/ P51_[8] is selected (k = 1)/ P52_[8] is selected (k = 2)/ P53_[8] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected</p> <p>10: Setting prohibited.*1</p> <p>11: Setting prohibited.*1</p>
15 to 14	MjMDOK7	<p>These bits select the signal of the seventh bit.</p> <p>00: P50_[7] is selected (k = 0)/ P51_[7] is selected (k = 1)/ P52_[7] is selected (k = 2)/ P53_[7] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected</p> <p>10: Setting prohibited.*1</p> <p>11: Setting prohibited.*1</p>

Table 22.45 RHSBGjCROk Register (GTM) Contents (3/4)

Bit Position	Bit Name	Function
13 to 12	MjMDOK6	<p>These bits select the signal of the sixth bit.</p> <p>00: P50_[6] is selected (k = 0)/ P51_[6] is selected (k = 1)/ P52_[6] is selected (k = 2)/ P53_[6] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected</p> <p>10: Setting prohibited.*1</p> <p>11: Setting prohibited.*1</p>
11 to 10	MjMDOK5	<p>These bits select the signal of the fifth bit.</p> <p>00: P50_[5] is selected (k = 0)/ P51_[5] is selected (k = 1)/ P52_[5] is selected (k = 2)/ P53_[5] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected</p> <p>10: Setting prohibited.*1</p> <p>11: Setting prohibited.*1</p>
9 to 8	MjMDOK4	<p>These bits select the signal of the fourth bit.</p> <p>00: P50_[4] is selected (k = 0)/ P51_[4] is selected (k = 1)/ P52_[4] is selected (k = 2)/ P53_[4] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected</p> <p>10: Setting prohibited.*1</p> <p>11: Setting prohibited.*1</p>
7 to 6	MjMDOK3	<p>These bits select the signal of the third bit.</p> <p>00: P50_[3] is selected (k = 0)/ P51_[3] is selected (k = 1)/ P52_[3] is selected (k = 2)/ P53_[3] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected</p> <p>10: Setting prohibited.*1</p> <p>11: Setting prohibited.*1</p>
5 to 4	MjMDOK2	<p>These bits select the signal of the second bit.</p> <p>00: P50_[2] is selected (k = 0)/ P51_[2] is selected (k = 1)/ P52_[2] is selected (k = 2)/ P53_[2] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected</p> <p>10: Setting prohibited.*1</p> <p>11: Setting prohibited.*1</p>
3 to 2	MjMDOK1	<p>These bits select the signal of the first bit.</p> <p>00: P50_[1] is selected (k = 0)/ P51_[1] is selected (k = 1)/ P52_[1] is selected (k = 2)/ P53_[1] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected</p> <p>10: Setting prohibited.*1</p> <p>11: Setting prohibited.*1</p>

Table 22.45 RHBGjCROk Register (GTM) Contents (4/4)

Bit Position	Bit Name	Function
1 to 0	MjMDOk0	<p>These bits select the signal of the zeroth bit.</p> <p>00: P50_[0] is selected (k = 0)/ P51_[0] is selected (k = 1)/ P52_[0] is selected (k = 2)/ P53_[0] is selected (k = 3).</p> <p>01: Select output from Table 1-1 is selected</p> <p>10: Setting prohibited.*1</p> <p>11: Setting prohibited.*1</p>

Note 1. If this is selected, the low level is output.

22.6.4 Operation

The number of units of RHSB is two, each of which has one XBAR. XBAR has four 16-bit sub XBARs in the 1st stage. The timer signals can be selected by setting the RHSBGjCRkH and RHSBGjCRkL registers in the 16-bit sub XBAR. The GPIO signals or the timer signals, which are selected in the 1st stage, are selected by setting of the RHSBGjCROk registers in the 2nd stage.

22.6.4.1 List of Bits that can be Selected in XBAR when Connected with RHSB and ATU-V is Used

See **Table 30.13, Block Configuration** about usable channels.

Table 22.46 List of Bits that can be Selected in XBAR when Connected with RHSB (Table1-1)

16-bit Sub XBAR		RHSBGjCRkH, RHSBGjCRkL															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
16-bit sub XBAR output	0	ATUC sub0 Ch.0	ATUC sub1 Ch.0	ATUC sub2 Ch.0	ATUC sub3 Ch.0	ATUC sub4 Ch.0	ATUC sub5 Ch.0	ATUC sub6 Ch.0	ATUC sub7 Ch.0	ATUC sub8 Ch.0	ATUC sub9 Ch.0	ATUC sub10 Ch.0	—	—	—	—	—
	1	ATUC sub0 Ch.1	ATUC sub1 Ch.1	ATUC sub2 Ch.1	ATUC sub3 Ch.1	ATUC sub4 Ch.1	ATUC sub5 Ch.1	ATUC sub6 Ch.1	ATUC sub7 Ch.1	ATUC sub8 Ch.1	ATUC sub9 Ch.1	ATUC sub10 Ch.1	—	—	—	—	—
	2	ATUC sub0 Ch.2	ATUC sub1 Ch.2	ATUC sub2 Ch.2	ATUC sub3 Ch.2	ATUC sub4 Ch.2	ATUC sub5 Ch.2	ATUC sub6 Ch.2	ATUC sub7 Ch.2	ATUC sub8 Ch.2	ATUC sub9 Ch.2	ATUC sub10 Ch.2	—	—	—	—	—
	3	ATUC sub0 Ch.3	ATUC sub1 Ch.3	ATUC sub2 Ch.3	ATUC sub3 Ch.3	ATUC sub4 Ch.3	ATUC sub5 Ch.3	ATUC sub6 Ch.3	ATUC sub7 Ch.3	ATUC sub8 Ch.3	ATUC sub9 Ch.3	ATUC sub10 Ch.3	—	—	—	—	—
	4	ATUC sub0 Ch.0	ATUC sub1 Ch.0	ATUC sub2 Ch.0	ATUC sub3 Ch.0	ATUC sub4 Ch.0	ATUC sub5 Ch.0	ATUC sub6 Ch.0	ATUC sub7 Ch.0	ATUC sub8 Ch.0	ATUC sub9 Ch.0	ATUC sub10 Ch.0	—	—	—	—	—
	5	ATUC sub0 Ch.1	ATUC sub1 Ch.1	ATUC sub2 Ch.1	ATUC sub3 Ch.1	ATUC sub4 Ch.1	ATUC sub5 Ch.1	ATUC sub6 Ch.1	ATUC sub7 Ch.1	ATUC sub8 Ch.1	ATUC sub9 Ch.1	ATUC sub10 Ch.1	—	—	—	—	—
	6	ATUC sub0 Ch.2	ATUC sub1 Ch.2	ATUC sub2 Ch.2	ATUC sub3 Ch.2	ATUC sub4 Ch.2	ATUC sub5 Ch.2	ATUC sub6 Ch.2	ATUC sub7 Ch.2	ATUC sub8 Ch.2	ATUC sub9 Ch.2	ATUC sub10 Ch.2	—	—	—	—	—
	7	ATUC sub0 Ch.3	ATUC sub1 Ch.3	ATUC sub2 Ch.3	ATUC sub3 Ch.3	ATUC sub4 Ch.3	ATUC sub5 Ch.3	ATUC sub6 Ch.3	ATUC sub7 Ch.3	ATUC sub8 Ch.3	ATUC sub9 Ch.3	ATUC sub10 Ch.3	—	—	—	—	—
	8	ATUC sub0 Ch.0	ATUC sub1 Ch.0	ATUC sub2 Ch.0	ATUC sub3 Ch.0	ATUC sub4 Ch.0	ATUC sub5 Ch.0	ATUC sub6 Ch.0	ATUC sub7 Ch.0	ATUC sub8 Ch.0	ATUC sub9 Ch.0	ATUC sub10 Ch.0	—	—	—	—	—
	9	ATUC sub0 Ch.1	ATUC sub1 Ch.1	ATUC sub2 Ch.1	ATUC sub3 Ch.1	ATUC sub4 Ch.1	ATUC sub5 Ch.1	ATUC sub6 Ch.1	ATUC sub7 Ch.1	ATUC sub8 Ch.1	ATUC sub9 Ch.1	ATUC sub10 Ch.1	—	—	—	—	—
	10	ATUC sub0 Ch.2	ATUC sub1 Ch.2	ATUC sub2 Ch.2	ATUC sub3 Ch.2	ATUC sub4 Ch.2	ATUC sub5 Ch.2	ATUC sub6 Ch.2	ATUC sub7 Ch.2	ATUC sub8 Ch.2	ATUC sub9 Ch.2	ATUC sub10 Ch.2	—	—	—	—	—
	11	ATUC sub0 Ch.3	ATUC sub1 Ch.3	ATUC sub2 Ch.3	ATUC sub3 Ch.3	ATUC sub4 Ch.3	ATUC sub5 Ch.3	ATUC sub6 Ch.3	ATUC sub7 Ch.3	ATUC sub8 Ch.3	ATUC sub9 Ch.3	ATUC sub10 Ch.3	—	—	—	—	—
	12	ATUC sub0 Ch.0	ATUC sub1 Ch.0	ATUC sub2 Ch.0	ATUC sub3 Ch.0	ATUC sub4 Ch.0	ATUC sub5 Ch.0	ATUC sub6 Ch.0	ATUC sub7 Ch.0	ATUC sub8 Ch.0	ATUC sub9 Ch.0	ATUC sub10 Ch.0	—	—	—	—	—
	13	ATUC sub0 Ch.1	ATUC sub1 Ch.1	ATUC sub2 Ch.1	ATUC sub3 Ch.1	ATUC sub4 Ch.1	ATUC sub5 Ch.1	ATUC sub6 Ch.1	ATUC sub7 Ch.1	ATUC sub8 Ch.1	ATUC sub9 Ch.1	ATUC sub10 Ch.1	—	—	—	—	—
	14	ATUC sub0 Ch.2	ATUC sub1 Ch.2	ATUC sub2 Ch.2	ATUC sub3 Ch.2	ATUC sub4 Ch.2	ATUC sub5 Ch.2	ATUC sub6 Ch.2	ATUC sub7 Ch.2	ATUC sub8 Ch.2	ATUC sub9 Ch.2	ATUC sub10 Ch.2	—	—	—	—	—
	15	ATUC sub0 Ch.3	ATUC sub1 Ch.3	ATUC sub2 Ch.3	ATUC sub3 Ch.3	ATUC sub4 Ch.3	ATUC sub5 Ch.3	ATUC sub6 Ch.3	ATUC sub7 Ch.3	ATUC sub8 Ch.3	ATUC sub9 Ch.3	ATUC sub10 Ch.3	—	—	—	—	—

Table 22.47 List of Bits that can be Selected in XBAR when Connected with RHSB (Table1-2)

16-bit Sub XBAR		RHSBGJCRkH, RHSBGJCRkL															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
16-bit sub XBAR output	0	ATUD sub0 Ch.0	ATUD sub1 Ch.0	ATUD sub2 Ch.0	ATUD sub3 Ch.0	ATUD sub4 Ch.0	ATUD sub5 Ch.0	ATUD sub6 Ch.0	ATUD sub7 Ch.0	ATUD sub8 Ch.0	—	—	—	—	—	—	—
	1	ATUD sub0 Ch.1	ATUD sub1 Ch.1	ATUD sub2 Ch.1	ATUD sub3 Ch.1	ATUD sub4 Ch.1	ATUD sub5 Ch.1	ATUD sub6 Ch.1	ATUD sub7 Ch.1	ATUD sub8 Ch.1	—	—	—	—	—	—	—
	2	ATUD sub0 Ch.2	ATUD sub1 Ch.2	ATUD sub2 Ch.2	ATUD sub3 Ch.2	ATUD sub4 Ch.2	ATUD sub5 Ch.2	ATUD sub6 Ch.2	ATUD sub7 Ch.2	ATUD sub8 Ch.2	—	—	—	—	—	—	—
	3	ATUD sub0 Ch.3	ATUD sub1 Ch.3	ATUD sub2 Ch.3	ATUD sub3 Ch.3	ATUD sub4 Ch.3	ATUD sub5 Ch.3	ATUD sub6 Ch.3	ATUD sub7 Ch.3	ATUD sub8 Ch.3	—	—	—	—	—	—	—
	4	ATUD sub0 Ch.0	ATUD sub1 Ch.0	ATUD sub2 Ch.0	ATUD sub3 Ch.0	ATUD sub4 Ch.0	ATUD sub5 Ch.0	ATUD sub6 Ch.0	ATUD sub7 Ch.0	ATUD sub8 Ch.0	—	—	—	—	—	—	—
	5	ATUD sub0 Ch.1	ATUD sub1 Ch.1	ATUD sub2 Ch.1	ATUD sub3 Ch.1	ATUD sub4 Ch.1	ATUD sub5 Ch.1	ATUD sub6 Ch.1	ATUD sub7 Ch.1	ATUD sub8 Ch.1	—	—	—	—	—	—	—
	6	ATUD sub0 Ch.2	ATUD sub1 Ch.2	ATUD sub2 Ch.2	ATUD sub3 Ch.2	ATUD sub4 Ch.2	ATUD sub5 Ch.2	ATUD sub6 Ch.2	ATUD sub7 Ch.2	ATUD sub8 Ch.2	—	—	—	—	—	—	—
	7	ATUD sub0 Ch.3	ATUD sub1 Ch.3	ATUD sub2 Ch.3	ATUD sub3 Ch.3	ATUD sub4 Ch.3	ATUD sub5 Ch.3	ATUD sub6 Ch.3	ATUD sub7 Ch.3	ATUD sub8 Ch.3	—	—	—	—	—	—	—
	8	ATUD sub0 Ch.0	ATUD sub1 Ch.0	ATUD sub2 Ch.0	ATUD sub3 Ch.0	ATUD sub4 Ch.0	ATUD sub5 Ch.0	ATUD sub6 Ch.0	ATUD sub7 Ch.0	ATUD sub8 Ch.0	—	—	—	—	—	—	—
	9	ATUD sub0 Ch.1	ATUD sub1 Ch.1	ATUD sub2 Ch.1	ATUD sub3 Ch.1	ATUD sub4 Ch.1	ATUD sub5 Ch.1	ATUD sub6 Ch.1	ATUD sub7 Ch.1	ATUD sub8 Ch.1	—	—	—	—	—	—	—
	10	ATUD sub0 Ch.2	ATUD sub1 Ch.2	ATUD sub2 Ch.2	ATUD sub3 Ch.2	ATUD sub4 Ch.2	ATUD sub5 Ch.2	ATUD sub6 Ch.2	ATUD sub7 Ch.2	ATUD sub8 Ch.2	—	—	—	—	—	—	—
	11	ATUD sub0 Ch.3	ATUD sub1 Ch.3	ATUD sub2 Ch.3	ATUD sub3 Ch.3	ATUD sub4 Ch.3	ATUD sub5 Ch.3	ATUD sub6 Ch.3	ATUD sub7 Ch.3	ATUD sub8 Ch.3	—	—	—	—	—	—	—
	12	ATUD sub0 Ch.0	ATUD sub1 Ch.0	ATUD sub2 Ch.0	ATUD sub3 Ch.0	ATUD sub4 Ch.0	ATUD sub5 Ch.0	ATUD sub6 Ch.0	ATUD sub7 Ch.0	ATUD sub8 Ch.0	—	—	—	—	—	—	—
	13	ATUD sub0 Ch.1	ATUD sub1 Ch.1	ATUD sub2 Ch.1	ATUD sub3 Ch.1	ATUD sub4 Ch.1	ATUD sub5 Ch.1	ATUD sub6 Ch.1	ATUD sub7 Ch.1	ATUD sub8 Ch.1	—	—	—	—	—	—	—
	14	ATUD sub0 Ch.2	ATUD sub1 Ch.2	ATUD sub2 Ch.2	ATUD sub3 Ch.2	ATUD sub4 Ch.2	ATUD sub5 Ch.2	ATUD sub6 Ch.2	ATUD sub7 Ch.2	ATUD sub8 Ch.2	—	—	—	—	—	—	—
	15	ATUD sub0 Ch.3	ATUD sub1 Ch.3	ATUD sub2 Ch.3	ATUD sub3 Ch.3	ATUD sub4 Ch.3	ATUD sub5 Ch.3	ATUD sub6 Ch.3	ATUD sub7 Ch.3	ATUD sub8 Ch.3	—	—	—	—	—	—	—

Table 22.48 List of Bits that can be Selected in XBAR when Connected with RHSB (Table1-3)

16-bit Sub XBAR		RHSBGJCRkH, RHSBGJCRkL															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
16-bit sub XBAR output	0	ATUE sub0 Ch.0	ATUE sub1 Ch.0	ATUE sub2 Ch.0	ATUE sub3 Ch.0	ATUE sub4 Ch.0	ATUE sub5 Ch.0	ATUE sub6 Ch.0	ATUE sub7 Ch.0	ATUE sub8 Ch.0	—	—	—	—	—	—	—
	1	ATUE sub0 Ch.1	ATUE sub1 Ch.1	ATUE sub2 Ch.1	ATUE sub3 Ch.1	ATUE sub4 Ch.1	ATUE sub5 Ch.1	ATUE sub6 Ch.1	ATUE sub7 Ch.1	ATUE sub8 Ch.1	—	—	—	—	—	—	—
	2	ATUE sub0 Ch.2	ATUE sub1 Ch.2	ATUE sub2 Ch.2	ATUE sub3 Ch.2	ATUE sub4 Ch.2	ATUE sub5 Ch.2	ATUE sub6 Ch.2	ATUE sub7 Ch.2	ATUE sub8 Ch.2	—	—	—	—	—	—	—
	3	ATUE sub0 Ch.3	ATUE sub1 Ch.3	ATUE sub2 Ch.3	ATUE sub3 Ch.3	ATUE sub4 Ch.3	ATUE sub5 Ch.3	ATUE sub6 Ch.3	ATUE sub7 Ch.3	ATUE sub8 Ch.3	—	—	—	—	—	—	—
	4	ATUE sub0 Ch.0	ATUE sub1 Ch.0	ATUE sub2 Ch.0	ATUE sub3 Ch.0	ATUE sub4 Ch.0	ATUE sub5 Ch.0	ATUE sub6 Ch.0	ATUE sub7 Ch.0	ATUE sub8 Ch.0	—	—	—	—	—	—	—
	5	ATUE sub0 Ch.1	ATUE sub1 Ch.1	ATUE sub2 Ch.1	ATUE sub3 Ch.1	ATUE sub4 Ch.1	ATUE sub5 Ch.1	ATUE sub6 Ch.1	ATUE sub7 Ch.1	ATUE sub8 Ch.1	—	—	—	—	—	—	—
	6	ATUE sub0 Ch.2	ATUE sub1 Ch.2	ATUE sub2 Ch.2	ATUE sub3 Ch.2	ATUE sub4 Ch.2	ATUE sub5 Ch.2	ATUE sub6 Ch.2	ATUE sub7 Ch.2	ATUE sub8 Ch.2	—	—	—	—	—	—	—
	7	ATUE sub0 Ch.3	ATUE sub1 Ch.3	ATUE sub2 Ch.3	ATUE sub3 Ch.3	ATUE sub4 Ch.3	ATUE sub5 Ch.3	ATUE sub6 Ch.3	ATUE sub7 Ch.3	ATUE sub8 Ch.3	—	—	—	—	—	—	—
	8	ATUE sub0 Ch.0	ATUE sub1 Ch.0	ATUE sub2 Ch.0	ATUE sub3 Ch.0	ATUE sub4 Ch.0	ATUE sub5 Ch.0	ATUE sub6 Ch.0	ATUE sub7 Ch.0	ATUE sub8 Ch.0	—	—	—	—	—	—	—
	9	ATUE sub0 Ch.1	ATUE sub1 Ch.1	ATUE sub2 Ch.1	ATUE sub3 Ch.1	ATUE sub4 Ch.1	ATUE sub5 Ch.1	ATUE sub6 Ch.1	ATUE sub7 Ch.1	ATUE sub8 Ch.1	—	—	—	—	—	—	—
	10	ATUE sub0 Ch.2	ATUE sub1 Ch.2	ATUE sub2 Ch.2	ATUE sub3 Ch.2	ATUE sub4 Ch.2	ATUE sub5 Ch.2	ATUE sub6 Ch.2	ATUE sub7 Ch.2	ATUE sub8 Ch.2	—	—	—	—	—	—	—
	11	ATUE sub0 Ch.3	ATUE sub1 Ch.3	ATUE sub2 Ch.3	ATUE sub3 Ch.3	ATUE sub4 Ch.3	ATUE sub5 Ch.3	ATUE sub6 Ch.3	ATUE sub7 Ch.3	ATUE sub8 Ch.3	—	—	—	—	—	—	—
	12	ATUE sub0 Ch.0	ATUE sub1 Ch.0	ATUE sub2 Ch.0	ATUE sub3 Ch.0	ATUE sub4 Ch.0	ATUE sub5 Ch.0	ATUE sub6 Ch.0	ATUE sub7 Ch.0	ATUE sub8 Ch.0	—	—	—	—	—	—	—
	13	ATUE sub0 Ch.1	ATUE sub1 Ch.1	ATUE sub2 Ch.1	ATUE sub3 Ch.1	ATUE sub4 Ch.1	ATUE sub5 Ch.1	ATUE sub6 Ch.1	ATUE sub7 Ch.1	ATUE sub8 Ch.1	—	—	—	—	—	—	—
	14	ATUE sub0 Ch.2	ATUE sub1 Ch.2	ATUE sub2 Ch.2	ATUE sub3 Ch.2	ATUE sub4 Ch.2	ATUE sub5 Ch.2	ATUE sub6 Ch.2	ATUE sub7 Ch.2	ATUE sub8 Ch.2	—	—	—	—	—	—	—
	15	ATUE sub0 Ch.3	ATUE sub1 Ch.3	ATUE sub2 Ch.3	ATUE sub3 Ch.3	ATUE sub4 Ch.3	ATUE sub5 Ch.3	ATUE sub6 Ch.3	ATUE sub7 Ch.3	ATUE sub8 Ch.3	—	—	—	—	—	—	—

Table 22.49 List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1) (1/4)

Output	List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1)			
	RHSBGjCROK(ATU-V)			
RHSB	00	01	10	11
0	P50_[0]	Table 1-1	Table 1-2	Table 1-3
1	P50_[1]	Table 1-1	Table 1-2	Table 1-3
2	P50_[2]	Table 1-1	Table 1-2	Table 1-3
3	P50_[3]	Table 1-1	Table 1-2	Table 1-3
4	P50_[4]	Table 1-1	Table 1-2	Table 1-3
5	P50_[5]	Table 1-1	Table 1-2	Table 1-3
6	P50_[6]	Table 1-1	Table 1-2	Table 1-3
7	P50_[7]	Table 1-1	Table 1-2	Table 1-3
8	P50_[8]	Table 1-1	Table 1-2	Table 1-3
9	P50_[9]	Table 1-1	Table 1-2	Table 1-3
10	P50_[10]	Table 1-1	Table 1-2	Table 1-3
11	P50_[11]	Table 1-1	Table 1-2	Table 1-3
12	P50_[12]	Table 1-1	Table 1-2	Table 1-3
13	P50_[13]	Table 1-1	Table 1-2	Table 1-3
14	P50_[14]	Table 1-1	Table 1-2	Table 1-3
15	P50_[15]	Table 1-1	Table 1-2	Table 1-3

Table 22.49 List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1) (2/4)

Output	List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1)			
	RHSBGjCROK(ATU-V)			
RHSB	00	01	10	11
16	P51_[0]	Table 1-1	Table 1-2	Table 1-3
17	P51_[1]	Table 1-1	Table 1-2	Table 1-3
18	P51_[2]	Table 1-1	Table 1-2	Table 1-3
19	P51_[3]	Table 1-1	Table 1-2	Table 1-3
20	P51_[4]	Table 1-1	Table 1-2	Table 1-3
21	P51_[5]	Table 1-1	Table 1-2	Table 1-3
22	P51_[6]	Table 1-1	Table 1-2	Table 1-3
23	P51_[7]	Table 1-1	Table 1-2	Table 1-3
24	P51_[8]	Table 1-1	Table 1-2	Table 1-3
25	P51_[9]	Table 1-1	Table 1-2	Table 1-3
26	P51_[10]	Table 1-1	Table 1-2	Table 1-3
27	P51_[11]	Table 1-1	Table 1-2	Table 1-3
28	P51_[12]	Table 1-1	Table 1-2	Table 1-3
29	P51_[13]	Table 1-1	Table 1-2	Table 1-3
30	P51_[14]	Table 1-1	Table 1-2	Table 1-3
31	P51_[15]	Table 1-1	Table 1-2	Table 1-3

Table 22.49 List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1) (3/4)

Output	List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1)			
	RHSBGjCROk(ATU-V)			
RHSB	00	01	10	11
32	P52_[0]	Table 1-1	Table 1-2	Table 1-3
33	P52_[1]	Table 1-1	Table 1-2	Table 1-3
34	P52_[2]	Table 1-1	Table 1-2	Table 1-3
35	P52_[3]	Table 1-1	Table 1-2	Table 1-3
36	P52_[4]	Table 1-1	Table 1-2	Table 1-3
37	P52_[5]	Table 1-1	Table 1-2	Table 1-3
38	P52_[6]	Table 1-1	Table 1-2	Table 1-3
39	P52_[7]	Table 1-1	Table 1-2	Table 1-3
40	P52_[8]	Table 1-1	Table 1-2	Table 1-3
41	P52_[9]	Table 1-1	Table 1-2	Table 1-3
42	P52_[10]	Table 1-1	Table 1-2	Table 1-3
43	P52_[11]	Table 1-1	Table 1-2	Table 1-3
44	P52_[12]	Table 1-1	Table 1-2	Table 1-3
45	P52_[13]	Table 1-1	Table 1-2	Table 1-3
46	P52_[14]	Table 1-1	Table 1-2	Table 1-3
47	P52_[15]	Table 1-1	Table 1-2	Table 1-3

Table 22.49 List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1) (4/4)

Output	List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1)			
	RHSBGjCROk(ATU-V)			
RHSB	00	01	10	11
48	P53_[0]	Table 1-1	Table 1-2	Table 1-3
49	P53_[1]	Table 1-1	Table 1-2	Table 1-3
50	P53_[2]	Table 1-1	Table 1-2	Table 1-3
51	P53_[3]	Table 1-1	Table 1-2	Table 1-3
52	P53_[4]	Table 1-1	Table 1-2	Table 1-3
53	P53_[5]	Table 1-1	Table 1-2	Table 1-3
54	P53_[6]	Table 1-1	Table 1-2	Table 1-3
55	P53_[7]	Table 1-1	Table 1-2	Table 1-3
56	P53_[8]	Table 1-1	Table 1-2	Table 1-3
57	P53_[9]	Table 1-1	Table 1-2	Table 1-3
58	P53_[10]	Table 1-1	Table 1-2	Table 1-3
59	P53_[11]	Table 1-1	Table 1-2	Table 1-3
60	P53_[12]	Table 1-1	Table 1-2	Table 1-3
61	P53_[13]	Table 1-1	Table 1-2	Table 1-3
62	P53_[14]	Table 1-1	Table 1-2	Table 1-3
63	P53_[15]	Table 1-1	Table 1-2	Table 1-3

The following figure shows the logic and the register which select 0-bit of the RHSB.

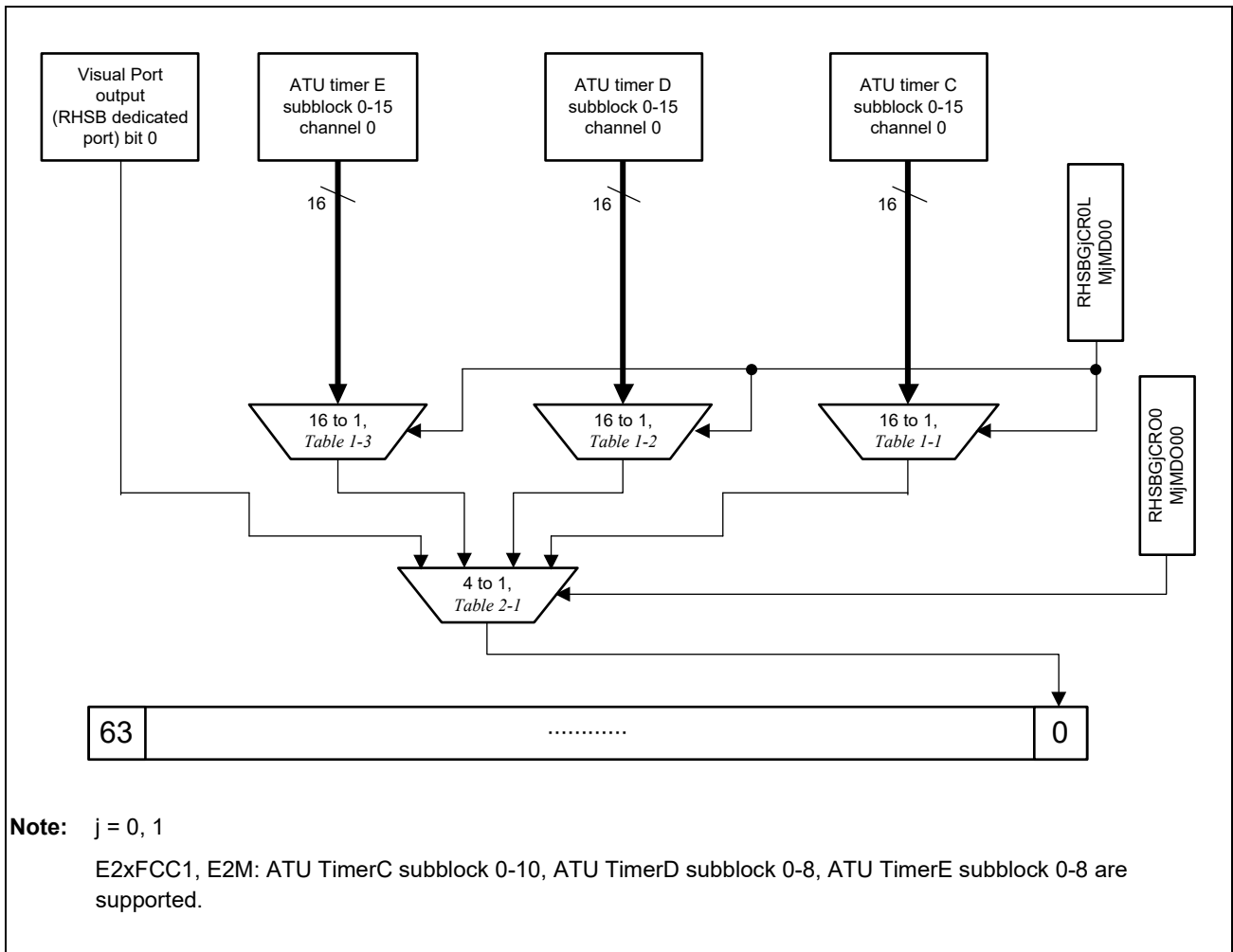


Figure 22.38 RHSB Downstream Frame 0-bit Selecting Logic when ATU-V is Used

22.6.4.2 List of Bits that can be Selected in XBAR when Connected with RHSB and GTM is Used

Table 22.50 List of Bits that can be Selected in XBAR when Connected with RHSB (Table 1-1)

16-bit Sub XBAR		RHSBGJCRkH, RHSBGJCRkL															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
16-bit Sub XBAR output	0	GTM TOM 0_0	GTM TOM 0_8	GTM TOM 1_0	GTM TOM 1_8	GTM TOM 2_0	GTM TOM 2_8	GTM TOM 3_0	GTM TOM 3_8	GTM ATOM 0_0	GTM ATOM 1_0	GTM ATOM 2_0	GTM ATOM 3_0	GTM ATOM 4_0	GTM ATOM 5_0	GTM ATOM 6_0	GTM ATOM 7_0
	1	GTM TOM 0_1	GTM TOM 0_9	GTM TOM 1_1	GTM TOM 1_9	GTM TOM 2_1	GTM TOM 2_9	GTM TOM 3_1	GTM TOM 3_9	GTM ATOM 0_1	GTM ATOM 1_1	GTM ATOM 2_1	GTM ATOM 3_1	GTM ATOM 4_1	GTM ATOM 5_1	GTM ATOM 6_1	GTM ATOM 7_1
	2	GTM TOM 0_2	GTM TOM 0_10	GTM TOM 1_2	GTM TOM 1_10	GTM TOM 2_2	GTM TOM 2_10	GTM TOM 3_2	GTM TOM 3_10	GTM ATOM 0_2	GTM ATOM 1_2	GTM ATOM 2_2	GTM ATOM 3_2	GTM ATOM 4_2	GTM ATOM 5_2	GTM ATOM 6_2	GTM ATOM 7_2
	3	GTM TOM 0_3	GTM TOM 0_11	GTM TOM 1_3	GTM TOM 1_11	GTM TOM 2_3	GTM TOM 2_11	GTM TOM 3_3	GTM TOM 3_11	GTM ATOM 0_3	GTM ATOM 1_3	GTM ATOM 2_3	GTM ATOM 3_3	GTM ATOM 4_3	GTM ATOM 5_3	GTM ATOM 6_3	GTM ATOM 7_3
	4	GTM TOM 0_4	GTM TOM 0_12	GTM TOM 1_4	GTM TOM 1_12	GTM TOM 2_4	GTM TOM 2_12	GTM TOM 3_4	GTM TOM 3_12	GTM ATOM 0_4	GTM ATOM 1_4	GTM ATOM 2_4	GTM ATOM 3_4	GTM ATOM 4_4	GTM ATOM 5_4	GTM ATOM 6_4	GTM ATOM 7_4
	5	GTM TOM 0_5	GTM TOM 0_13	GTM TOM 1_5	GTM TOM 1_13	GTM TOM 2_5	GTM TOM 2_13	GTM TOM 3_5	GTM TOM 3_13	GTM ATOM 0_5	GTM ATOM 1_5	GTM ATOM 2_5	GTM ATOM 3_5	GTM ATOM 4_5	GTM ATOM 5_5	GTM ATOM 6_5	GTM ATOM 7_5
	6	GTM TOM 0_6	GTM TOM 0_14	GTM TOM 1_6	GTM TOM 1_14	GTM TOM 2_6	GTM TOM 2_14	GTM TOM 3_6	GTM TOM 3_14	GTM ATOM 0_6	GTM ATOM 1_6	GTM ATOM 2_6	GTM ATOM 3_6	GTM ATOM 4_6	GTM ATOM 5_6	GTM ATOM 6_6	GTM ATOM 7_6
	7	GTM TOM 0_7	GTM TOM 0_15	GTM TOM 1_7	GTM TOM 1_15	GTM TOM 2_7	GTM TOM 2_15	GTM TOM 3_7	GTM TOM 3_15	GTM ATOM 0_7	GTM ATOM 1_7	GTM ATOM 2_7	GTM ATOM 3_7	GTM ATOM 4_7	GTM ATOM 5_7	GTM ATOM 6_7	GTM ATOM 7_7
	8	GTM TOM 0_0	GTM TOM 0_8	GTM TOM 1_0	GTM TOM 1_8	GTM TOM 2_0	GTM TOM 2_8	GTM TOM 3_0	GTM TOM 3_8	GTM ATOM 0_0	GTM ATOM 1_0	GTM ATOM 2_0	GTM ATOM 3_0	GTM ATOM 4_0	GTM ATOM 5_0	GTM ATOM 6_0	GTM ATOM 7_0
	9	GTM TOM 0_1	GTM TOM 0_9	GTM TOM 1_1	GTM TOM 1_9	GTM TOM 2_1	GTM TOM 2_9	GTM TOM 3_1	GTM TOM 3_9	GTM ATOM 0_1	GTM ATOM 1_1	GTM ATOM 2_1	GTM ATOM 3_1	GTM ATOM 4_1	GTM ATOM 5_1	GTM ATOM 6_1	GTM ATOM 7_1
	10	GTM TOM 0_2	GTM TOM 0_10	GTM TOM 1_2	GTM TOM 1_10	GTM TOM 2_2	GTM TOM 2_10	GTM TOM 3_2	GTM TOM 3_10	GTM ATOM 0_2	GTM ATOM 1_2	GTM ATOM 2_2	GTM ATOM 3_2	GTM ATOM 4_2	GTM ATOM 5_2	GTM ATOM 6_2	GTM ATOM 7_2
	11	GTM TOM 0_3	GTM TOM 0_11	GTM TOM 1_3	GTM TOM 1_11	GTM TOM 2_3	GTM TOM 2_11	GTM TOM 3_3	GTM TOM 3_11	GTM ATOM 0_3	GTM ATOM 1_3	GTM ATOM 2_3	GTM ATOM 3_3	GTM ATOM 4_3	GTM ATOM 5_3	GTM ATOM 6_3	GTM ATOM 7_3
	12	GTM TOM 0_4	GTM TOM 0_12	GTM TOM 1_4	GTM TOM 1_12	GTM TOM 2_4	GTM TOM 2_12	GTM TOM 3_4	GTM TOM 3_12	GTM ATOM 0_4	GTM ATOM 1_4	GTM ATOM 2_4	GTM ATOM 3_4	GTM ATOM 4_4	GTM ATOM 5_4	GTM ATOM 6_4	GTM ATOM 7_4
	13	GTM TOM 0_5	GTM TOM 0_13	GTM TOM 1_5	GTM TOM 1_13	GTM TOM 2_5	GTM TOM 2_13	GTM TOM 3_5	GTM TOM 3_13	GTM ATOM 0_5	GTM ATOM 1_5	GTM ATOM 2_5	GTM ATOM 3_5	GTM ATOM 4_5	GTM ATOM 5_5	GTM ATOM 6_5	GTM ATOM 7_5
	14	GTM TOM 0_6	GTM TOM 0_14	GTM TOM 1_6	GTM TOM 1_14	GTM TOM 2_6	GTM TOM 2_14	GTM TOM 3_6	GTM TOM 3_14	GTM ATOM 0_6	GTM ATOM 1_6	GTM ATOM 2_6	GTM ATOM 3_6	GTM ATOM 4_6	GTM ATOM 5_6	GTM ATOM 6_6	GTM ATOM 7_6
	15	GTM TOM 0_7	GTM TOM 0_15	GTM TOM 1_7	GTM TOM 1_15	GTM TOM 2_7	GTM TOM 2_15	GTM TOM 3_7	GTM TOM 3_15	GTM ATOM 0_7	GTM ATOM 1_7	GTM ATOM 2_7	GTM ATOM 3_7	GTM ATOM 4_7	GTM ATOM 5_7	GTM ATOM 6_7	GTM ATOM 7_7

Note: The GTMTOM0_0 signal indicates the GTMT[0]O0 signal in the GTM section.
 The GTMATOM0_0 signal indicates the GTMAT[0]O0 signal in the GTM section.

Table 22.51 List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1) (1/4)

Output	List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1)			
	RHSBGjCROk(GTM)			
RHSB	00	01	10	11
0	P50_[0]	Table 1-1	Reserved	Reserved
1	P50_[1]	Table 1-1	Reserved	Reserved
2	P50_[2]	Table 1-1	Reserved	Reserved
3	P50_[3]	Table 1-1	Reserved	Reserved
4	P50_[4]	Table 1-1	Reserved	Reserved
5	P50_[5]	Table 1-1	Reserved	Reserved
6	P50_[6]	Table 1-1	Reserved	Reserved
7	P50_[7]	Table 1-1	Reserved	Reserved
8	P50_[8]	Table 1-1	Reserved	Reserved
9	P50_[9]	Table 1-1	Reserved	Reserved
10	P50_[10]	Table 1-1	Reserved	Reserved
11	P50_[11]	Table 1-1	Reserved	Reserved
12	P50_[12]	Table 1-1	Reserved	Reserved
13	P50_[13]	Table 1-1	Reserved	Reserved
14	P50_[14]	Table 1-1	Reserved	Reserved
15	P50_[15]	Table 1-1	Reserved	Reserved

Table 22.51 List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1) (2/4)

Output	List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1)			
	RHSBGjCROk(GTM)			
RHSB	00	01	10	11
16	P51_[0]	Table 1-1	Reserved	Reserved
17	P51_[1]	Table 1-1	Reserved	Reserved
18	P51_[2]	Table 1-1	Reserved	Reserved
19	P51_[3]	Table 1-1	Reserved	Reserved
20	P51_[4]	Table 1-1	Reserved	Reserved
21	P51_[5]	Table 1-1	Reserved	Reserved
22	P51_[6]	Table 1-1	Reserved	Reserved
23	P51_[7]	Table 1-1	Reserved	Reserved
24	P51_[8]	Table 1-1	Reserved	Reserved
25	P51_[9]	Table 1-1	Reserved	Reserved
26	P51_[10]	Table 1-1	Reserved	Reserved
27	P51_[11]	Table 1-1	Reserved	Reserved
28	P51_[12]	Table 1-1	Reserved	Reserved
29	P51_[13]	Table 1-1	Reserved	Reserved
30	P51_[14]	Table 1-1	Reserved	Reserved
31	P51_[15]	Table 1-1	Reserved	Reserved

Table 22.51 List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1) (3/4)

Output	List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1)			
	RHSBGjCROk(GTM)			
RHSB	00	01	10	11
32	P52_[0]	Table 1-1	Reserved	Reserved
33	P52_[1]	Table 1-1	Reserved	Reserved
34	P52_[2]	Table 1-1	Reserved	Reserved
35	P52_[3]	Table 1-1	Reserved	Reserved
36	P52_[4]	Table 1-1	Reserved	Reserved
37	P52_[5]	Table 1-1	Reserved	Reserved
38	P52_[6]	Table 1-1	Reserved	Reserved
39	P52_[7]	Table 1-1	Reserved	Reserved
40	P52_[8]	Table 1-1	Reserved	Reserved
41	P52_[9]	Table 1-1	Reserved	Reserved
42	P52_[10]	Table 1-1	Reserved	Reserved
43	P52_[11]	Table 1-1	Reserved	Reserved
44	P52_[12]	Table 1-1	Reserved	Reserved
45	P52_[13]	Table 1-1	Reserved	Reserved
46	P52_[14]	Table 1-1	Reserved	Reserved
47	P52_[15]	Table 1-1	Reserved	Reserved

Table 22.51 List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1) (4/4)

Output	List of Bits that can be Selected in XBAR when Connected with RHSB (Table 2-1)			
	RHSBGjCROk(GTM)			
RHSB	00	01	10	11
48	P53_[0]	Table 1-1	Reserved	Reserved
49	P53_[1]	Table 1-1	Reserved	Reserved
50	P53_[2]	Table 1-1	Reserved	Reserved
51	P53_[3]	Table 1-1	Reserved	Reserved
52	P53_[4]	Table 1-1	Reserved	Reserved
53	P53_[5]	Table 1-1	Reserved	Reserved
54	P53_[6]	Table 1-1	Reserved	Reserved
55	P53_[7]	Table 1-1	Reserved	Reserved
56	P53_[8]	Table 1-1	Reserved	Reserved
57	P53_[9]	Table 1-1	Reserved	Reserved
58	P53_[10]	Table 1-1	Reserved	Reserved
59	P53_[11]	Table 1-1	Reserved	Reserved
60	P53_[12]	Table 1-1	Reserved	Reserved
61	P53_[13]	Table 1-1	Reserved	Reserved
62	P53_[14]	Table 1-1	Reserved	Reserved
63	P53_[15]	Table 1-1	Reserved	Reserved

The following figure shows the logic and the register that selects bit 0 of the RHSB.

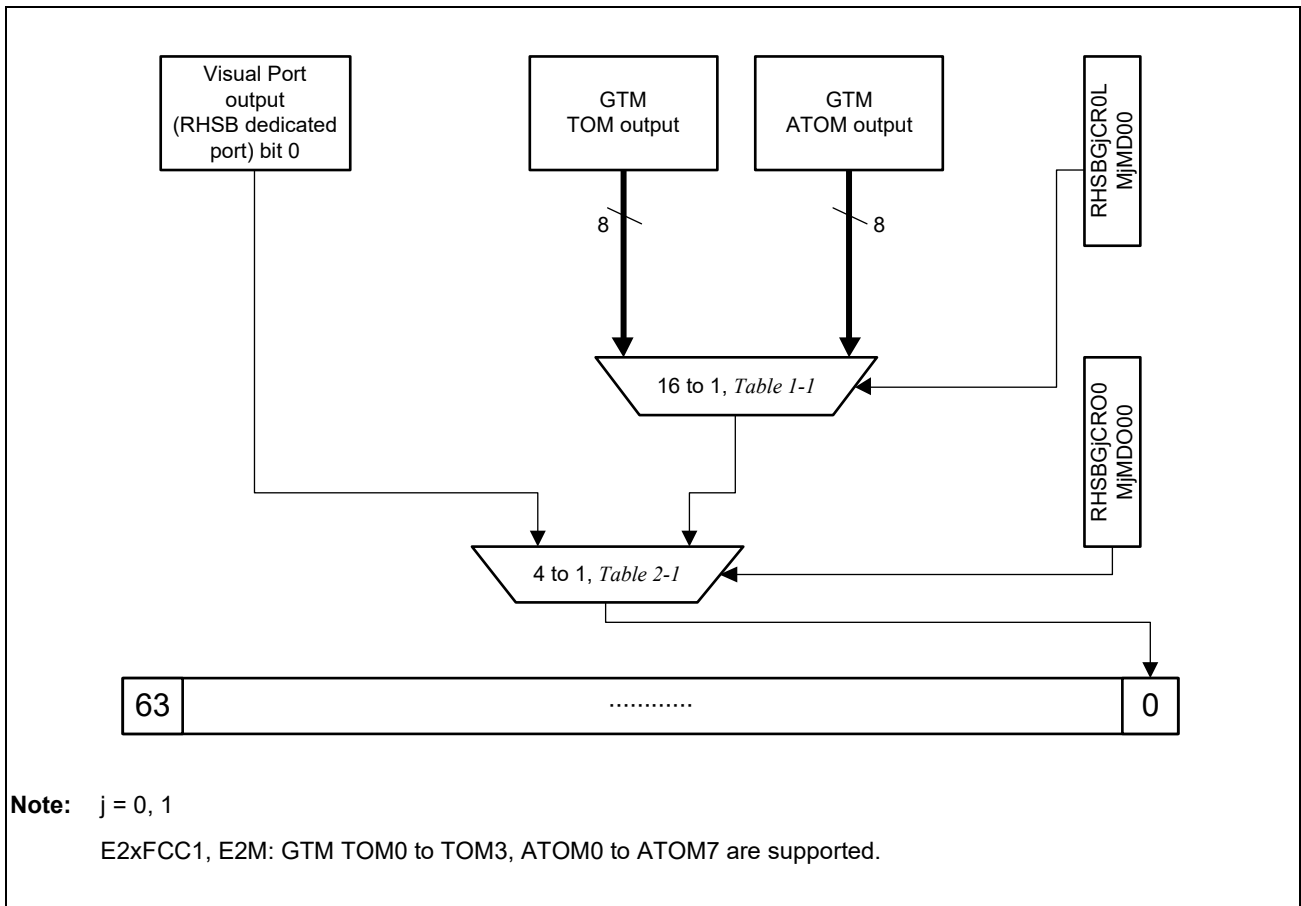


Figure 22.39 RHSB Downstream Frame 0-bit Selecting Logic when GTM is Used

Section 23 Single Edge Nibble Transmission (RSENT)

This section describes the Renesas single edge nibble transmission (RSENT) module.

23.1 Features of RSENT

23.1.1 Number of Channels

This microcontroller has the following number of RSENT channels.

Table 23.1 Number of Channels

Product Name	RH850/E2x-FCC1	
	373 pins	292 pins
Number of Channels	17	17
Unit Name	RSENTn (n = 0 to 16)	

Table 23.2 Number of Channels

Product Name	RH850/E2M	
	373 pins	292 pins
Number of Channels	17	17
Name	RSENTn (n = 0 to 16)	

Table 23.3 Index

Index	Description
n	This section identifies each RSENT channel by "n" ("n" is channel numbers.). For example, the RSENT timestamp register of RSENTn is described as RSENTnTSPC.

23.1.2 Register Base Addresses

The RSENT register base addresses are given as offsets from the base address <RSENTn_base>.

The following table shows the base addresses <RSENTn_base> for each RSENT.

Table 23.4 Register Base Addresses

Base Address Name	Base Address	Bus Group
<RSENT0_base>	FFCF 0000 _H	Peripheral Group 6
<RSENT1_base>	FFCF 0080 _H	Peripheral Group 6
<RSENT2_base>	FFCF 0100 _H	Peripheral Group 6
<RSENT3_base>	FFCF 0180 _H	Peripheral Group 6
<RSENT4_base>	FFCF 0200 _H	Peripheral Group 6
<RSENT5_base>	FFCF 0280 _H	Peripheral Group 6
<RSENT6_base>	FFCF 0300 _H	Peripheral Group 6
<RSENT7_base>	FFCF 0380 _H	Peripheral Group 6
<RSENT8_base>	FFCF 0400 _H	Peripheral Group 6
<RSENT9_base>	FFCF 0480 _H	Peripheral Group 6
<RSENT10_base>	FFCF 0500 _H	Peripheral Group 6
<RSENT11_base>	FFCF 0580 _H	Peripheral Group 6
<RSENT12_base>	FFCF 0600 _H	Peripheral Group 6
<RSENT13_base>	FFCF 0680 _H	Peripheral Group 6
<RSENT14_base>	FFCF 0700 _H	Peripheral Group 6
<RSENT15_base>	FFCF 0780 _H	Peripheral Group 6
<RSENT16_base>	FFCF 0800 _H	Peripheral Group 6

23.1.3 Clock Supply

The RSENT clock supply is shown in the following table.

Table 23.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
RSENTn	clkc (RSENT communication clock)	CLK_LSB
	PCLK	CLK_LSB

23.1.4 Interrupts Requests

The RSENT interrupt requests are listed in the following table.

Table 23.6 Interrupt Requests (1/2)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
RSENT0					
INTRSENT0RI	INTSENT0RI	RSENT receive interrupt	422	group0-165	group1-73
INTRSENT0SI	INTSENT0SI	RSENT status interrupt	423	—	—
RSENT1					
INTRSENT1RI	INTSENT1RI	RSENT receive interrupt	424	group0-166	group1-74
INTRSENT1SI	INTSENT1SI	RSENT status interrupt	425	—	—
RSENT2					
INTRSENT2RI	INTSENT2RI	RSENT receive interrupt	426	group0-167	group1-75
INTRSENT2SI	INTSENT2SI	RSENT status interrupt	427	—	—
RSENT3					
INTRSENT3RI	INTSENT3RI	RSENT receive interrupt	428	group0-168	group1-76
INTRSENT3SI	INTSENT3SI	RSENT status interrupt	429	—	—
RSENT4					
INTRSENT4RI	INTSENT4RI	RSENT receive interrupt	430	group0-169	group1-77
INTRSENT4SI	INTSENT4SI	RSENT status interrupt	431	—	—
RSENT5					
INTRSENT5RI	INTSENT5RI	RSENT receive interrupt	432	group0-170	group1-78
INTRSENT5SI	INTSENT5SI	RSENT status interrupt	433	—	—
RSENT6					
INTRSENT6RI	INTSENT6RI	RSENT receive interrupt	434	group0-171	group1-79
INTRSENT6SI	INTSENT6SI	RSENT status interrupt	435	—	—
RSENT7					
INTRSENT7RI	INTSENT7RI	RSENT receive interrupt	436	group0-172	group1-80
INTRSENT7SI	INTSENT7SI	RSENT status interrupt	437	—	—
RSENT8					
INTRSENT8RI	INTSENT8RI	RSENT receive interrupt	438	group0-173	group1-81
INTRSENT8SI	INTSENT8SI	RSENT status interrupt	439	—	—
RSENT9					
INTRSENT9RI	INTSENT9RI	RSENT receive interrupt	440	group0-174	group1-82
INTRSENT9SI	INTSENT9SI	RSENT status interrupt	441	—	—
RSENT10					
INTRSENT10RI	INTSENT10RI	RSENT receive interrupt	442	group0-175	group1-83
INTRSENT10SI	INTSENT10SI	RSENT status interrupt	443	—	—

Table 23.6 Interrupt Requests (2/2)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
RSENT11					
INTRSENT11RI	INTSENT11RI	RSENT receive interrupt	444	group0-176	group1-84
INTRSENT11SI	INTSENT11SI	RSENT status interrupt	445	—	—
RSENT12					
INTRSENT12RI	INTSENT12RI	RSENT receive interrupt	446	group0-177	group1-85
INTRSENT12SI	INTSENT12SI	RSENT status interrupt	447	—	—
RSENT13					
INTRSENT13SI	INTSENT13RI	RSENT receive interrupt	448	group0-1178	group1-86
INTRSENT13RI	INTSENT13SI	RSENT status interrupt	449	—	—
RSENT14					
INTRSENT14RI	INTSENT14RI	RSENT receive interrupt	450	group0-179	group1-87
INTRSENT14SI	INTSENT14SI	RSENT status interrupt	451	—	—
RSENT15					
INTRSENT15RI	INTSENT15RI	RSENT receive interrupt	452	group0-180	group1-88
INTRSENT15SI	INTSENT15SI	RSENT status interrupt	453	—	—
RSENT16					
INTRSENT16RI	INTSENT16RI	RSENT receive interrupt	454	group0-181	group1-89
INTRSENT16SI	INTSENT16SI	RSENT status interrupt	455	—	—

23.1.5 Reset Source

The RSENT reset source is listed in the following table. RSENT is initialized by the reset source.

Table 23.7 Reset Source

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
RSENTn	All registers	✓	✓	✓	✓	—	—

23.1.6 External Input/Output Signals

The external input/output signals of RSENT are listed below.

Table 23.8 External Input/Output Signals (1/2)

Unit Signal Name	Description	Alternative Port Pin Signal
RSENT0		
sent_rx	RSENT data input	RSENT0RX
sent_spc	RSENT data output	RSENT0SPCO
RSENT1		
sent_rx	RSENT data input	RSENT1RX
sent_spc	RSENT data output	RSENT1SPCO
RSENT2		
sent_rx	RSENT data input	RSENT2RX
sent_spc	RSENT data output	RSENT2SPCO
RSENT3		
sent_rx	RSENT data input	RSENT3RX
sent_spc	RSENT data output	RSENT3SPCO
RSENT4		
sent_rx	RSENT data input	RSENT4RX
sent_spc	RSENT data output	RSENT4SPCO
RSENT5		
sent_rx	RSENT data input	RSENT5RX
sent_spc	RSENT data output	RSENT5SPCO
RSENT6		
sent_rx	RSENT data input	RSENT6RX
sent_spc	RSENT data output	RSENT6SPCO
RSENT7		
sent_rx	RSENT data input	RSENT7RX
sent_spc	RSENT data output	RSENT7SPCO
RSENT8		
sent_rx	RSENT data input	RSENT8RX
sent_spc	RSENT data output	RSENT8SPCO
RSENT9		
sent_rx	RSENT data input	RSENT9RX
sent_spc	RSENT data output	RSENT9SPCO
RSENT10		
sent_rx	RSENT data input	RSENT10RX
sent_spc	RSENT data output	RSENT10SPCO
RSENT11		
sent_rx	RSENT data input	RSENT11RX
sent_spc	RSENT data output	RSENT11SPCO
RSENT12		
sent_rx	RSENT data input	RSENT12RX
sent_spc	RSENT data output	RSENT12SPCO

Table 23.8 External Input/Output Signals (2/2)

Unit Signal Name	Description	Alternative Port Pin Signal
RSENT13		
sent_rx	RSENT data input	RSENT13RX
sent_spc	RSENT data output	RSENT13SPCO
RSENT14		
sent_rx	RSENT data input	RSENT14RX
sent_spc	RSENT data output	RSENT14SPCO
RSENT15		
sent_rx	RSENT data input	RSENT15RX
sent_spc	RSENT data output	RSENT15SPCO
RSENT16		
sent_rx	RSENT data input	RSENT16RX
sent_spc	RSENT data output	RSENT16SPCO

23.2 Overview

23.2.1 Functional Overview

- The RSENT interface supports the standard specification SAE J2716_201604 functions.
- Triple speed expansion tick time: Clock cycle (1 μ s to 90 μ s)
- Variable data transmission rate
 - up to 74.9 kbps (based on 8 nibble data at 3 μ s clock rate)
 - up to 224.7 kbps (based on 8 nibble data at 1 μ s clock rate)
- Unidirectional communication: Between the sensor and MCU
- Bidirectional communication: Between the sensor and MCU (supported in SPC mode)
- Single edge data transmission: Coded by the temporal distance of two serially-detected falling edges on a data line.
- Transmission frame with up to 8 data nibbles + status / communication nibbles.
- Data transmission protected with CRC is available.
 - CRC data can be read with the RSENTnSRXD.SCRC bits.
- Calibration phrase in each data frame (RSENTnCPL.CPLV bits)
- 1-wire interface (sent_rx and sent_spc share a single terminal.)
- Multiple sensors can be connected to the RSENT channel that has the standard expansion function.
- Received data from sensors is detected by software or DMA.
- Timestamp function is supported: Master can only be set for RSENT0.
For others, slave can only be set (RSENTnTSPC.TMS bit).

23.2.2 Block Diagram

Figure 23.1 shows a block diagram of the RSENT interface.

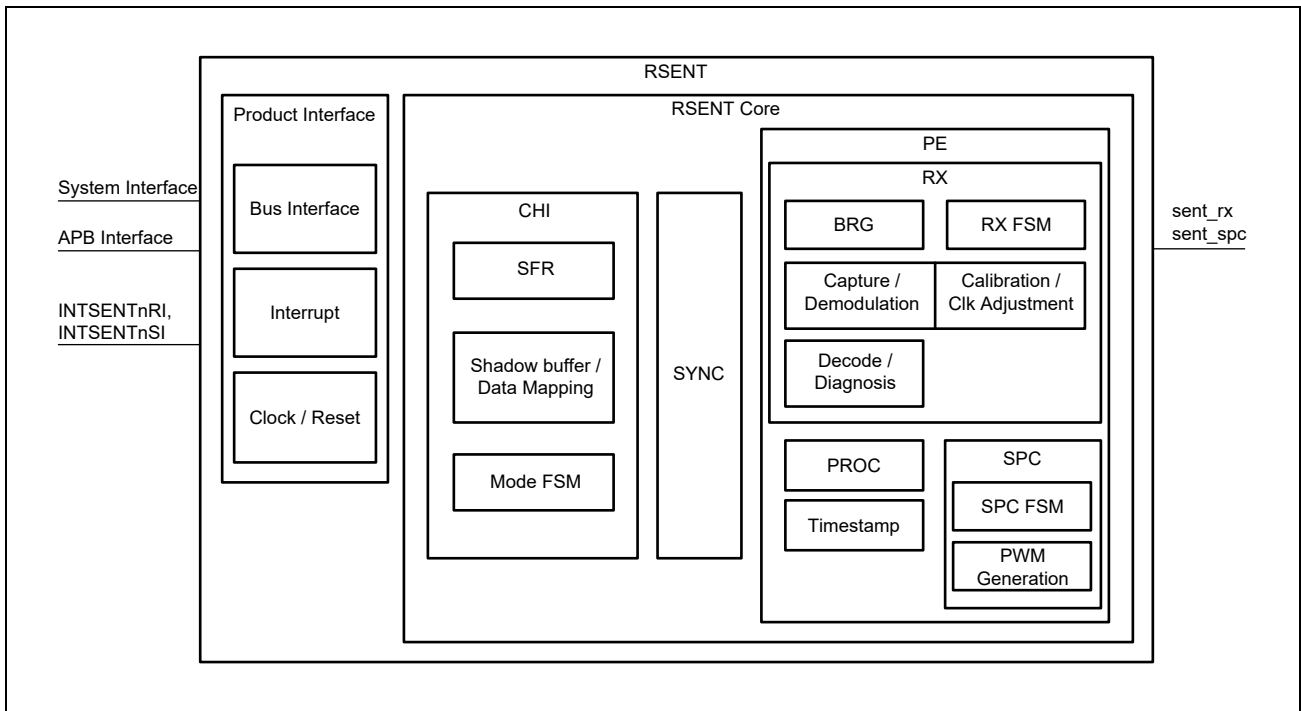


Figure 23.1 Block Diagram of RSENT

23.3 Registers

23.3.1 List of Registers

The RSENT registers are listed in the following table. For details about <RSENTn_base>, see **Section 23.1.2, Register Base Address**.

Table 23.9 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection
RSENTn	RSENT Timestamp Register	RSENTnTSPC	<RSENTn_base> + 0000 _H	32	—
	RSENT Timestamp Counter Register	RSENTnTSC	<RSENTn_base> + 0004 _H	32	—
	RSENT Communication Configuration Register	RSENTnCC	<RSENTn_base> + 0010 _H	32	—
	RSENT Baud Rate Prescaler Register	RSENTnBRP	<RSENTn_base> + 0014 _H	32	—
	RSENT Interrupt/DMA Enable Register	RSENTnIDE	<RSENTn_base> + 0018 _H	32	—
	RSENT Mode Control Register	RSENTnMDC	<RSENTn_base> + 001C _H	32	—
	RSENT SPC Transmission Register	RSENTnSPCT	<RSENTn_base> + 0020 _H	32	—
	RSENT Mode Status Register	RSENTnMST	<RSENTn_base> + 0024 _H	32	—
	RSENT Communication Status Register	RSENTnCS	<RSENTn_base> + 0028 _H	32	—
	RSENT Communication Status Clear Register	RSENTnCSC	<RSENTn_base> + 002C _H	32	—
	RSENT Slow Channel Receive Timestamp Register	RSENTnSRTS	<RSENTn_base> + 0030 _H	32	—
	RSENT Slow Channel Receive Data Register	RSENTnSRXD	<RSENTn_base> + 0034 _H	32	—
	RSENT Calibration Pulse Length Register	RSENTnCPL	<RSENTn_base> + 0038 _H	32	—
	RSENT Message Length Register	RSENTnML	<RSENTn_base> + 003C _H	32	—
	RSENT Fast Channel Receive Timestamp Register	RSENTnFRTS	<RSENTn_base> + 0040 _H	32	—
	RSENT Fast Channel Receive Data Register	RSENTnFRXD	<RSENTn_base> + 0044 _H	32	—
	RSENT Calibration Pulse Length Mirror Register	RSENTnCPLM	<RSENTn_base> + 0050 _H	32	—
	RSENT Message Length Mirror Register	RSENTnMLM	<RSENTn_base> + 0054 _H	32	—
	RSENT Fast Channel Receive TimeStamp Mirror Register	RSENTnFRISM	<RSENTn_base> + 0058 _H	32	—
	RSENT Expanded Fast Channel Receive Data Register 0	RSENTnEFRD0	<RSENTn_base> + 005C _H	32	—
RSENT Expanded Fast Channel Receive Data Register 1	RSENTnEFRD1	<RSENTn_base> + 0060 _H	32	—	

23.3.2 RSENTnTSPC — RSENT Timestamp Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TTM[6:0]						—	TTPV[6:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.10 RSENTnTSPC Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved. When read, the value after reset is read. When writing, write the value after reset.
16	TMS	Timestamp Mode Selection 0: Master mode 1: Slave mode
15	—	Reserved. When read, the value after reset is read. When writing, write the value after reset.
14 to 8	TTM[6:0]	Timestamp Tick Multiplier 0000000 _B : 1 0000001 _B : 2 0000010 _B : 3 : 1111111 _B : 128
7	—	Reserved. When read, the value after reset is read. When writing, write the value after reset.
6 to 0	TTPV[6:0]	Timestamp Tick Prescaler Value 00100111 _B : 40 Other than above: Setting prohibited

Write the above value in the TTPV bits for transmission / reception via the RSENT.

Timestamp Mode Selection (RSENTnTSPC.TMS)

This bit defines the timestamp counter synchronization mode.

For information about the timestamp clock settings, see **Section 23.4.2.1, Timestamp**.

When this bit is set to 0, the timestamp counter operates in master mode.

When writing 0000 0000_H to RSENTnTSC, the timestamp counter is cleared. In addition all RSENT timestamp counters operating as slave of RSENT are also cleared.

When this bit is set to 1, the timestamp counter operates in slave mode.

The timestamp counter is only cleared by writing 0000 0000_H to the timestamp counter of RSENT that operates in master mode.

The CPU can only write to this bit if the RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

The RSENT operating in slave mode should have the same timestamp counter prescaler settings as the RSENT that

operates in master mode.

The CPU should not set this bit to 1 for RSENT that operates in master mode.

NOTE

In this product series, RSENT0 is the only channel that can be set as a master.

Before synchronizing the timestamp counter, be sure to set RSENT0 and other channels as master mode and slave mode, respectively.

Synchronization of the timestamp counter when the above setting is not made is not guaranteed.

Timestamp Tick Multiplier (RSENTnTSPC.TTM)

These bits define the multiplication value of the 1- μ s time tick used for the timestamp counter.

For the timestamp clock configuration, see **Section 23.4.2.1, Timestamp**.

The CPU can only write to these bits if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

Timestamp Tick Prescaler Value (RSENTnTSPC.TTPV)

These bits define the prescaler value to generate a 1- μ s clock tick.

For the timestamp clock configuration, see **Section 23.4.2.1, Timestamp**.

The CPU can only write to these bits if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

The CPU should configure this value in such a way that, based on the supplied communication clock, a 1- μ s clock tick is generated.

23.3.3 RSENTnTSC — RSENT Timestamp Counter Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TS[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.11 RSENTnTSC Register Contents

Bit Position	Bit Name	Function
31 to 0	TS[31:0]	Timestamp counter value

Timestamp (RSENTnTSC.TS)

These bits indicate the current timestamp counter value.

The CPU can only write to these bits values other than 0000 0000_H if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

When the timestamp counter is configured to operate in slave mode (RSENTnTSPC.TMS = 1), writing to this register has no effect when RSENT is in either of the OPERATION IDLE or OPERATION ACTIVE mode (the RSENTnMST.OMS bits are either 011_B or 101_B).

The timestamp counter is incremented on every timestamp counter tick (as configured in the RSENTnTSPC.TTPV and RSENTnTSPC.TTM bits) when RSENT is in either of the OPERATION IDLE or OPERATION ACTIVE mode (the RSENTnMST.OMS bits are either 011_B or 101_B).

When the timestamp counter is configured to operate in master mode (RSENTnTSPC.TMS = 0), the CPU writes 0000 0000_H to these bits and RSENTnTSC.TS is set to 0000 0000_H.

When the slave mode setting is made for the timestamp counter (RSENTnTSPC.TMS = 1), writing 0000 0000_H to the RSENTnTSC.TS bit leads to the RSENTnTSC.TS bits being set to 0000 0000_H.

For timestamp mode selection, see **Section 23.4.2.1, Timestamp**.

23.3.4 RSENTnCC — RSENT Communication Configuration Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SOPC	FCM	SCCD	FCCD	DCF	SMF[1:0]	PPTC	PPC	NDN[2:0]		SPCE		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.12 RSENTnCC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 13	—	Reserved. When read, the value after reset is read. When writing, write the value after reset.
12	SOPC	SPC Output Polarity Control 0: SPC pulse active high 1: SPC pulse active low
11	FCM	Frame Check Method 0: Check against next calibration pulse 1: Check against previous calibration pulse
10	SCCD	Slow Channel CRC Check 0: Slow channel CRC check enabled 1: Slow channel CRC check disabled
9	FCCD	Fast Channel CRC Check 0: Fast channel CRC check enabled 1: Fast channel CRC check disabled
8	DCF	Data nibble CRC Format 0: SAE J2716 2010/2016 format 1: Pre SAE J2716 2010 format
7, 6	SMF	Serial Message Format 00 _B : No serial message extraction 01 _B : Short serial message format 10 _B : Enhanced serial message format 11 _B : Setting prohibited
5	PPTC	Pause Pulse Type Configuration 0: Pause pulse for variable message length 1: Pause pulse for fixed message length
4	PPC	Pause Pulse Configuration 0: Pause pulse absent 1: Pause pulse present

Table 23.12 RSENTnCC Register Contents (2/2)

Bit Position	Bit Name	Function
3 to 1	NDN[2:0]	Number of Data Nibbles 000 _B : 1 data nibble 001 _B : 2 data nibbles 010 _B : 3 data nibbles 011 _B : 4 data nibbles 100 _B : 5 data nibbles 101 _B : 6 data nibbles 110 _B : 7 data nibbles 111 _B : 8 data nibbles
0	SPCE	SPC Mode Enable 0: SPC mode disabled 1: SPC mode enabled

SPC Output Polarity Control (RSENTnCC.SOPC)

When this bit is set to 0, the SPC pulse is sent as an active high signal. The default output value is low level.

When this bit is set to 1, the SPC pulse is sent as an active low signal. The default output value is high level.

For the SPC operation, see also **Section 23.4.4, SPC Function**.

The CPU can only write to this bit if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

Frame Check Method (RSENTnCC.FCM)

When this bit is set to 0, the current calibration pulse is compared with the next received calibration pulse.

The buffer update mechanism operates according to the preferred option as described in SAE J2716 2010.

When this bit is set to 1, the current calibration pulse is compared with the previously received calibration pulse.

The buffer update mechanism operates according to the second option as described in SAE J2716 2010. This mechanism should be used only if extra latency to process the second calibration pulse cannot be tolerated.

For the buffer update timings, see also **Section 23.4.3.2(3), Fast Channel Message Reception**.

The CPU can only write to this bit if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

Slow Channel CRC Check Disable (RSENTnCC.SCCD)

When this bit is set to 1, CRC check for the slow channel is disabled. In this case, messages are stored in the slow channel message buffer with the received CRC.

When this bit is set to 1, the RSENTnCS.SCS bit is not set.

The CPU can only write to this bit if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

Fast Channel CRC Check Disable (RSENTnCC.FCCD)

When this bit is set to 1, CRC check for the fast channel is disabled. In this case, messages are stored in the fast channel message buffer with the received CRC.

When this bit is set to 1, the RSENTnCS.FCS bit is not set.

The CPU can only write to this bit if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

Date Nibble CRC Format (RSENTnCC.DCF)

This bit selects between the SAE J2716 2016 data nibble CRC format and the legacy format.

When this bit is set to 0, the recommended CRC implementation according to SAE J2716 2016 is selected.

When this bit is set to 1, the legacy CRC implementation according to SAE J2716 2008 is selected.

The CPU can only write to this bit if RSENT is in the CONFIGURATION mode (RSENTnMST.OMS = 001_B).

Serial Message Format (RSENTnCC.SMF)

These bits define the serial message format expected to be received for automatic extraction.

When these bits are set to 00_B, no serial message is extracted and the communication and status nibble are provided in the RSENTnSRXD register.

The CPU can only write to these bits if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

The CPU shall set these bits to 00 when RSENTnCC.SPCE is set to 1 and more than one sensor is connected to RSENT.

Pause Pulse Type Configuration (RSENTnCC.PPTC)

This bit defines the pause pulse type.

The CPU can only write to this bit if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

The CPU should not set this bit to 1 when the RSENTnCC.PPC bit is set to 0.

Pause Pulse Configuration (RSENTnCC.PPC)

This bit defines the presence or absence of the pause pulse.

The CPU can only write to this bit if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

Number of Data Nibbles (RSENTnCC.NDN)

These bits define the number of data nibbles included in a SENT message.

The CPU can only write to these bits if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

SPC Mode Enable (RSENTnCC.SPCE)

This bit enables the SPC mode.

For details about SPC mode operation, see also **Section 23.4.4, SPC Function**.

The CPU can only write to this bit if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

23.3.5 RSENTnBRP — RSENT Baud Rate Prescaler Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	TTF[3:0]				—	TTI[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	SCDV[6:0]							—	—	—	SCMV[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 23.13 RSENTnBPR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	—	Reserved. When read, the value after reset is read. When writing, write the value after reset.
27 to 24	TTF[3:0]	Time Tick Decimal Fraction 0000 _B : 0.0 μs 0001 _B : 0.1 μs 0010 _B : 0.2 μs : 1000 _B : 0.8 μs 1001 _B : 0.9 μs Other than above: Setting prohibited
23	—	Reserved. When read, the value after reset is read. When writing, write the value after reset.
22 to 16	TTI[6:0]	Time Tick Integer 0000000 _B : 1 μs 0000001 _B : 2 μs 0000010 _B : 3 μs : 1011000 _B : 89 μs 1011001 _B : 90 μs Other than above: Setting prohibited
15	—	Reserved. When read, the value after reset is read. When writing, write the value after reset.
14 to 8	SCDV[6:0]	Sample Clock Division Value 0000000 _B : 1 0000001 _B : 2 0000010 _B : 3 : 1111110 _B : 127 1111111 _B : 128
7 to 5	—	Reserved. When read, the value after reset is read. When writing, write the value after reset.

Table 23.13 RSENTnBPR Register Contents (2/2)

Bit Position	Bit Name	Function
4 to 0	SCMV[4:0]	Sample Clock Multiplication Value
		00000 _B : 1
		00001 _B : 2
		00010 _B : 3
		:
		11110 _B : 31
		11111 _B : 32

Time Tick Decimal Fraction (RSENTnBRP.TTF)

These bits define the decimal part of the tick length in 0.1- μ s granularity.

For the tick length configuration, see **Section 23.4.2.2(2), RX and SPC Tick Settings**.

The CPU can only write to these bits if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

Time Tick Integer (RSENTnBRP.TTI)

These bits define the integer part of the tick length.

For the tick length configuration, see **Section 23.4.2.2(2), RX and SPC Tick Settings**.

The CPU can only write to these bits if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

Sample Clock Division Value (RSENTnBRP.SCDV)

These bits define the division value for the sample clock generation logic.

For the RSENTnBRP settings, see **Section 23.4.2.2(2), RX and SPC Tick Settings**.

The CPU can only write to these bits if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

Sample Clock Multiplication Value (RSENTnBRP.SCMV)

These bits define the multiplication value for the sample clock generation logic.

For the RSENTnBRP settings, see **Section 23.4.2.2(2), RX and SPC Tick Settings**.

The CPU can only write to these bits if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

23.3.6 RSENTnIDE — RSENT Interrupt / DMA Enable Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SEIE	SMIE	SCIE	NRIE	CVIE	CLIE	FNIE	FEIE	FMIE	FCIE	FRIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.14 RSENTnIDE Register Contents

Bit Position	Bit Name	Function
31 to 11	—	Reserved. When read, the value after reset is read. When writing, write the value after reset.
10	SEIE	Slow Channel Encoding Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
9	SMIE	Slow Channel Message Lost Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
8	SCIE	Slow Channel CRC Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
7	NRIE	No Response Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
6	CVIE	Calibration Pulse Length Variation Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
5	CLIE	Calibration Pulse Length Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
4	FNIE	Fast Channel Nibble Count Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
3	FEIE	Fast Channel Nibble Encoding Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
2	FMIE	Fast Channel Message Lost Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
1	FCIE	Fast Channel CRC Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
0	FRIE	Fast Channel Receive Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled

Slow Channel Encoding Error Interrupt Enable (RSENTnIDE.SEIE)

This bit enables the generation of the slow channel encoding error interrupt.
The CPU cannot write to this bit if RSENT is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

Slow Channel Message Lost Interrupt Enable (RSENTnIDE.SMIE)

This bit enables the generation of the slow channel message lost interrupt.
The CPU cannot write to this bit if RSENT is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

Slow Channel CRC Error Interrupt Enable (RSENTnIDE.SCIE)

This bit enables the generation of the slow channel CRC error interrupt.
The CPU cannot write to this bit if RSENT is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

No Response Error Interrupt Enable (RSENTnIDE.NRIE)

This bit enables the generation of the no response error interrupt.
The CPU cannot write to this bit if RSENT is in the RESET mode (the RSENTnMST.OMS bits are 000_B).
The CPU should not set this bit when the SPC mode is disabled (RSENTnCC.SPCE set to 0).

Calibration Pulse Length Variation Error Interrupt Enable (RSENTnIDE.CVIE)

This bit enables the generation of the calibration pulse length variation error interrupt.
The CPU cannot write to this bit if RSENT is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

Calibration Pulse Length Error Interrupt Enable (RSENTnIDE.CLIE)

This bit enables the generation of the calibration pulse length error interrupt.
The CPU cannot write to this bit if RSENT is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

Fast Channel Nibble Count Error Interrupt Enable (RSENTnIDE.FNIE)

This bit enables the generation of the fast channel nibble count error interrupt.
The CPU cannot write to this bit if RSENT is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

Fast Channel Nibble Encoding Error Interrupt Enable (RSENTnIDE.FEIE)

This bit enables the generation of the fast channel nibble encoding error interrupt.
The CPU cannot write to this bit if RSENT is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

Fast Channel Message Lost Interrupt Enable (RSENTnIDE.FMIE)

This bit enables the generation of the fast channel message lost interrupt.
The CPU cannot write to this bit if the RSENT is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

Fast Channel CRC Error Interrupt Enable (RSENTnIDE.FCIE)

This bit enables the generation of the fast channel CRC error interrupt.
The CPU cannot write to this bit if RSENT is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

Fast Channel Receive Interrupt Enable (RSENTnIDE.FRIE)

This bit enables the generation of the fast channel receive interrupt.
The fast channel receive interrupt can be also used to notify a DMA request.
The CPU cannot write to this bit if RSENT is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

23.3.7 RSENTnMDC — RSENT Mode Control Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	OMC[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 23.15 RSENTnMDC Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved. When read, the value after reset is read. When writing, write the value after reset.
2 to 0	OMC[2:0]	Operation Mode Control 000 _B : RESET 001 _B : CONFIGURATION 011 _B : OPERATION IDLE 101 _B : OPERATION ACTIVE Other than above: Setting prohibited

Operation Mode Control (RSENTnMDC.OMC)

These bits are used to control the operation mode of RSENT.

- 000_B: RESET
In RESET mode, the mode can only be changed to CONFIGURATION mode.
- 001_B: CONFIGURATION
In CONFIGURATION mode, the mode can only be changed to RESET mode or OPERATION ACTIVE mode.
- 011_B: OPERATION IDLE
In OPERATION IDLE mode, the mode can be changed to OPERATION ACTIVE mode, CONFIGURATION mode, or RESET mode.
- 101_B: OPERATION ACTIVE
In OPERATION ACTIVE mode, the mode can be changed to OPERATION IDLE mode, CONFIGURATION mode, or RESET mode. However, it is recommended to proceed to the OPERATION IDLE mode first.
For the recommended methods to change between operation modes, see **Section 23.4.3.1, Changing Operation Modes**.
- Other than above: Setting prohibited
The CPU should not write any other value than listed above into this register.
The CPU should follow the mode change flows as shown in **Section 23.4.3.1, Changing Operation Modes**.

23.3.8 RSENTnSPCT — RSENT SPC Transmission Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	TLL[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.16 RSENTnSPCT Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved. When read, the value after reset is read. When writing, write the value after reset.
6 to 0	TLL[6:0]	Length of the Trigger Low Phase in Ticks 000000 _B : 1 tick 000001 _B : 2 ticks 000010 _B : 3 ticks : 1111110 _B : 127 ticks 1111111 _B : 128 ticks

Trigger Low Length (RSENTnSPCT.TLL)

These bits define the length of the SPC trigger pulse.

When the CPU writes to these bits, an SPC trigger pulse with the configured length is sent immediately independently of the current status of the RSENT.

For details about SPC communication, see **Section 23.4.4, SPC Function**.

The CPU can only write to these bits if RSENT is in the OPERATION ACTIVE mode (the RSENTnMST.OMS bits are 101_B) and SPC communication is enabled (RSENTnCC.SPCE is 1).

It is important to note that two consecutive write access might not cause a no response error as the previous request might not have started yet.

After writing to this register, the CPU should wait for at least one SPC trigger tick before writing again to this register.

23.3.9 RSENTnMST — RSENT Mode Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	OMS[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.17 RSENTnMST Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved. When read, the value after reset is read.
2 to 0	OMS[2:0]	Operation Mode Status 000 _B : RESET 001 _B : CONFIGURATION 011 _B : OPERATION IDLE 101 _B : OPERATION ACTIVE Other than above: Reserved

Operation Mode Status (RSENTnMST.OMS)

These bits indicate the current operation mode.

These bits are read only.

These bits are updated after a mode change request is made in the RSENTnMDC.OMC register.

- 000_B: RESET mode

When in RESET mode, all registers are set to their reset values, and write access to all registers except the RSENTnMDC register is disabled.

When in RESET mode, SENT communication is disabled.

- 001_B: CONFIGURATION mode

When in CONFIGURATION mode, write access to the timestamp registers (RSENTnTSPC and RSENTnTSC registers), configuration registers (RSENTnCC and RSENTnBRP registers), RSENTnIDE register, and mode control register (RSENTnMDC.OMC) is enabled.

When in CONFIGURATION mode, SENT communication is disabled.

When entering CONFIGURATION mode, all status registers and receive buffer registers are set to their reset values.

- 011_B: OPERATION IDLE mode

In OPERATION IDLE mode, no reception or SPC trigger transmission is possible.

When entering OPERATION IDLE mode, frames in the receive buffer can be analyzed as in OPERATION ACTIVE mode, but no new frames are received.

- 101_B: OPERATION ACTIVE mode

In OPERATION ACTIVE mode, reception and SPC trigger transmission are possible.

- Other than above: Reserved

23.3.10 RSENTnCS — RSENT Communication Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SES	SMS	SCS	NRS	CVS	CLS	FNS	FES	FMS	FCS	FRS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.18 RSENTnCS Register Contents

Bit Position	Bit Name	Function
31 to 11	—	Reserved. When read, the value after reset is read.
10	SES	Slow channel Encoding Error Interrupt Detection 0: Not detected 1: Detected
9	SMS	Slow Channel Message Lost Interrupt Detection 0: Not detected 1: Detected
8	SCS	Slow Channel CRC Error Interrupt Detection 0: Not detected 1: Detected
7	NRS	No Response Error Interrupt Detection 0: Not detected 1: Detected
6	CVS	Calibration Pulse Length Variation Error Interrupt Detection 0: Not detected 1: Detected
5	CLS	Calibration Pulse Length Error Interrupt Detection 0: Not detected 1: Detected
4	FNS	Fast Channel Nibble Count Error Interrupt Detection 0: Not detected 1: Detected
3	FES	Fast Channel Nibble Encoding Error Interrupt Detection 0: Not detected 1: Detected
2	FMS	Fast Channel Message Lost Interrupt Detection 0: Not detected 1: Detected
1	FCS	Fast Channel CRC Error Interrupt Detection 0: Not detected 1: Detected
0	FRS	Fast Channel Receive Interrupt Detection 0: Not detected 1: Detected

Slow Channel Encoding Error Status (RSENTnCS.SES)

This bit represents the slow channel encoding error status.

This bit is read only.

In the short serial message format (RSENTnCC.SMF = 01_B), this bit is set when the sequence on serial start bit (status & communication nibble bit #3) is different from “1000 0000 0000 0000_B” (a single 1 and 15 0s).

In the enhanced serial message format (RSENTnCC.SMF = 10_B), this bit is set after receiving the serial message start frame (sequence on status & communication nibble bit 3 is 0111 1110_B), and if bit 13 or bit 18 are not received as “0”.

This bit is cleared by writing 1 to RSENTnCSC.SEC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

Slow Channel Message Lost Status (RSENTnCS.SMS)

This bit represents the slow channel message lost status.

This bit is read only.

This bit is set when there is an attempt to update the slow channel message buffer, but the previous message has not been read yet.

This bit is cleared by writing 1 to RSENTnCSC.SEC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

Slow Channel CRC Error Status (RSENTnCS.SCS)

This bit represents the slow channel CRC error status.

This bit is read only.

This bit is set when a CRC error is detected on the slow channel and the slow channel CRC detection is enabled (RSENTnCC.SCCD is set to 0).

This bit is cleared by writing 1 to RSENTnCSC.SCC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

No Response Error Status (RSENTnCS.NRS)

This bit represents the no response error status.

This bit is read only.

This bit is set when

- The CPU writes to the RSENTnSPCT.TLL bits
- SPC mode enabled (RSENTnCC.SPCE set to 1)
- No complete response was received from the sensor for the previous SPC trigger

This bit is set after 4 PCLK + 5 clk (Maximum time) from the CPU writes to the RSENTnSPCT.TLL.

This bit is cleared by writing 1 to RSENTnCSC.NRC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

Calibration Pulse Length Variation Error Status (RSENTnCS.CVS)

This bit represents the calibration pulse length variation error status.

This bit is read only.

When RSENTnCC.PPTC is 0, then this bit is set when two successive calibration pulses differ by more than 1.5625%.

When RSENTnCC.PPTC is 1, this bit is never set. In this mode (pause pulse with fixed message length), the CPU needs to check the variation of the ratio of calibration pulse to message length by reading the RSENTnCPL and RSENTnML registers.

This bit is cleared by writing 1 to RSENTnCSC.CVC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).
If the set condition occurs simultaneously with the clear condition, the bit is set.

Calibration Pulse Length Error Status (RSENTnCS.CLS)

This bit represents the calibration pulse length error status.

This bit is read only.

This bit is set when the measured calibration pulse length is less than 42 clock ticks or more than 70 clock ticks (deviation of 25% from specification length (56 clock ticks)).

This bit is cleared by writing 1 to RSENTnCSC.CLC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

Fast Channel Nibble Count Error Status (RSENTnCS.FNS)

This bit represents the fast channel nibble count error status.

This bit is read only.

This bit is set when there is an unexpected number of falling edges between two calibration pulses.

This bit is cleared by writing 1 to RSENTnCSC.FNC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

Fast Channel Nibble Encoding Error Status (RSENTnCS.FES)

This bit represents the fast channel nibble encoding error status.

This bit is read only.

This bit is set when on the fast channel a measured nibble period is less than 12 clock ticks or more than 27 clock ticks.

This bit is cleared by writing 1 to RSENTnCSC.FEC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

Fast Channel Message Lost Status (RSENTnCS.FMS)

This bit represents the fast channel message lost status.

This bit is read only.

This bit is set when the fast channel message buffer is updated, but the previous messages in the foreground and background buffer have not been read yet.

This bit is cleared by writing 1 to RSENTnCSC.FMC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

Fast Channel CRC Error Status (RSENTnCS.FCS)

This bit represents the fast channel CRC error status.

This bit is read only.

This bit is set when a CRC error is detected on the fast channel and the fast channel CRC detection is enabled (RSENTnCC.FCCD is set to 0).

This bit is cleared by writing 1 to RSENTnCSC.FCC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

Fast Channel Receive Status (RSENTnCS.FRS)

This bit represents the fast channel receive status.

This bit is read only.

This bit is set when the fast channel receive message buffer was updated.

This bit is cleared when the CPU reads the RSENTnFRXD.FND bit.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).
If the set condition occurs simultaneously with the clear condition, the bit is set.

23.3.11 RSENTnCSC — RSENT Communication Status Clear Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SEC	SMC	SCC	NRC	CVC	CLC	FNC	FEC	FMC	FCC	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 23.19 RSENTnCSC Register Contents

Bit Position	Bit Name	Function
31 to 11	—	Reserved. When read, the value after reset is read. When writing, write the value after reset.
10	SEC	Slow Channel Encoding Error Interrupt Clear 0: — 1: Clear
9	SMC	Slow Channel Message Lost Interrupt Clear 0: — 1: Clear
8	SCC	Slow Channel CRC Error Interrupt Clear 0: — 1: Clear
7	NRC	No Response Error Interrupt Clear 0: — 1: Clear
6	CVC	Calibration Pulse Length Variation Error Interrupt Clear 0: — 1: Clear
5	CLC	Calibration Pulse Length Error Interrupt Clear 0: — 1: Clear
4	FNC	Fast Channel Nibble Count Error Interrupt Clear 0: — 1: Clear
3	FEC	Fast Channel Nibble Encoding Error Interrupt Clear 0: — 1: Clear
2	FMC	Fast Channel Message Lost Interrupt Clear 0: — 1: Clear
1	FCC	Fast Channel CRC Error Interrupt Clear 0: — 1: Clear
0	—	Reserved. When read, the value after reset is read. When writing, write the value after reset.

Slow Channel Encoding Error Clear (RSENTnCSC.SEC)

Writing 1 sets RSENTnCS.SES to 0.

Writing 0 has no effect.

This bit is always read as 0.

Slow Channel Message Lost Clear (RSENTnCSC.SMC)

Writing 1 sets RSENTnCS.SMS to 0.

Writing 0 has no effect.

This bit is always read as 0.

Slow Channel CRC Error Clear (RSENTnCSC.SCC)

Writing 1 sets RSENTnCS.SCS to 0.

Writing 0 has no effect.

This bit is always read as 0.

No Response Error Clear (RSENTnCSC.NRC)

Writing 1 sets RSENTnCS.NRS to 0.

Writing 0 has no effect.

This bit is always read as 0.

Calibration Pulse Length Variation Error Clear (RSENTnCSC.CVC)

Writing 1 sets RSENTnCS.CVS to 0.

Writing 0 has no effect.

This bit is always read as 0.

Calibration Pulse Length Error Clear (RSENTnCSC.CLC)

Writing 1 sets RSENTnCS.CLS to 0.

Writing 0 has no effect.

This bit is always read as 0.

Fast Channel Nibble Count Error Clear (RSENTnCSC.FNC)

Writing 1 sets RSENTnCS.FNS to 0.

Writing 0 has no effect.

This bit is always read as 0.

Fast Channel Nibble Encoding Error Clear (RSENTnCSC.FEC)

Writing 1 sets RSENTnCS.FES to 0.

Writing 0 has no effect.

This bit is always read as 0.

Fast Channel Message Lost Clear (RSENTnCSC.FMC)

Writing 1 sets RSENTnCS.FMS to 0.

Writing 0 has no effect.

This bit is always read as 0.

Fast Channel CRC Error Clear (RSENTnCSC.FCC)

Writing 1 sets RSENTnCS.FCS to 0.

Writing 0 has no effect.

This bit is always read as 0.

23.3.12 RSENTnSRTS — RSENT Slow Channel Receive Timestamp Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STS[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STS[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.20 RSENTnSRTS Register Contents

Bit Position	Bit Name	Function
31 to 0	STS	Slow Channel Receive Timestamp

Slow Channel Timestamp (RSENTnSRTS.STS)

These bits are read only.

These bits are updated when the slow channel receive buffer is updated with the timestamp counter value of the last frame provided to the slow channel message.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

23.3.13 RSENTnSRXD — RSENT Slow Channel Receive Data Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SND	—	SCRC[5:0]					—	—	—	SMGC	IDD[19:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IDD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.21 RSENTnSRXD Register Contents

Bit Position	Bit Name	Function
31	SND	Slow Channel New Data 0: Slow channel frame data is not updated since last read. 1: Slow channel frame data is updated since last read.
30	—	Reserved. When read, the value after reset is read.
29 to 24	SCRC[5:0]	Slow Channel CRC Data
23 to 21	—	Reserved. When read, the value after reset is read.
20	SMGC	Slow Channel Configuration 0: The slow channel receive message buffer is not updated. 1: The slow channel receive message buffer is updated.
19 to 0	IDD[19:0]	Slow Channel Data / ID Information

Slow Channel New Data (RSENTnSRXD.SND)

This bit indicates that the slow channel message buffer is holding data that has not been read.

This bit is read only.

This bit is set when the slow channel receive message buffer is updated.

This bit is cleared automatically whenever it is read.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

Slow Channel CRC (RSENTnSRXD.SCRC)

These bits represent the slow channel CRC data.

These bits are read only.

These bits are updated when the slow channel receive message buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

Slow Channel Configuration (RSENTnSRXD.SMGC)

This bit represents the slow channel configuration bit data.

This bit is read only.

This bit is updated when the slow channel receive message buffer is updated.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

ID / Data (RSENTnSRXD.IDD)

These bits represent the slow channel data and ID information.

The alignment within this register depends on the message format. For details, see **Section 23.4.3.2(5), Slow Channel Message Reception**.

These bits are read only.

These bits are updated when the slow channel receive message buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

23.3.14 RSENTnCPL — RSENT Calibration Pulse Length Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPLV [16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CPLV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.22 RSENTnCPL Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved. When read, the value after reset is read.
16 to 0	CPLV[16:0]	Calibration Pulse Length Value of Received Message

Calibration Pulse Length Value (RSENTnCPL.CPLV)

These bits are used by the CPU to calculate the ratio of two consecutive calibration pulses or the calibration pulse to message length in pause pulse with fixed message length mode for message diagnostics.

These bits are read only.

In modes other than pause pulse with fixed message length (RSENTnCC.PPTC = 1) or SPC mode (RSENTnCC.SPCE = 1), these bits are invalid and should not be used.

These bits are updated with the measured calibration pulse length in sample clock ticks when the fast channel receive message buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

23.3.15 RSENTnML — RSENT Message Length Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	MLV[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MLV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.23 RSENTnML Register Contents

Bit Position	Bit Name	Function
31 to 21	—	Reserved. When read, the value after reset is read.
20 to 0	MLV[20:0]	Message length of received message

Message Length Value (RSENTnML.MLV)

These bits are used by the CPU to calculate the ratio of the calibration pulse to message length in pause pulse with fixed message length mode for message diagnostics.

These bits are read only.

In modes other than pause pulse with fixed message length, these bits are invalid and should not be used.

These bits are updated with the measured message length in sample clock ticks when the fast channel receive message buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

23.3.16 RSENTnFRTS — RSENT Fast Channel Receive Timestamp Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FTS[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.24 RSENTnFRTS Register Contents

Bit Position	Bit Name	Function
31 to 0	FTS[31:0]	Fast Channel Receive Timestamp

Fast Channel Timestamp (RSENTnFRTS.FTS)

These bits are read only.

These bits are updated when the fast channel receive message buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

23.3.17 RSENTnFRXD — RSENT Fast Channel Receive Data Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SNDM	FND	FCCN[1:0]		FCRC[3:0]			ND[23:16]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ND[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.25 RSENTnFRXD Register Contents

Bit Position	Bit Name	Function
31	SNDM	Slow Channel New Data Mirror 0: Slow channel frame data is not updated since last read. 1: Slow channel frame data is updated since last read.
30	FND	Fast Channel New Data 0: Fast channel frame data is not updated since last read. 1: Fast channel frame data is updated since last read.
29, 28	FCCN[1:0]	Fast Channel Communication Nibble bits [1:0]
27 to 24	FCRC[3:0]	Fast Channel CRC Data
23 to 0	ND[23:0]	Fast Channel Nibble Data

Slow Channel New Data Mirror (RSENTnFRXD.SNDM)

This bit indicates that the slow channel message buffer is holding data that has not been read.

This bit is read only.

This bit is set when the slow channel receive message buffer is updated.

This bit is cleared automatically whenever the slow channel new data bit (RSENTnSRXD.SND) is read.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

Fast Channel New Data (RSENTnFRXD.FND)

This bit indicates that the fast channel message buffer is holding data that has not been read.

This bit is read only.

This bit is set when the fast channel receive message buffer is updated.

This bit is cleared automatically whenever it is read RSENTnFRXD or RSENTnEFRD1.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

Fast Channel Status and Communication Nibble (RSENTnFRXD.FCCN)

These bits represent the fast channel status and communication nibble bits [1:0].

These bits are read only.

These bits are updated when the fast channel receive message buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

Fast Channel CRC (RSENTnFRXD.FCRC)

These bits represent the fast channel CRC data.

These bits are read only.

These bits are updated when the fast channel receive message buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

Nibble Data (RSENTnFRXD.ND)

These bits represent the fast channel nibble data.

The alignment of the nibble data depends on nibble data count (RSENTnCC.NDN). For details, see **Section 23.4.3.2(3), Fast Channel Message Reception**.

These bits are read only.

These bits are updated when the fast channel receive message buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

23.3.18 RSENTnCPLM — RSENT Calibration Pulse Length Mirror Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPLVM [16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CPLVM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.26 RSENTnCPLM Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved. When read, the value after reset is read.
16 to 0	CPLVM[16:0]	Calibration pulse length Value (Mirror) of received message

Calibration Pulse Length Value Mirror (RSENTnCPLM.CPLVM)

These bits are mirror bits of RSENTnCPL.CPLV.

These bits are read only.

In modes other than pause pulse with fixed message length (RSENTnCC.PPTC = 1) or SPC mode (RSENTnCC.SPCE = 1), these bits are invalid and should not be used.

These bits are updated with the measured calibration pulse length in sample clock ticks when the fast channel receive message buffer is updated.

These bits are cleared when RSENTnMST.OMS is changed to 001_B (CONFIGURATION).

23.3.19 RSENTnMLM — RSENT Message Length Mirror Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	MLVM[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MLVM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.27 RSENTnMLM Register Contents

Bit Position	Bit Name	Function
31 to 21	—	Reserved. When read, the value after reset is read.
20 to 0	MLV[20:0]	Message length value (Mirror) of received message

Message Length Value Mirror (RSENTnMLM.MLVM)

These bits are mirror bits of RSENTnML.MLV.

These bits are read only.

These bits are updated when the fast channel receive message buffer is updated.

These bits are cleared when RSENTnMST.OMS is changed to 001_B (CONFIGURATION).

23.3.20 RSENTnFRTSM — RSENT Fast Channel Receive Timestamp Mirror Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FTSM[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTSM[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.28 RSENTnFRTSM Register Contents

Bit Position	Bit Name	Function
31 to 0	FRTSM[31:0]	Fast channel receive Timestamp (Mirror)

Fast Channel TimeStamp Mirror (RSENTnFRTSM.FTSM)

These bits are mirror bits of RSENTnFRTS.FTS.

These bits are read only.

These bits are updated when the fast channel receive message buffer is updated.

These bits are cleared when RSENTnMST.OMS is changed to 001_B (CONFIGURATION).

23.3.21 RSENTnEFRD0 — RSENT Expanded Fast Channel Receive Data Register 0

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SNDM	FND	FCCN[1:0]	FCRC[3:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.29 RSENTnEFRD0 Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved. When read, the value after reset is read.
7	SNDM	Slow channel New Data (Mirror) 0: Slow channel frame data not updated since last read 1: Slow channel frame data updated since last read
6	FND	Fast channel New Data 0: Fast channel frame data not updated since last read 1: Fast channel frame data updated since last read
5 to 4	FCCN[1:0]	Fast Channel Communication Nibble bits [1:0]
3 to 0	FCRC[3:0]	Fast channel CRC data

Slow Channel New Data Mirror (RSENTnEFRD0.SNDM)

This bit indicates that the slow channel message buffer is holding data that has not been read.

This bit is read only.

This bit is set when the slow channel receive message buffer is updated.

This bit is cleared automatically whenever the slow channel new data bit (RSENTnSRXD.SND) is read.

This bit is cleared when RSENTnMST.OMS is changed to 001_B (CONFIGURATION).

Fast Channel New Data (RSENTnEFRD0.FND)

This bit indicates that the fast channel message buffer is holding data that has not been read.

This bit is read only.

This bit is set when the fast channel receive message buffer is updated.

This bit is cleared automatically whenever it is read RSENTnFRXD or RSENTnEFRD1.

This bit is cleared when RSENTnMST.OMS is changed to 001_B (CONFIGURATION).

Fast Channel Communication Nibble (RSENTnEFRD0.FCCN)

These bits represent the fast channel communication nibble bits [1:0].

These bits are read only.

These bits are updated when the fast channel receive message buffer is updated.

These bits are cleared when RSENTnMST.OMS is changed to 001_B (CONFIGURATION).

Fast Channel CRC (RSENTn EFRD0.FCRC)

These bits represent the fast channel CRC data.

These bits are read only.

These bits are updated when the fast channel receive message buffer is updated.

These bits are cleared when RSENTnMST.OMS is changed to 001_B (CONFIGURATION).

23.3.22 RSENTnEFRD1 — RSENT Expanded Fast Channel Receive Data Register 1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ND[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ND[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.30 RSENTnEFRD1 Register Contents

Bit Position	Bit Name	Function
31 to 0	ND[31:0]	Nibble Data Fast channel nibble data

Nibble Data (RSENTnEFRD1.ND)

These bits represent the fast channel nibble data.

The alignment of the nibble data depends on nibble count. For details, refer to **Section 23.4.3.2(3), Fast Channel Message Reception**.

These bits are read only.

These bits are updated when the fast channel receive message buffer is updated.

These bits are cleared when RSENTnMST.OMS is changed to 001_B (CONFIGURATION).

23.4 Operation

23.4.1 Modes of Operation

RSENTn operates in any of the following modes:

- RESET mode
- CONFIGURATION mode
- OPERATION IDLE mode
- OPERATION ACTIVE mode

The mode transitions should follow the flow in **Section 23.4.3.1, Changing Operation Modes**.

The RSENTnMST.OMS bit can be used to confirm the current operation mode.

Figure 23.2 shows the possible transitions between the channel modes:

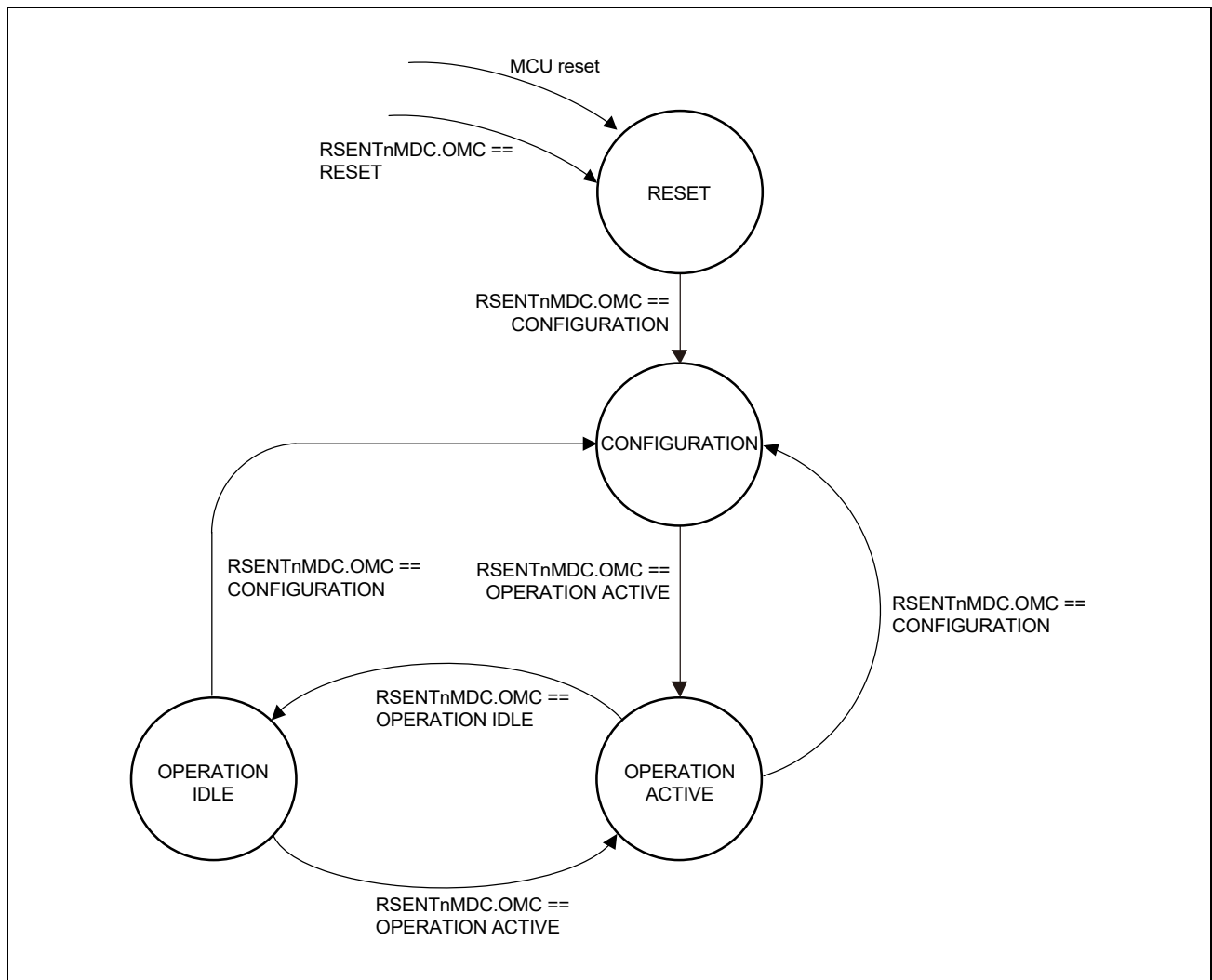


Figure 23.2 Transition between Operation Modes

23.4.1.1 RESET Mode

This mode is the initial mode that the RSENT automatically enters after release from the hardware reset (MCU reset). Its purpose is to provide a clean reset of the registers in RSENT.

The RESET mode is also entered after the RSENTnMDC.OMC bits have been set to 000_B. In this state, all configuration, control (except RSENTnMDC.OMC bits), and status registers are set to their reset value. Any on-going transmission or reception process is stopped immediately and the interface pins of RSENT are set to the value after their reset.

Read access to all registers is possible in RESET mode. Write access is limited to the RSENTnMDC register.

23.4.1.2 CONFIGURATION Mode

The CONFIGURATION mode is entered after the RSENTnMDC.OMC bits have been set to 001_B.

The interface pins of the RSENT are set to their default values.

In CONFIGURATION mode, all status registers (RSENTnCS) and the receive buffer registers (RSENTnSRTS, RSENTnSRXD, RSENTnCPL, RSENTnML, RSENTnFRTS, and RSENTnFRXD, RSENTnCPLM, RSENTnMLM, RSENTnFRTSM, RSENTnEFRD0, RSENTnEFRD1) are set to their default value.

Read access to all registers is possible in this state.

Write access is limited to both timestamp registers (RSENTnTSPC and RSENTnTSC) and configuration registers (RSENTnCC, RSENTnBRP, RSENTnIDE, and RSENTnMDC).

23.4.1.3 OPERATION IDLE Mode

This mode is entered after the RSENTnMDC.OMC bits have been set to 011_B.

In OPERATION IDLE mode, no reception and transmission are done.

When entering OPERATION IDLE mode, frames in the receive buffer can be analyzed as in OPERATION ACTIVE mode, but no new frames are received.

Read access to all registers is possible in OPERATION IDLE mode.

Write access is limited to RSENTnTSC, RSENTnIDE, RSENTnMDC, and RSENTnCSC.

23.4.1.4 OPERATION ACTIVE Mode

This mode is entered after the RSENTnMDC.OMC bits have been set to 101_B.

In OPERATION ACTIVE mode, transmission and reception can take place.

Read access to all registers is possible in this state.

Write access is limited to RSENTnTSC, RSENTnIDE, RSENTnMDC, RSENTnSPCT, and RSENTnCSC.

23.4.1.5 Register Behavior in Operation Modes

Table 23.31 shows the register behavior when RSENT transitions to the indicated operation modes. The table also gives an overview of the access restriction in each operation mode.

Table 23.31 Register Behavior in Operation Modes

Register Name	Symbol	Reset	RESET Mode		CONFIGURATION Mode		OPERATION IDLE Mode		OPERATION ACTIVE Mode	
		Change	Change	R/W	Change	R/W	Change	R/W	Change	R/W
Timestamp Prescaler Configuration Register	RSENTnTSPC	0000 0000 _H	0000 0000 _H	R	Unchanged	R/W	Unchanged	R	Unchanged	R
Timestamp Counter Register	RSENTnTSC	0000 0000 _H	0000 0000 _H	R	Unchanged	R/W	Unchanged	R/W*1	Unchanged	R/W*1
Communication Configuration Register	RSENTnCC	0000 0000 _H	0000 0000 _H	R	Unchanged	R/W	Unchanged	R	Unchanged	R
Baud Rate Prescaler Register	RSENTnBRP	0000 0000 _H	0000 0000 _H	R	Unchanged	R/W	Unchanged	R	Unchanged	R
Interrupt/DMA Enable Register	RSENTnIDE	0000 0000 _H	0000 0000 _H	R	Unchanged	R/W	Unchanged	R/W	Unchanged	R/W
Mode Control Register	RSENTnMDC	0000 0000 _H	0000 0000 _H	R/W	Unchanged	R/W	Unchanged	R/W	Unchanged	R/W
SPC Transmission Register	RSENTnSPCT	0000 0000 _H	0000 0000 _H	R	Unchanged	R	Unchanged	R	Unchanged	R/W
Mode Status Register	RSENTnMST	0000 0000 _H	0000 0000 _H	R	0000 0001 _H	R	0000 0003 _H	R	0000 0005 _H	R
Communication Status Register	RSENTnCS	0000 0000 _H	0000 0000 _H	R	0000 0000 _H	R	Unchanged	R	Unchanged	R
Communication Status Clear Register	RSENTnCSC	0000 0000 _H	0000 0000 _H	R	Unchanged	R/W	Unchanged	R/W	Unchanged	R/W
Slow channel Receive Timestamp Register	RSENTnSRTS	0000 0000 _H	0000 0000 _H	R	0000 0000 _H	R	Unchanged	R	Unchanged	R
Slow channel Receive Data Register	RSENTnSRXD	0000 0000 _H	0000 0000 _H	R	0000 0000 _H	R	Unchanged	R	Unchanged	R
Calibration Pulse Length Register	RSENTnCPL	0000 0000 _H	0000 0000 _H	R	0000 0000 _H	R	Unchanged	R	Unchanged	R
Message Length Register	RSENTnML	0000 0000 _H	0000 0000 _H	R	0000 0000 _H	R	Unchanged	R	Unchanged	R
Fast channel Receive Timestamp Register	RSENTnFRTS	0000 0000 _H	0000 0000 _H	R	0000 0000 _H	R	Unchanged	R	Unchanged	R
Fast channel Receive Data Register	RSENTnFRXD	0000 0000 _H	0000 0000 _H	R	0000 0000 _H	R	Unchanged	R	Unchanged	R
Calibration Pulse Length Mirror Register	RSENTnCPLM	0000 0000 _H	0000 0000 _H	R	0000 0000 _H	R	Unchanged	R	Unchanged	R
Message Length Mirror Register	RSENTnMLM	0000 0000 _H	0000 0000 _H	R	0000 0000 _H	R	Unchanged	R	Unchanged	R
Fast channel Receive Timestamp Mirror Register	RSENTnFRTSM	0000 0000 _H	0000 0000 _H	R	0000 0000 _H	R	Unchanged	R	Unchanged	R
Expanded Fast channel Receive Data Register0	RSENTnEFRD0	0000 0000 _H	0000 0000 _H	R	0000 0000 _H	R	Unchanged	R	Unchanged	R
Expanded Fast channel Receive Data Register1	RSENTnEFRD1	0000 0000 _H	0000 0000 _H	R	0000 0000 _H	R	Unchanged	R	Unchanged	R

Note 1. R/W: Means write restriction exists.

23.4.1.6 Mode Status Change Timing

This section shows the change timing of the RSENTnMST.OMS register, i.e. the timing properties of the arrows in **Figure 23.2**. **Table 23.32** shows the change timing with respect to falling edge detection, PCLK, clk_c and sample clock.

Table 23.32 Mode Status Change Timing

RESET → CONFIG	5 PCLK + 3 clk _c
CONFIG → OPA	5 PCLK + 6 clk _c
OPA → OPI	Falling edge of next frame sync nibble + 5 sample clocks + 4 clk _c + 4 PCLK
OPI → CONFIG	4 PCLK + 4 clk _c
OPA → CONFIG	4 PCLK + 7 clk _c + 1 sample clock (+ 1 PCLK – safety margin for user)
OPI → OPA	5 PCLK + 6 clk _c
In case frame reception has not started so far (falling edge of calibration pulse of 1st frame not detected)	
OPA → OPI	5 PCLK + 1 sample clock + 7 clk _c

Note: Latency of each mode to reset is immediate after CPU write.

OPI means "OPERATION IDLE mode".

OPA means "OPERATION ACTIVE mode".

23.4.2 Clock Configuration

23.4.2.1 Timestamp

(1) Timestamp Clock Configuration

RSENT incorporates the timestamp counter.

Depending on the supplied communication frequency(*clk*), the user should configure the RSENTnTSPC.TTPV bits to set the output of prescaler TPV to 1 μ s clock tick.

Depending on the configured tick lengths, the resolution can be decreased by configuring the RSENTnTSPC.TTM bits. The already divided input frequency is divided further by the value of the RSENTnTSPC.TTM bits.

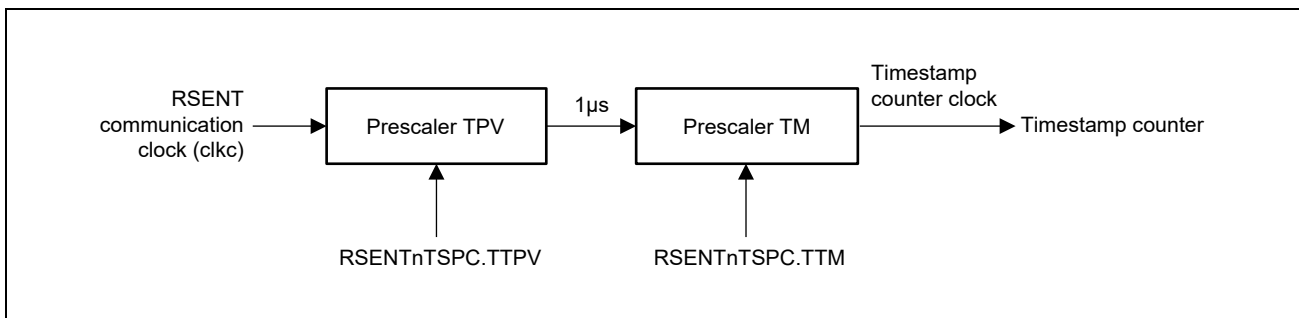


Figure 23.3 Timestamp Counter Clock Generation

(2) Timestamp Counter Operation

The timestamp counter value can be initialized to any value by writing to the RSENTnTSC.TS bits only when RSENT is in CONFIGURATION mode.

When timestamp counters are configured to operate in master mode (RSENTnTSPC.TMS = 0), the CPU can reset the timestamp counter by writing 0000 0000_H to the RSENTnTSC.TS bits when RSENT is in OPERATION_IDLE or OPERATION_ACTIVE mode.

When timestamp counters are configured to operate in slave mode (RSENTnTSPC.TMS = 1), the timestamp counter is cleared when the CPU writes 0000 0000_H to the RSENTnTSC.TS bits of the channel set in the master when RSENT is in OPERATION_IDLE or OPERATION_ACTIVE mode.

Make the same settings for the timestamp counter prescalers of channels operating in master mode and slave mode.

When timestamp counter synchronization occurs, the internal timestamp counter prescalers are also synchronized.

The current timestamp counter value can be read from the RSENTnTSC.TS bits.

When RSENT is in OPERATION_ACTIVE mode, each received message is stored with its related timestamp. Timestamp values are taken for fast channel and slow channel data.

The timestamp value is captured when the calibration pulse is detected.

The timestamp value for the fast channel is stored in the RSENTnFRTS.FTS bits or RSENTnFRTSM.FTSM bits register field when the fast channel receive message buffer is updated.

The timestamp value for the slow channel is stored in the RSENTnSRTS.STS bits. The timestamp value for the slow channel is identical to the timestamp value of the last fast channel message contributing to the slow channel message.

In case timestamp counter synchronization is required, the following flow should be used.

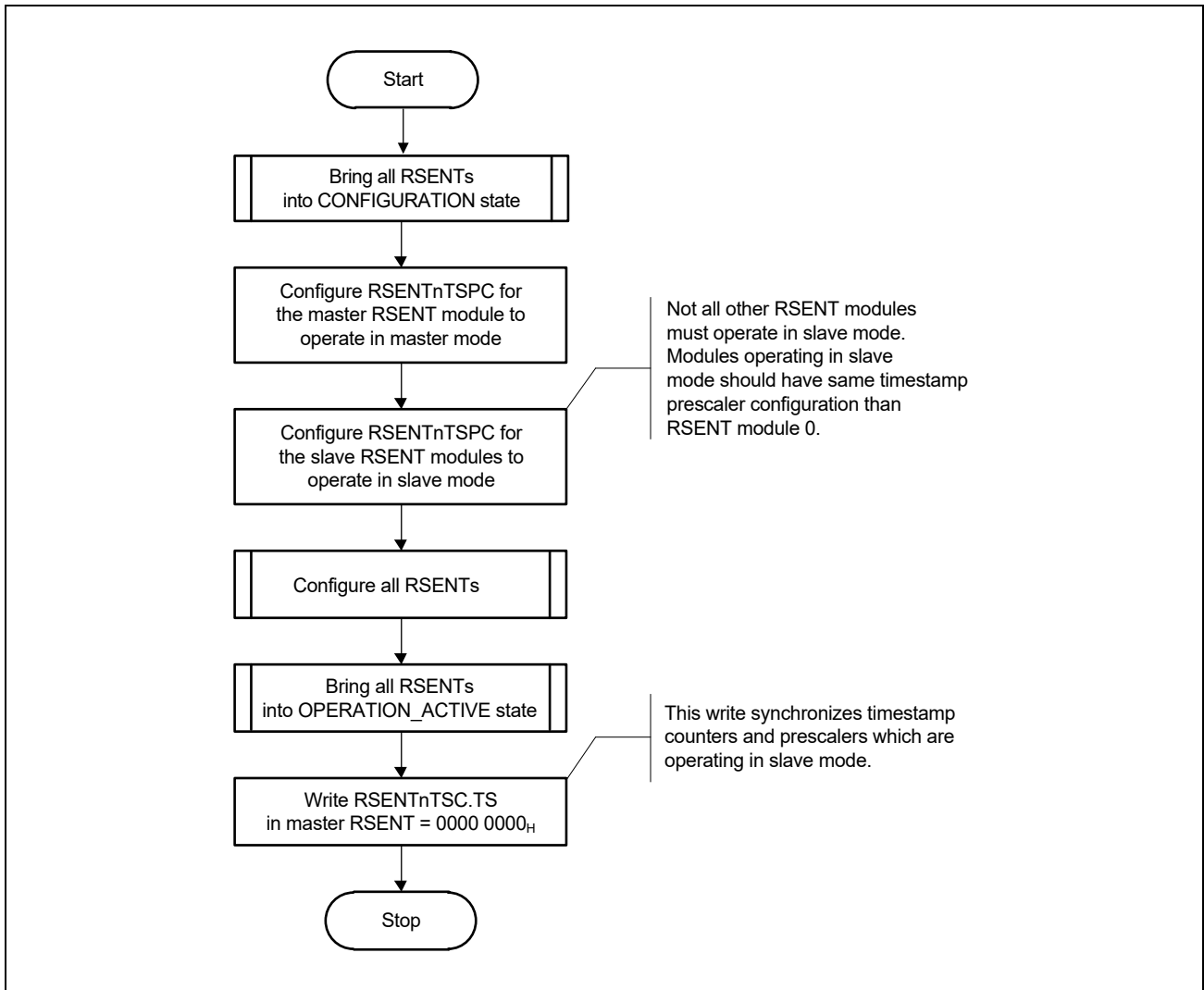


Figure 23.4 Timestamp Counter Synchronization

Synchronization of the timestamp can be done if the state of the master module has changed to OPERATION_ACTIVE or OPERATION_IDLE.

23.4.2.2 Communication Clock Configuration

(1) RX BRP Setting

“RSENT communication clock frequency ($f_{\text{COMMUNICATION}}$)” is multiplied by the configured lowest term fraction set by RSENTnBRP.SCDV and RSENTnBRP.SCMV to generate the sample clock.

The RSENTnBRP.SCDV and RSENTnBRP.SCMV values should be selected in such a way that a sample clock frequency (f_{SAMPLE}) of 16 MHz is generated according to the following formula.

$$f_{\text{SAMPLE}} = 16 \text{ MHz} = f_{\text{COMMUNICATION}} \times \frac{\text{Sample Clock Multiplication Value (RSENTnBRP.SCMV+1)}}{\text{Sample Clock Division Value (RSENTnBRP.SCDV+1)}}$$

Where Sample Clock Multiplication Value (RSENTnBRP.SCMV) = 00001_B,

Sample Clock Division Value (RSENTnBRP.SCDV) = 0000100_B,

$f_{\text{COMMUNICATION}} = 40 \text{ MHz}$

$f_{\text{SAMPLE}} = 40 \times 2/5 = 16 \text{ MHz}$

(2) RX and SPC Tick Settings

The used tick length in the RX and SPC functions can be configured with the RSENTnBRP.TTI and RSENTnBRP.TTF bits. Tick lengths from 1.0 μs to 90.0 μs with a resolution of 0.1 μs can be configured.

The RSENTnBRP.TTI holds the integer part of the tick length and the RSENTnBRP.TTF bits hold the fractional part of the tick length. The tick length is then calculated by:

$$T_{\text{Tick}} = T_{\text{RSENTnBRP.TTI}} + T_{\text{RSENTnBRP.TTF}}$$

Where RSENTnBRP.TTI = 0000000_B, RSENTnBRP.TTF = 0011_B

$T_{\text{Tick}} = 1 + 0.3 = 1.3 \mu\text{s}$

23.4.3 RSENT Operation

When starting transfer through RSENT_n in this product series, be sure to make the pin settings corresponding to the RSENT_n interface, then follow the flow for the individual settings described in this section.

For details about the pin settings, see **Section 2, Pin Function**.

23.4.3.1 Changing Operation Modes

Once initialization has been completed in CONFIGURATION mode, operation can be enabled by entering OPERATION ACTIVE mode. This is done by setting the RSENT_nMDC.OMC bits to OPERATION ACTIVE and waiting for RSENT_nMST.OMS to transition to OPERATION ACTIVE.

Once in OPERATION ACTIVE mode, RSENT begins to receive messages or SPC communication can be started depending on the configuration.

Figure 23.5 shows the communication enable flow assuming that the RSENT is in RESET mode.

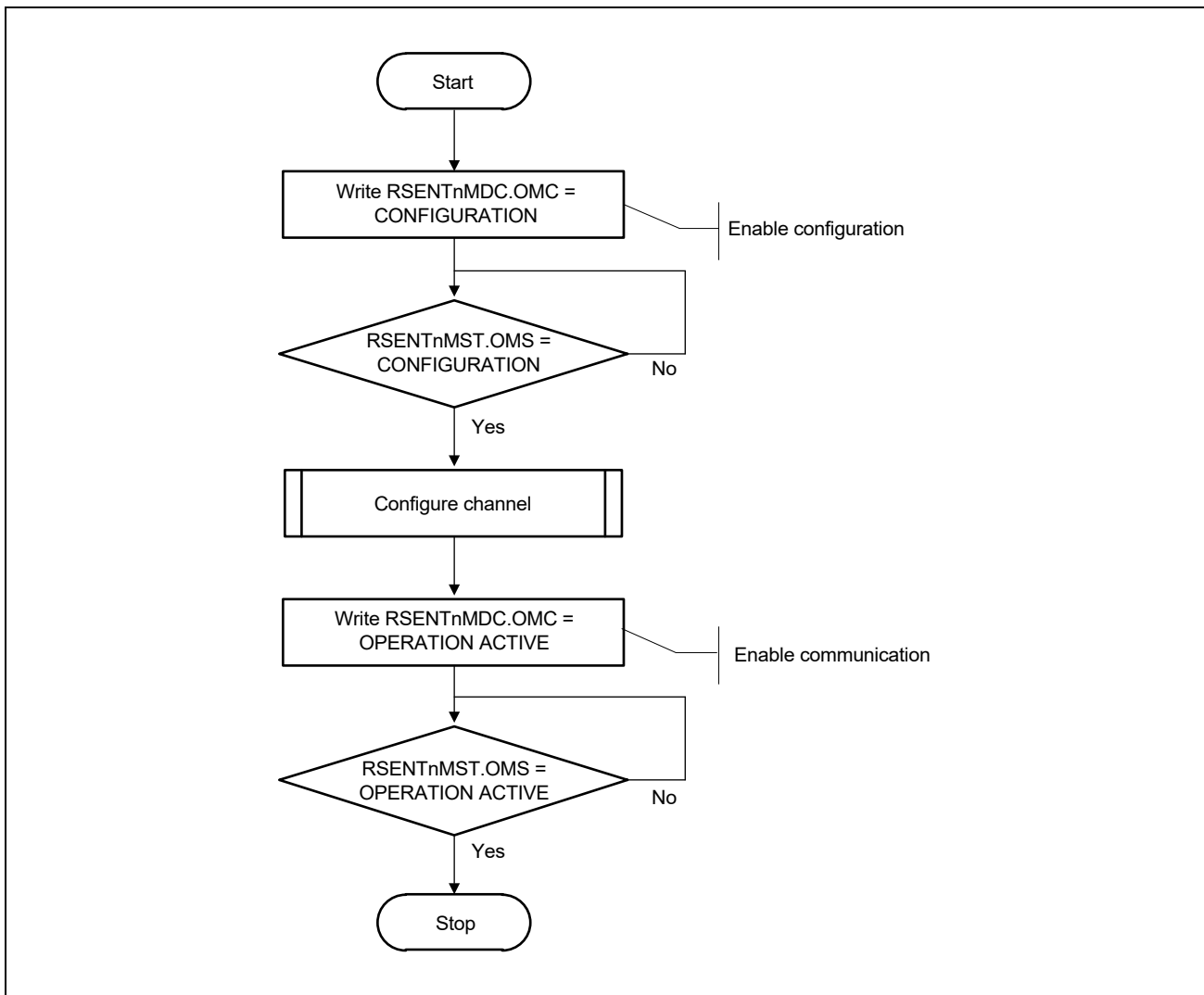


Figure 23.5 Communication Enable Flow

To leave OPERATION ACTIVE mode, communication should be disabled first by transitioning to OPERATION IDLE mode. This is done by setting the RSENTnMDC.OMC bits to OPERATION IDLE and waiting for the RSENTnMST.OMS bits to transition to OPERATION IDLE.

However, when the SPC is enabled (RSENTnCC.SPCE = 1) and the SPC trigger transmission has not been requested after the previous SPC communication has been completed (e.g. successful reception for the previous SPC request), the RSENT can directly enter the CONFIGURATION mode.

The transition between OPERATION ACTIVE and OPERATION IDLE depends on the setting of the RSENTnCC.SPCE bit.

(1) RSENTnCC.SPCE = 0

In case a reception is currently ongoing, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the receive buffer was updated (see **Section 23.4.3.2(3), Fast Channel Message Reception**).

In case no reception is ongoing, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place immediately.

(2) RSENTnCC.SPCE = 1

In case a reception is ongoing, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the falling edge of the end pulse is received.

In case a no response error is flagged, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place at the same time as the error flagging.

The mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the sequence of making a SPC trigger and receiving the response has been completed. This means when a response was already received, the transition takes place immediately. When the response is still pending, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the falling edge of the end pulse is received.

CONFIGURATION mode can be entered by writing CONFIGURATION to the RSENTnMDC.OMC bits and waiting for the RSENTnMST.OMS to transition to CONFIGURATION.

Once CONFIGURATION mode is entered, the remaining status and message information stored in RSENT is lost since status and message information is cleared in CONFIGURATION mode.

Figure 23.6 shows the communication disable flow assuming that RSENT is in OPERATION ACTIVE mode.

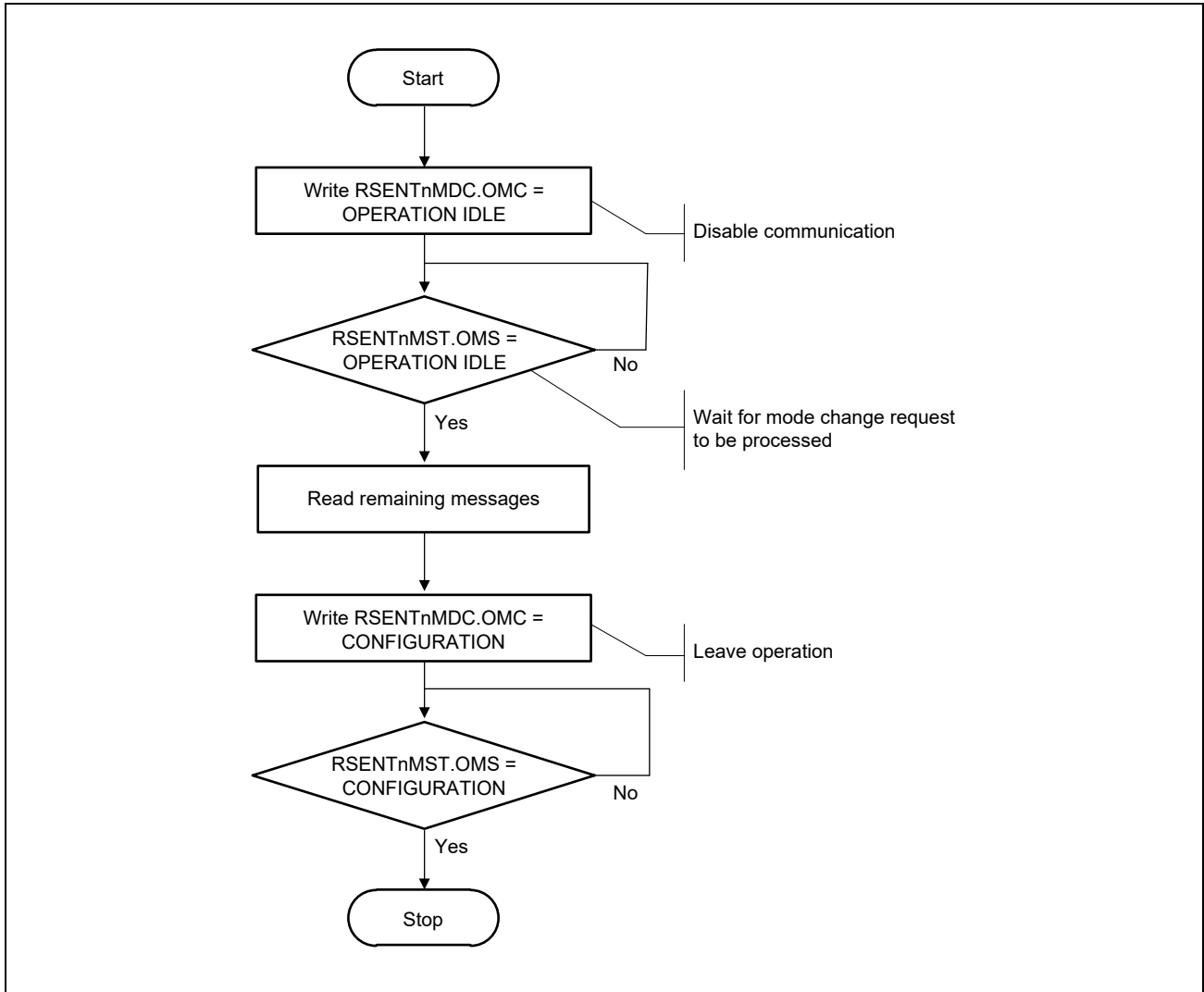


Figure 23.6 Communication Stop Flow

23.4.3.2 Message Reception

SENT message reception is composed of the calibration pulse reception followed by the data nibble pulse reception.

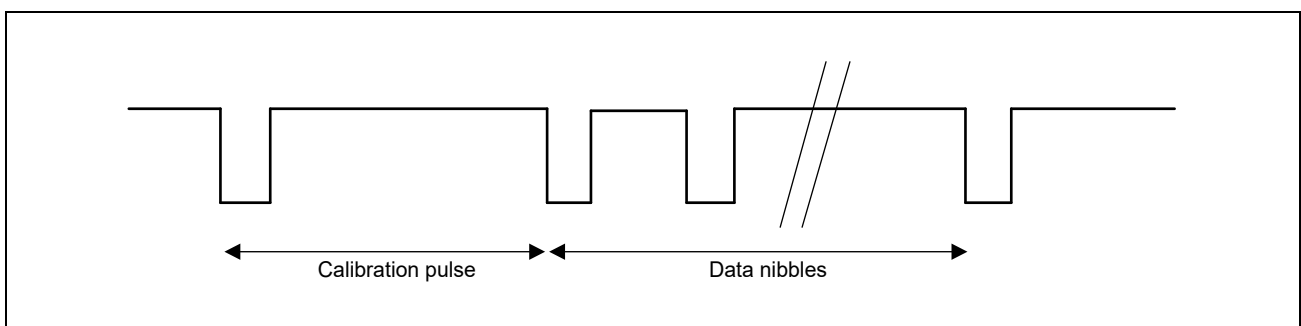


Figure 23.7 SENT Received Message Structure

(1) Calibration Pulse Reception

Within the calibration pulse reception phase, the internally generated clock tick is adjusted to the transmit clock speed.

In addition, the calibration pulse is used to end the previous message and perform message diagnostics. RSENT supports automatic calibration pulse length diagnostics in variable message length modes (RSENTnCC.PPTC = 0). In case the calibration pulse ratio check fails, the calibration pulse length variation error flag (RSENTnCS.CVS) is set to 1.

(2) Data Nibble Reception

The receive function of RSENT is a straightforward capture and compare function. The RSENT receives sensor information encoded by the temporal distance of two consecutive falling edges on the data line. The temporal distance (in # of clock ticks) is captured and compared against a set of values to determine the actual nibble value. The data encoding is illustrated in **Table 23.33** below.

Table 23.33 Data Nibble Encoding

Nibble Period (# Clock Ticks)	Nibble Value (Binary)
12	0000 _B
13	0001 _B
14	0010 _B
15	0011 _B
16	0100 _B
17	0101 _B
18	0110 _B
19	0111 _B
20	1000 _B
21	1001 _B
22	1010 _B
23	1011 _B
24	1100 _B
25	1101 _B
26	1110 _B
27	1111 _B

The received data nibbles are composed into a SENT message that is then stored in the fast channel receive message buffer.

Any other received nibble period during the reception of data nibbles will cause a fast channel nibble encoding error.

(3) Fast Channel Message Reception

Messages received on the fast message channel are stored in a receive buffer.

A fast channel receive buffer is composed of the calibration pulse length register (RSENTnCPL), the message length register (RSENTnML), the fast channel receive timestamp register (RSENTnFRTS), and the fast channel receive data register (RSENTnFRXD).

These registers are arranged on successive addresses for a transfer of the register content into memory using DMA.

The RSENT is equipped with a double receive buffer structure that allows the storage of two complete SENT messages including the related timestamp and message length information. Message decoding and assembling are done in a separate register stage.

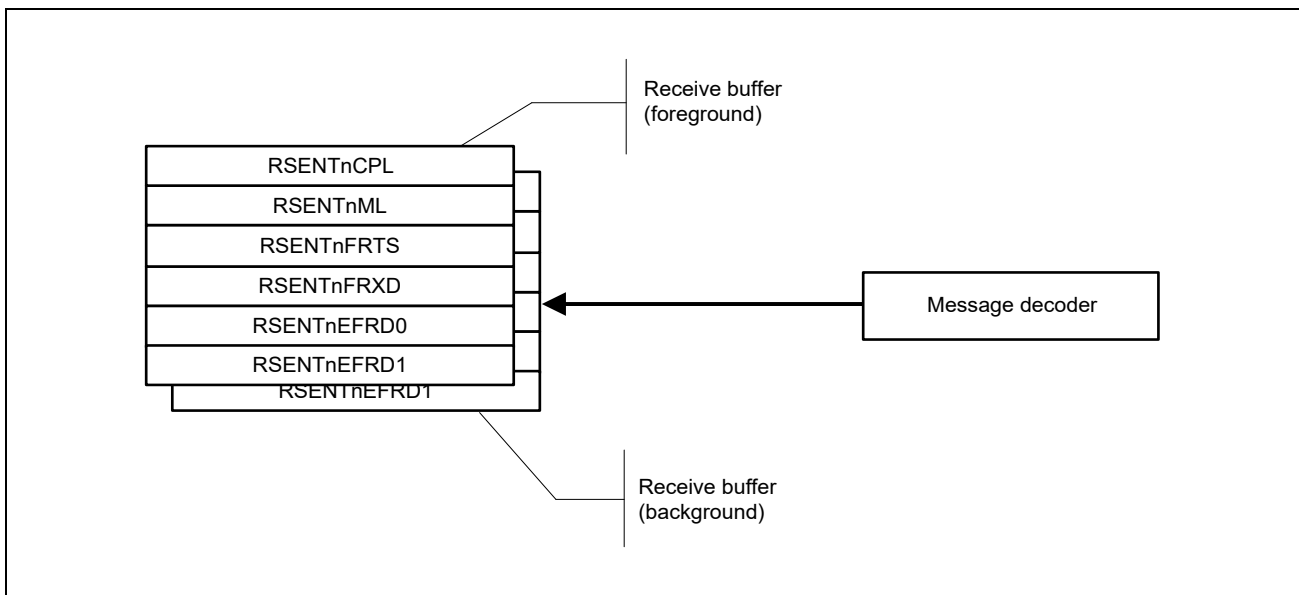


Figure 23.8 Fast Channel Receive Message Buffer

The first received message is placed into the message buffer that can be accessed by the CPU. This buffer (except the RSENTnFRXD.SNDM and RSENTnEFRD0.SNDM bits) is not updated any more until the RSENTnFRXD or RSENTnEFRD1 register is read.

When a new message is placed into a receive buffer, the RSENTnFRXD.FND and RSENTnEFRD0.FND bits are set. At the same time, the RSENTnCS.FRS bit is set and, if enabled, a receive interrupt request is generated.

When the foreground receive buffer is holding an unprocessed message (the RSENTnFRXD.FND and RSENTnEFRD0.FND bits are 1), any further incoming message is placed in the background receive buffer. The background receive buffer is updated with any further incoming messages. In case an un-processed message in the background receive buffer is overwritten, the RSENTnCS.FMS bit is set to 1.

When the CPU reads the RSENTnFRXD or RSENTnEFRD1 register and there is valid data in the background buffer, the data previously located in the background receive buffer becomes available in the receive buffer and is accessible by the CPU. If enabled, a new interrupt request for fast channel data is generated and RSENTnCS.FRS is set.

When the RSENTnFRXD.FND and RSENTnEFRD0.FND / RSENTnCS.FRS bits are not set, the data in the receive buffer is not defined and the CPU should not access the receive buffer.

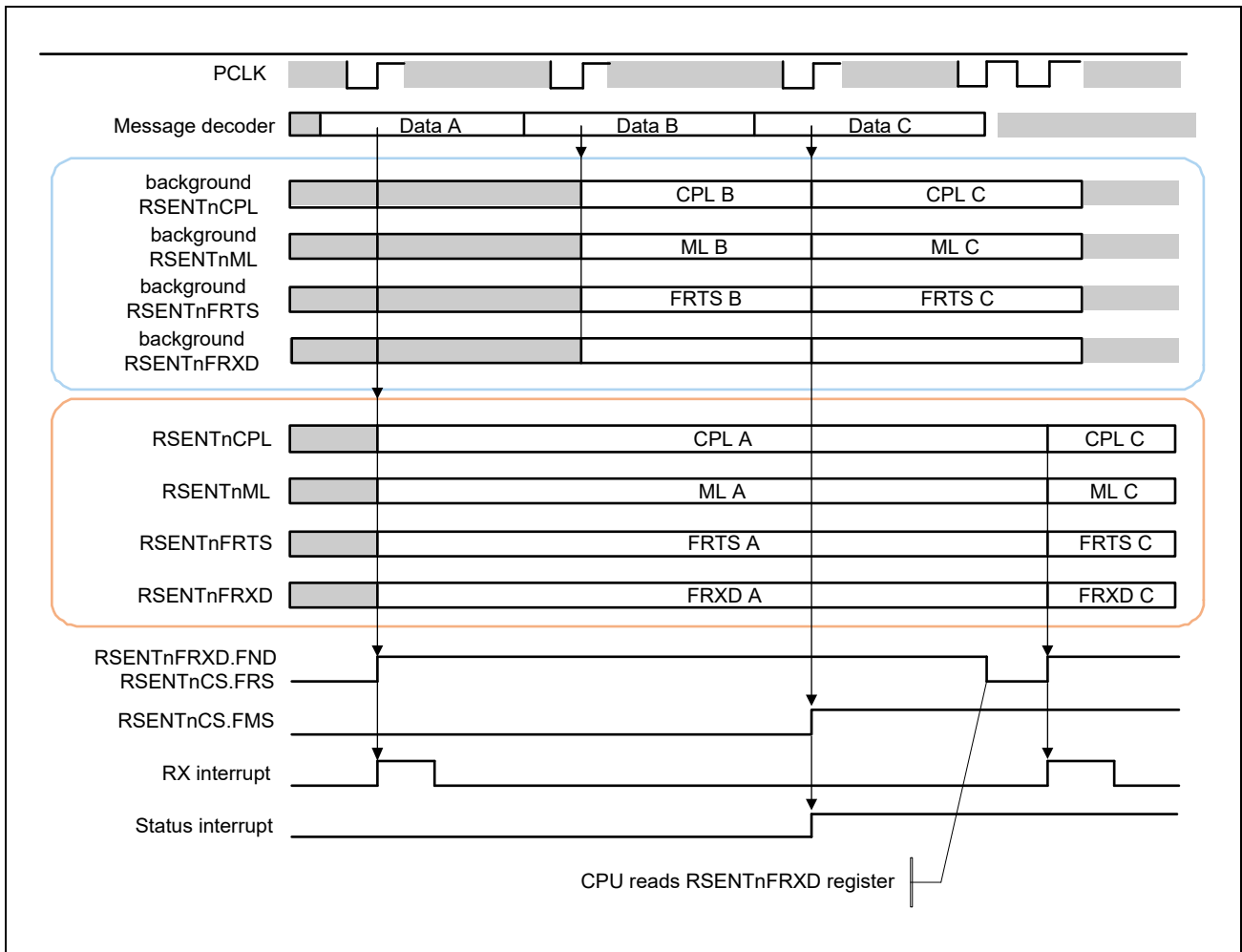


Figure 23.9 Fast Channel Receive Buffer Update Timing

The update timing of the receive buffer depends on the applied configuration as shown in **Figure 23.10** to **Figure 23.13**.

The RSENTnFRTS register is updated with the current timestamp counter register value when the calibration pulse is detected.

The data alignment in the RSENTnFRXD register depends on the nibble data count (RSENTnCC.NDN).

Table 23.34 Data Nibble Alignment in RSENTnFRXD Register

RSENTnFRXD RSENTnCC.NDN	23:20	19:16	15:12	11:8	7:4	3:0
000 _B	Undefined	Undefined	Undefined	Undefined	Undefined	Nibble 1
001 _B	Undefined	Undefined	Undefined	Undefined	Nibble 1	Nibble 2
010 _B	Undefined	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3
011 _B	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4
100 _B	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5
101 _B	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5	Nibble 6

Table 23.35 Data Nibble Alignment in RSENTnEFRD1 Register

RSENTnEFRD1 RSENTnCC.NDN	31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
000 _B	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Nibble 1
001 _B	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Nibble 1	Nibble 2
010 _B	Undefined	Undefined	Undefined	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3
011 _B	Undefined	Undefined	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4
100 _B	Undefined	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5
101 _B	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5	Nibble 6
110 _B	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5	Nibble 6	Nibble 7
111 _B	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5	Nibble 6	Nibble 7	Nibble 8

- (1) SAE operation with variable message length and preferred check method (RSENTnCC.SPCE = 0, RSENTnCC.PPTC = 0, RSENTnCC.FCM = 0)

In this operation mode, RSENT automatically performs the check for successive calibration pulse variation according to the preferred option in the J2716 2010 specification. In this mode, message diagnostics is done after the calibration pulse is received following a message.

If this check is passed, the receive message buffer is updated.

If this check is not passed, the receive message buffer is not updated and RSENTnCS.CVS is set to 1.

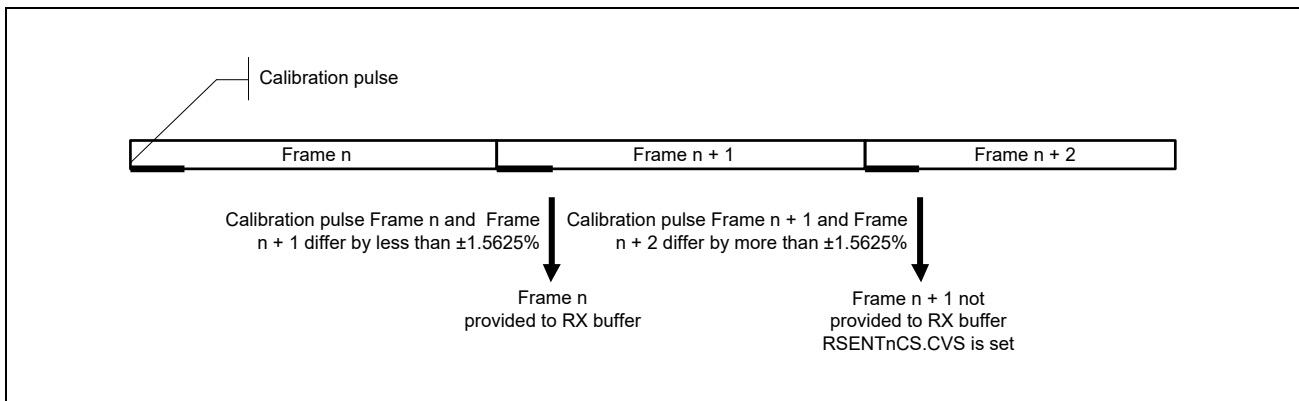


Figure 23.10 Buffer Update in Variable Message Length Mode and Preferred Check Method

- (2) SAE operation with variable message length and optional check method (RSENTnCC.SPCE = 0, RSENTnCC.PPTC = 0, RSENTnCC.FCM = 1)

In this operation mode, RSENT automatically performs the check for successive calibration pulse variation according to the optional frame check method as described in the J2716 2010 specification. In this mode, the calibration pulse of the current frame is compared with the calibration pulse of the last valid preceding frame.

If this check is passed, the receive message buffer is updated.

If this check is not passed, the receive message buffer is not updated and RSENTnCS.CVS is set to 1.

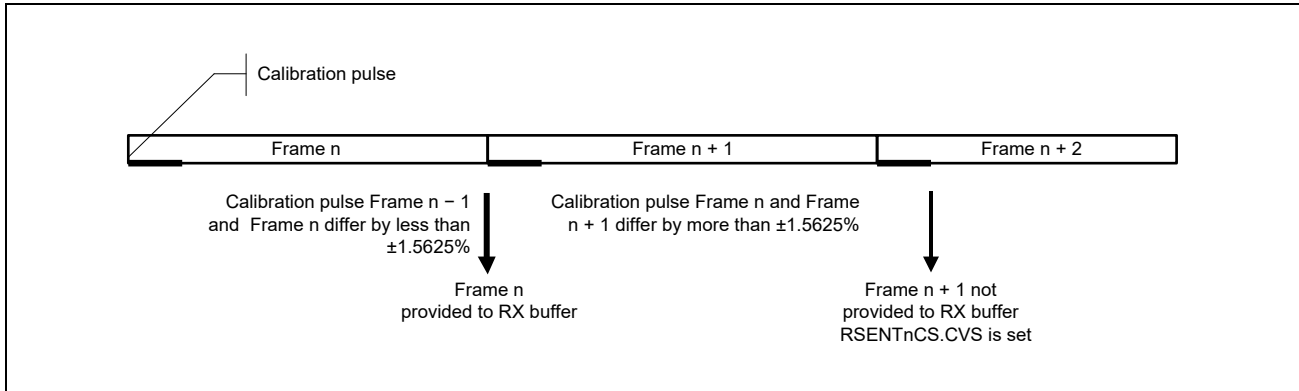


Figure 23.11 Buffer Update in Variable Message Length Mode and Optional Check Method

- (3) SAE operation with fixed message length (RSENTnCC.SPCE = 0, RSENTnCC.PPTC = 1, and RSENTnCC.PPC = 1)

In this mode, RSENT does not perform the check for calibration pulse and message length ratio according to the preferred option in the J2716 2010 specification. In this mode, RSENT provides the calibration pulse length in the RSENTnCPL register and the message length information in the RSENTnML register. The numbers provided are based on samples.

The message buffer is updated at the beginning of the following calibration pulse irrespective of the values in the RSENTnCPL and RSENTnML registers. The CPU needs to calculate the ratio and either accept or discard the message.

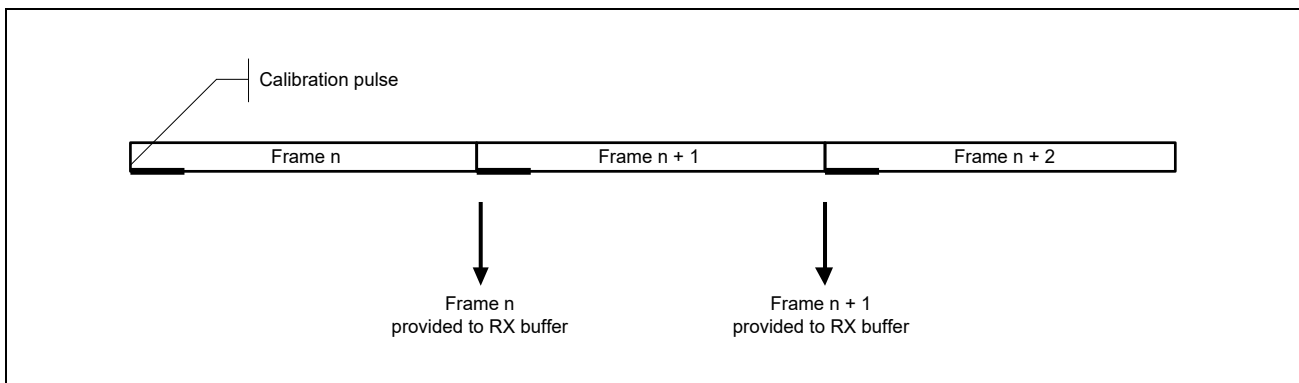


Figure 23.12 Buffer Update in Fixed Message Length Mode

The RSENTnCS.CVS (calibration pulse width variation error status) bit is never set in this mode.

(4) SPC operation (RSENTnCC.SPCE = 1)

In this operation mode, sensor data transmission is done following a SPC master trigger pulse. Within SAE SENT communication, the calibration pulse or pause pulse aborts the previous message. In SPC communication, the sensor only sends data following a SPC trigger request. An end pulse sent by the sensor aborts the message. The message buffer is updated at the beginning of the end pulse.

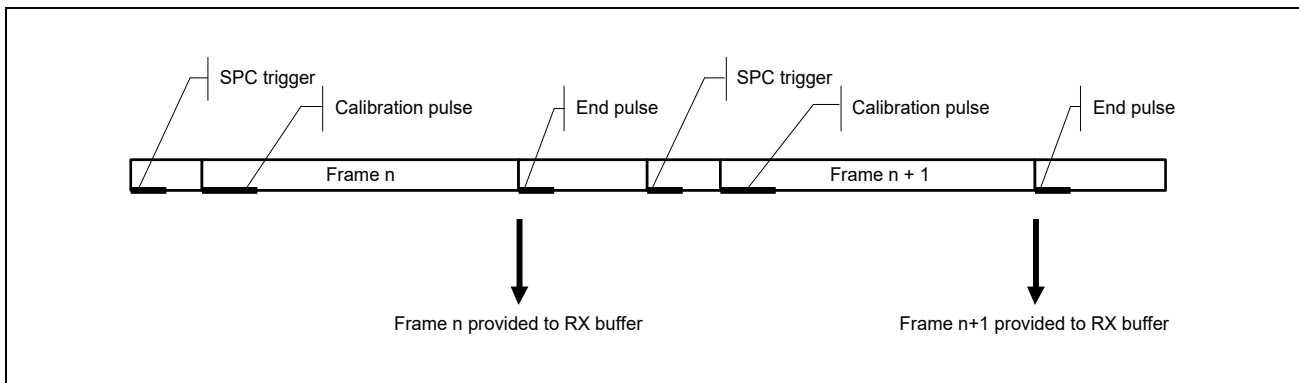


Figure 23.13 Buffer Update in SPC Mode

The RSENTnCS.CVS (calibration pulse width variation error status) bit is never set in this mode.

RSENT provides the calibration pulse length in the RSENTnCPL register and the message length information in the RSENTnML register. The numbers provided are based on samples. The CPU needs to calculate the ratio of calibration pulses and/or message length and either accept or discard the message.

In case of variable message length mode, RSENT cannot perform this check because the receive timing of the next calibration pulse depends on the next SPC trigger timing.

(4) Fast Channel Reception Flow

In **Figure 23.14**, the recommended reception flow for the fast channel receive buffer is shown.

When using a polling or event driven method, the CPU should only read the setting of the RSENTnCS.FRS bit to check the availability of new fast channel data.

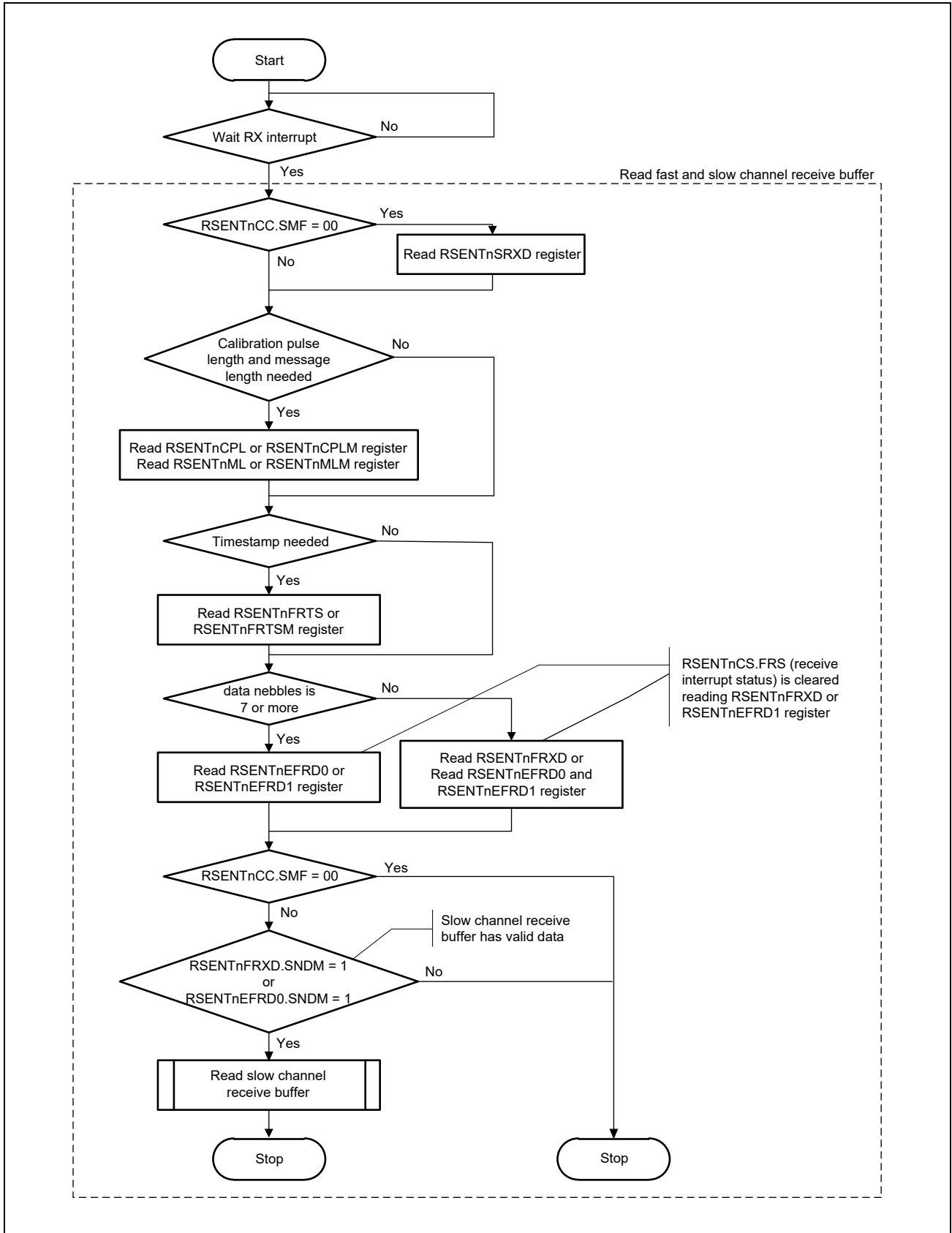


Figure 23.14 Fast Channel Reception Flow

In any case, the CPU should keep the order in reading the receive buffer registers as shown in the flow. The RSENTnFRXD or RSENTnEFRD0 register should be the last register to be accessed.

The handling of the slow channel receive buffer is described in **Section 23.4.3.2(6), Slow Channel Reception Flow**.

In case of SAE communication with pause pulse and fixed message length, the flow must be extended by checking of the ratio of the calibration pulse to message length. This variation check must be performed by the CPU. In case the variation check fails, the CPU must discard the received message.

(5) Slow Channel Message Reception

RSENT supports extraction of the slow message out of the fast channel messages by using the status and communication nibble bits 3 and 2 out of the status and communication nibble. In order to enable the slow channel extraction, the CPU should set the RSENTnCC.SMF bits to the expected serial message format.

When no serial message extraction is selected (RSENTnCC.SMF = 00_B), the RSENTnSRXD register becomes part of the fast channel receive buffer structure (including buffer 2) and the RSENTnSRTS register should be ignored. The communication and status nibble is placed in the RSENTnSRXD.IDD bits. Furthermore, no slow channel new data and slow channel message lost flags are generated.

In order to receive the slow channel serial message, all fast channel serial messages contributing to a slow channel serial message must be received successfully and the received slow channel serial message must comply with the selected serial message format.

A message lost on the fast channel does not impact the reception on the slow channel.

A slow channel receive buffer is composed of the slow channel receive timestamp register (RSENTnSRTS) and the slow channel receive data register (RSENTnSRXD).

Contrary to the fast channel receive buffer, the slow channel receive buffer does not support a double receive buffer structure; only a single receive buffer structure is available.

Message decoding and assembling is done in a separate register stage.

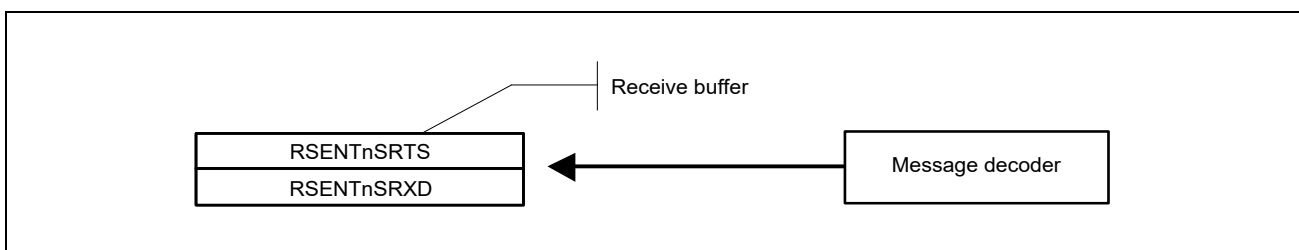


Figure 23.15 Slow Channel Receive Message Buffer

The slow channel receive buffer is updated at the same time as the fast channel receive buffer that holds the last status and communication nibble required for the slow channel message. The RSENT-nSRXD.SND bit is set to 1 at the same time.

Further updates to the buffer are not carried out until after the RSENTnSRXD.SND bit has been read.

When the receive buffer is holding an unprocessed message (RSENTnSRXD.SND is 1), any further incoming message is lost (the slow channel receive buffer is not updated) and RSENTnCS.SMS is set to 1.

When the CPU reads the RSENTnSRXD register, RSENTnSRXD.SND is automatically cleared.

The RSENTnSRTS register is updated with the current timestamp counter register value of the last frame contributing to the slow channel message.

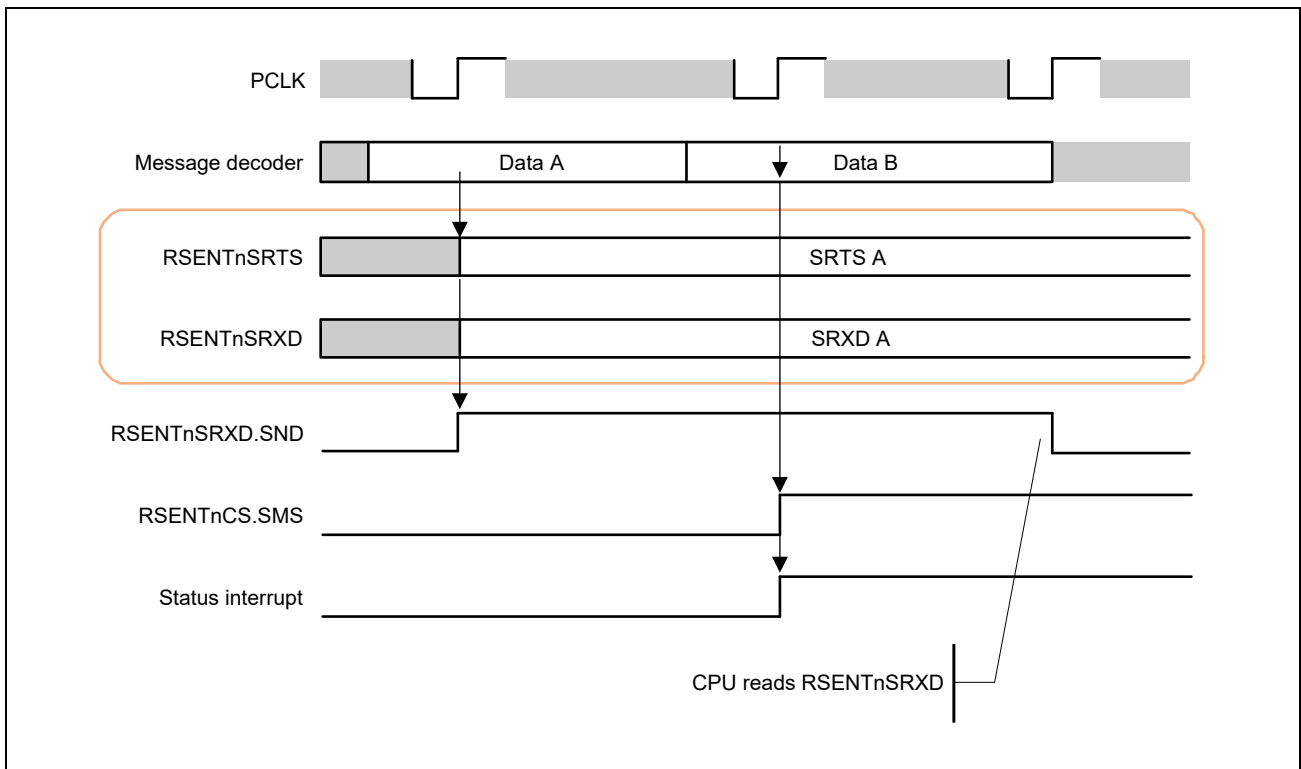


Figure 23.16 Slow Channel Receive Buffer Update Timing

The data alignment in the RSENTnSRDX register depends on the slow channel message format (RSENTnCC.SMF) and the received configuration bit.

Table 23.36 Data Alignment in RSENTnSRXD Register

RSENTn CC.SMF	RSENTnSRXD. SMGC	RSENTnSRXD. IDD[19:16]	RSENTnSRXD. IDD[15:12]	RSENTnSRXD. IDD[11:8]	RSENTnSRXD. IDD[7:4]	RSENTnSRXD. IDD[3:0]
00 _B	Undefined	Undefined	Undefined	Undefined	Undefined	C & S nibble
01 _B	Undefined	Undefined	Undefined	Message ID[3:0]	DATA[7:4]	DATA[3:0]
10 _B	0	Message ID[7:4]	Message ID[3:0]	DATA[11:8]	DATA[7:4]	DATA[3:0]
10 _B	1	Message ID[3:0]	DATA[15:12]	DATA[11:8]	DATA[7:4]	DATA[3:0]

(6) Slow Channel Reception Flow

In **Figure 23.17**, the recommended reception flow for the slow channel receive buffer is shown. When the slow channel receive data is required, this process should be executed as part of the fast channel reception flow.

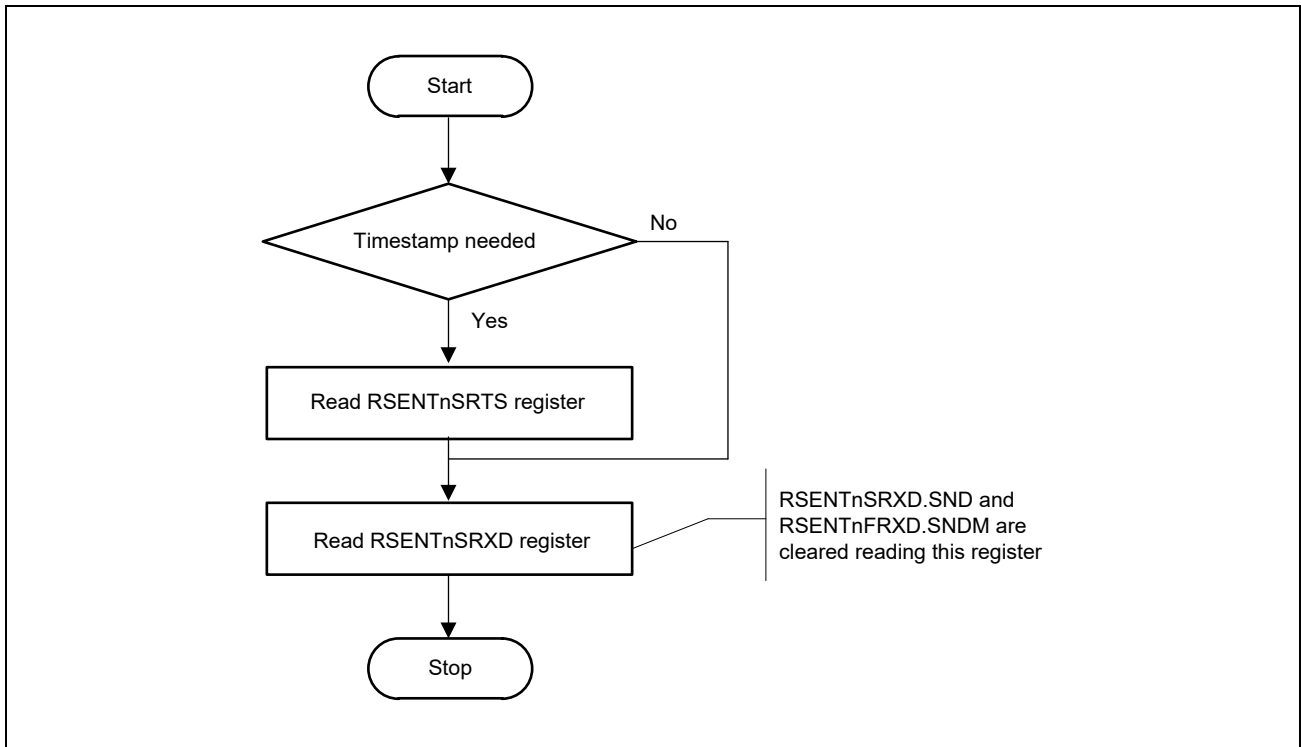


Figure 23.17 Slow Channel Reception Flow

In any case, the CPU should keep the order in reading the slow channel receive buffer registers as shown in the flow. The RSENTnSRXD.SND bit should be accessed last.

(7) DMA Flow

In case of DMA usage, the start address for the DMA usage and the number of transfers define which part of the receive buffer will be transferred. The RSENTnFRXD or RSENTnEFRD1 register should be the last register to be accessed using the 32-bit access method.

In case of SAE communication with pause pulse and fixed message length, the flow must be extended by checking of the ratio of the calibration pulse to message length. This variation check must be performed by the CPU. In case the variation check fails, the CPU must discard the received message.

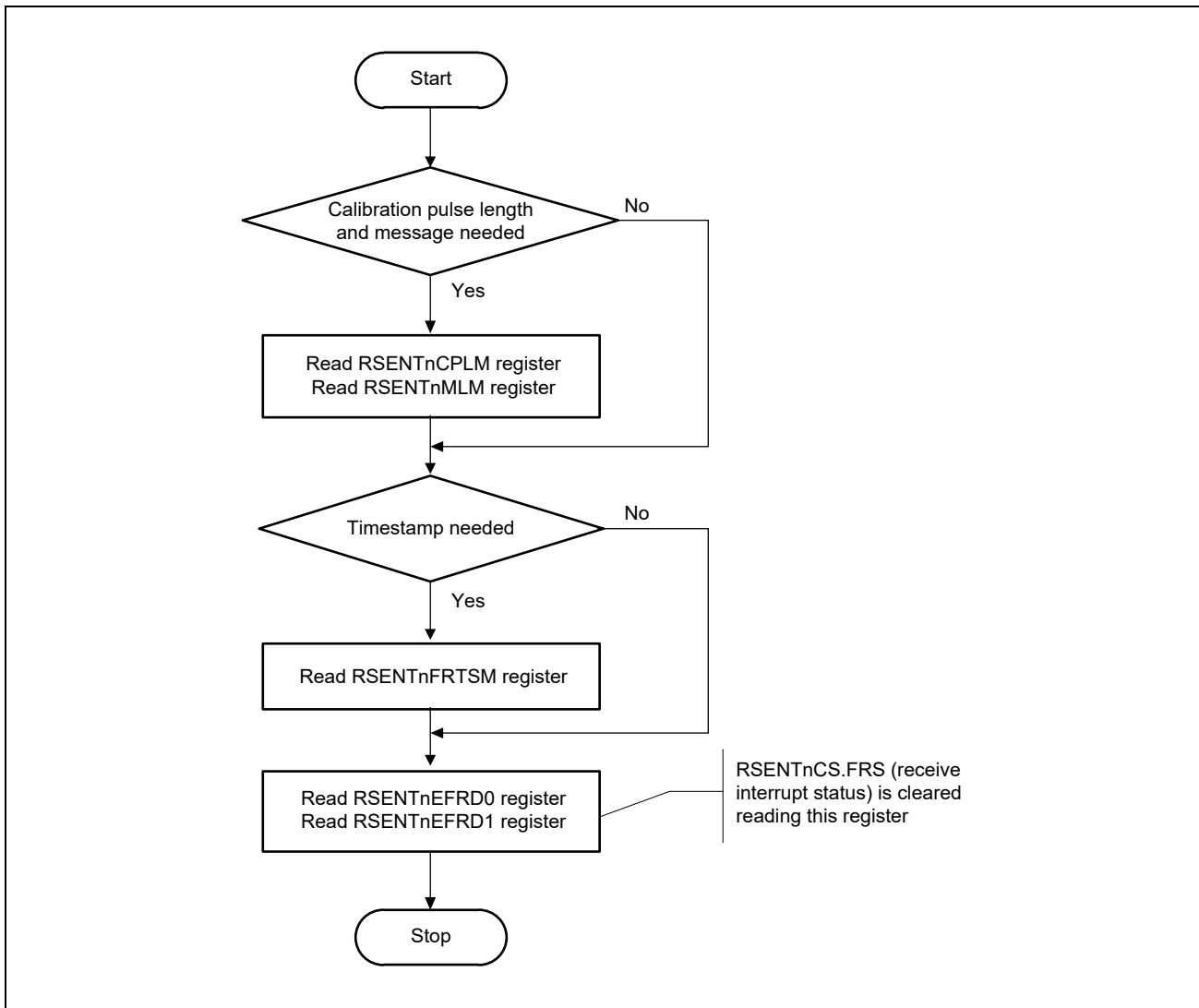


Figure 23.18 DMA Reception Flow

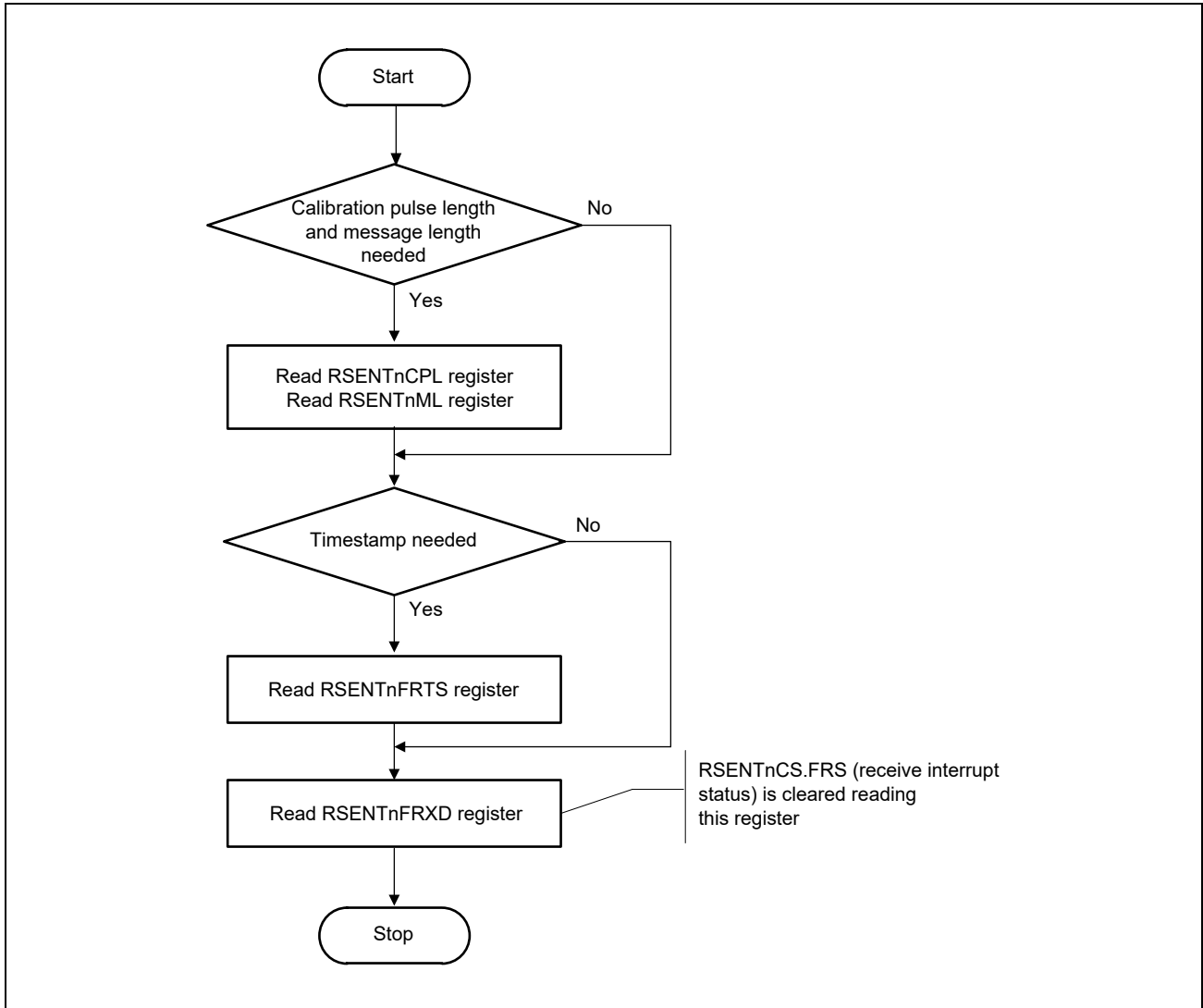


Figure 23.19 DMA Reception Flow (Data Nibble are 6 or Less)

In the software processing, when the transferred data set, the CPU should check the status of the transferred RSENTnFRXD.SNDM bit. If this bit is set to 1, then the user needs to read the slow channel receive buffer if needed.

(8) Error Flagging

Message lost errors (shown in RSENTnCS.SMS, RSENTnCS.FMS) are flagged when a new message's diagnostics pass before the previous message is read.

The SPC communication error (shown in RSENTnCS.NRS) is flagged when the CPU writes to RSENTnSPCT.TLL before/during response reception.

The update timings for fast channel reception errors (RSENTnCS.CVS, RSENTnCS.CLS, RSENTnCS.FNS, RSENTnCS.FES and RSENTnCS.FCS) and slow channel reception errors (RSENTnCS.SCS and RSENTnCS.SES) depends on the configuration of RSENTnCC.SPCE, RSENTnCC.FCM, RSENTnCC.PPC and RSENTnCC.PPTC. The update timings for each configuration are shown in **Table 23.37** and **Table 23.38**.

In case a nibble encoding error or calibration pulse length error is detected, message reception is terminated immediately. No further error flagging for this message is done. Message decoding starts again after a calibration pulse without calibration length error (RSENTnCS.CLS) is detected.

Table 23.37 Error Flag Set Timing when RSENTnCC.SPCE = 0

RSENTnCC.SPCE	0	0	0	0	0	0	0	0
RSENTnCC.FCM	0	0	0	0	1	1	1	1
RSENTnCC.PPC	0	0	1	1	0	0	1	1
RSENTnCC.PPTC	0	1	0	1	0	1	0	1
RSENTnCS.FCS	EC	X	EC	IM	IM	X	IM	IM
RSENTnCS.FES	EC	X	EC	IM	IM	X	IM	IM
RSENTnCS.FNS	EC	X	EC	—	—	X	—	—
RSENTnCS.SCS	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.SES	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.CLS	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.CVS	EC	X	EC	—	EC	X	EC	—

Note: EC: End of calibration pulse
 IM: Immediately when detected
 —: Not set
 X: Invalid configuration

NOTE

When RSENTnCC.PPC and RSENTnCC.PPTC are set to 1, the mode is pause pulse with fixed message length.

In this mode, this diagnostic can be used as the receiver does not need to wait for the next calibration pulse to diagnose the current received frame, so RSENTnCS.FNS and RSENTnCS.CVS are not set.

Table 23.38 Error Flag Set Timing when RSENTnCC.SPCE = 1

RSENTnCC.SPCE	1	1	1	1	1	1	1	1
RSENTnCC.FCM	0	0	0	0	1	1	1	1
RSENTnCC.PPC	0	0	1	1	0	0	1	1
RSENTnCC.PPTC	0	1	0	1	0	1	0	1
RSENTnCS.FCS	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.FES	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.FNS	—	X	—	—	—	X	—	—
RSENTnCS.SCS	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.SES	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.CLS	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.CVS	—	X	—	—	—	X	—	—

Note: EC: End of calibration pulse
IM: Immediately when detected
—: Not set
X: Invalid configuration

NOTE

In case the sensor stops communication, no buffer update or status update for last message takes place. The SW should take care of this by timeout checks.

SPC mode, started by the SPC trigger from the starting point, is a mode for one-frame communication.

The Error flags are immediately set when they are detected. Therefore, RSENTnCS.FNE and RSENT-nCS.CVS are not set.

When a transition to OPERATION IDLE is configured in RSENTnMDC.OMC and an error for the calibration or the fast channel reception is detected in the message in which the mode transition was requested, the error is not flagged and the message is aborted.

In case of a fast channel encoding error or a calibration pulse length error, the OPERATION IDLE mode is entered immediately.

In case of a fast channel nibble count error, fast channel CRC error, or fast channel calibration pulse variation error, the OPERATION IDLE state is entered at the end of the next STATUS/COM nibble.

RSENTnCS.FNS is only set after a valid calibration pulse was detected and all following nibbles have a valid length (≥ 12 ticks and ≤ 27 ticks) or no nibble was received between two valid calibration pulses.

RSENTnCS.FES is only set if the nibble with an encoding error occurred in the communication and status nibble, CRC nibble or in one of the expected data nibbles.

If SPC is enabled (RSENTnCC.SPCE = 1), RSENTnCS.CLS is set if a calibration pulse was expected but the pulse length does not meet the calibration pulse range. If SPC is disabled (RSENTnCC.SPCE = 0), RSENTnCS.CLS is set only after a valid calibration pulse has been received and a calibration pulse was expected but the pulse length does not meet the calibration pulse range.

During re-synchronization, additional error flags might be set, which does not affect the reception of the following frame.

23.4.4 SPC Function

RSENT supports an extension of the SAE J2716 specification known as SPC.

The user can enable or disable the SPC extension by setting the RSENTnCC.SPCE bit.

In SPC mode, the RX line will be driven low to initiate SENT message transmission. This can be realized by controlling an external transistor by the RSENTnSPCO pin.

The user can configure the polarity of the RSENTnSPCO port.

The text below describes the behavior of the RSENTnSPCO port with the default settings of RSENTnCC.SOPC. When the default value of RSENTnCC.SOPC is changed, the polarity is inverted.

The RX line will be held low for a configured length of tick time specified in the RSENTnSPCT.TLL bits. The tick time is configured with the RSENTnBRP.TTI bits and the RSENTnBRP.TTF bits, which are equal to the transmission tick time. For details, see **Section 23.4.2.2(2), RX and SPC Tick Settings**.

Figure 23.20 and **Figure 23.21** show an example of the SPC circuit connection.

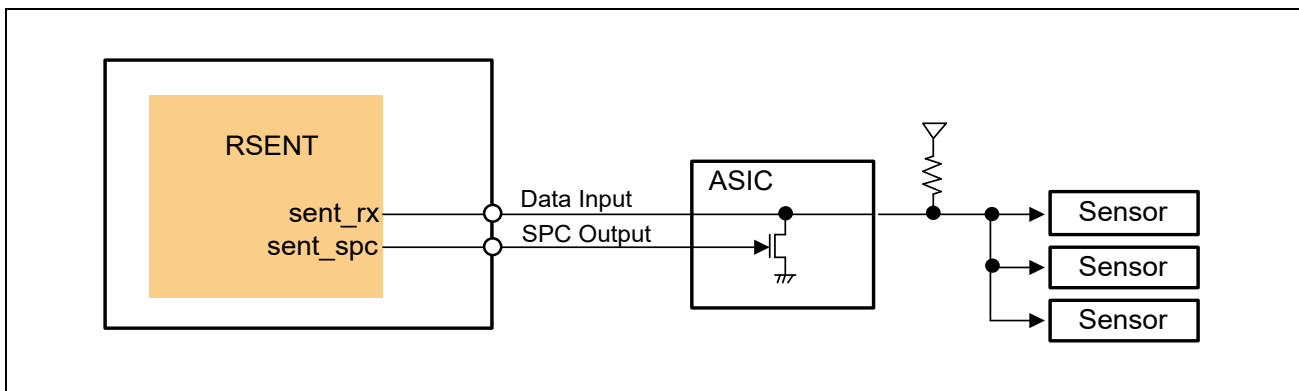


Figure 23.20 Example of the Circuit for two-wire interface

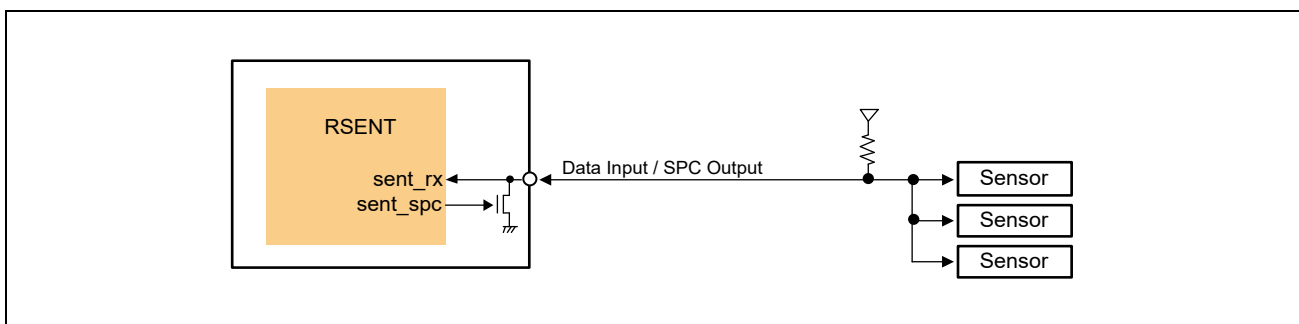


Figure 23.21 Example of the Circuit for One-wire interface

In a single sensor system, this function can be used to trigger data transmission from the sensor. Further data can be sent to the sensor by varying the trigger pulse length. In a multi sensor system, this function can be used to address a dedicated sensor and request a data transmission.

Once RSENT SPC initialization is complete, transmission can be triggered by writing the trigger pulse width to the RSENTnSPCT.TLL register. When a transmission is triggered, the trigger pulse with the configured length is sent. Then a frame reception is expected. After frame reception was done, a new trigger pulse can be sent.

Writing to RSENTnSPCT.TLL requests a SPC trigger transmission. After writing to RSENTnSPCT.TLL, the CPU should read RSENTnCS.NRS to check whether the previous request was completed or not.

The CPU has to wait for more than 10 pclk cycles before RSENTnCS.NRS is set.

In case RSENTnCS.NRS is set, no SPC trigger is sent and any potentially ongoing reception at this time is aborted. The CPU should clear RSENTnCS.NRS by writing 1 to RSENTnCSC.NRC. The CPU can write again to RSENTnSPCT.TLL to request a SPC trigger transmission.

In case RSENTnCS.NRS is not set, the CPU should set a reception timeout counter in software. If a reception occurs before the timeout counter elapses, the user should process the received slow and fast channel data as shown in the fast channel reception flow (**Figure 23.14**) and slow channel reception flow (**Figure 23.17**).

When the timeout counter elapses without any successful reception, the addressed sensor seems not to send any valid response. The CPU should analyze the RSENTnCS register to analyze the reason for no successful reception. A new request can be made considering that when RSENTnCS.NRS gets set, no SPC trigger is sent.

The purpose of the timeout function is to define a timeout window for response reception in software.

Figure 23.22 shows the transmission flow with a timeout function implemented in software. The timeout function is optional and can be omitted if not needed.

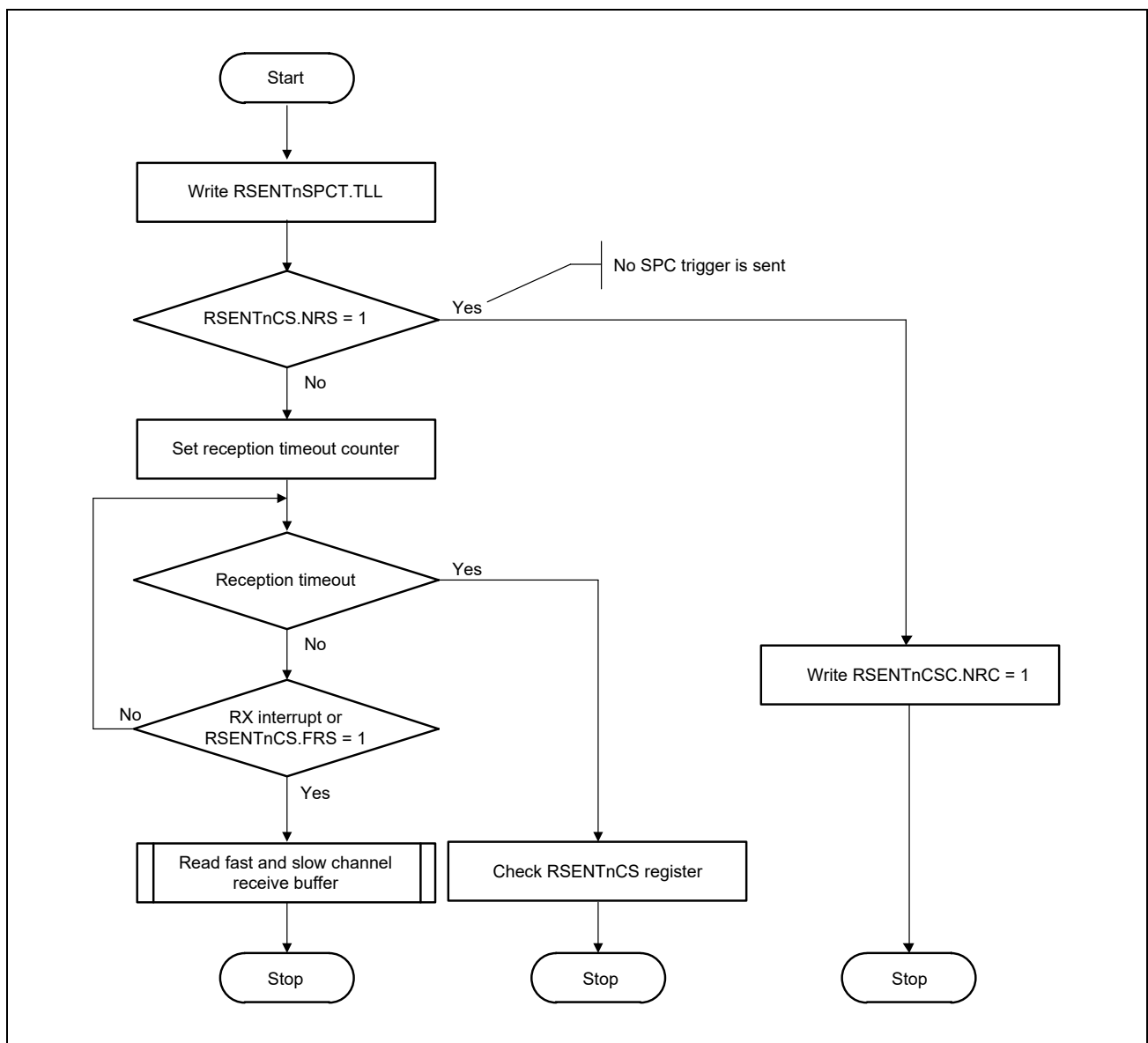


Figure 23.22 Transmission Flow

23.4.4.1 Multiplexing of the RSENTnRX and RSENTnSPCO Pin Functions

The RSENTnRX input pin and RSENTnSPCO output pin functions are assigned to the same multiplexed pin.

Two functions can be used on the pin, when it is set for an N-ch open-drain configuration.

Figure 23.23 shows an example of the circuit for connection to an external transistor.

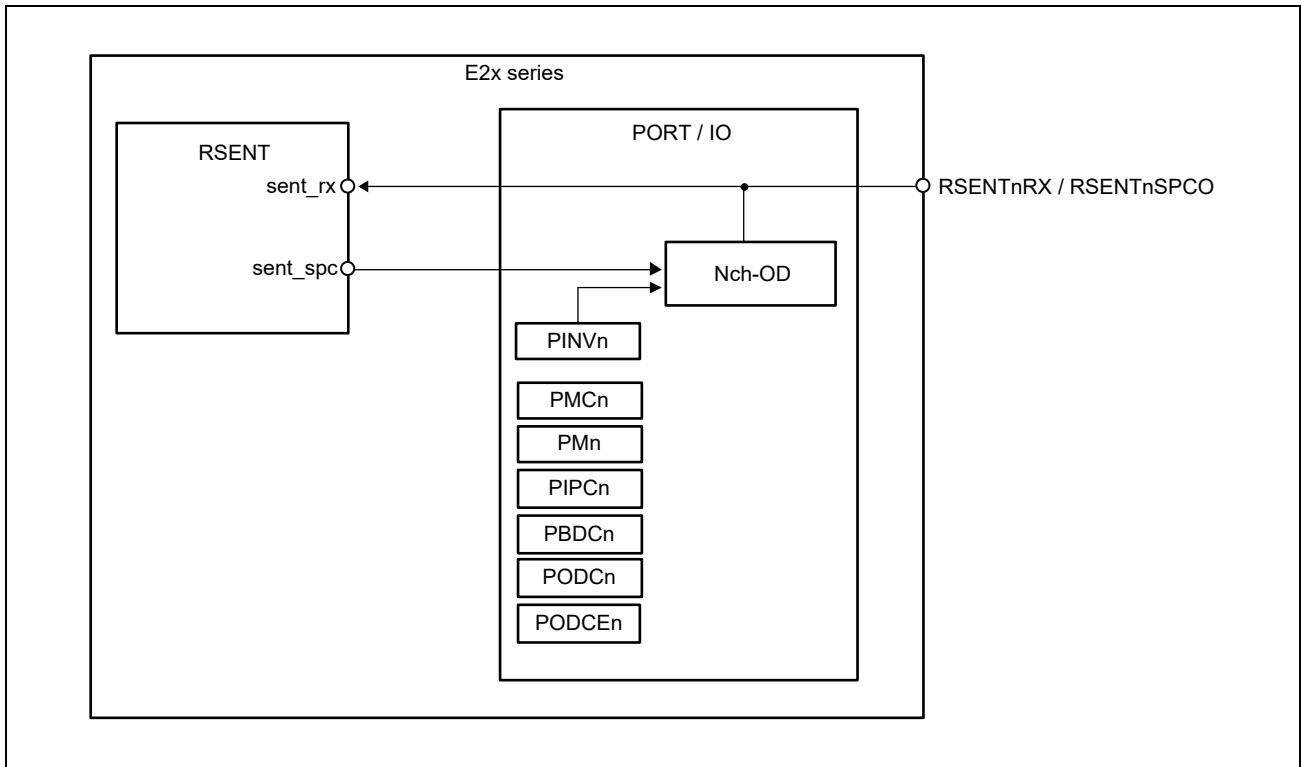


Figure 23.23 Multiplexing of the RSENTnRX and RSENTnSPCO Pin Functions

Connect the sent_spc signal, which is an RSENT output signal, to an N-ch open drain I/O buffer.

RSENT is able to control the polarity of setnt_spc output and the polarity of the pin output by setting the RSENTnCC.SOPC bit and the PINVn register of the I/O buffer, respectively. These registers must be set in an appropriate combination.

Table 23.39 lists the register settings when the RSENTnRX and RSENTnSPCO pin functions share the same pin.

Table 23.39 Setting for Multiplexing of the RSENTnRX and RSENTnSPCO Pin Functions

Register Name		Setting Value	Description
PMCn	Port mode control register	1	Alternative mode
PMn	Port mode register	0	Output mode (output enabled)
PIPCn	Port IP control register	0	I/O mode is controlled by PMn.PMn_m (software or hardware I/O control)
PBDCn	Port bi-direction control register	1	Bi-direction mode is enabled
PODCn	Port open-drain control register	1	N-ch open drain (PODCn_m = 1 or PODCEn_m = 0)
PODCEn	Port open-drain expansion register	0	N-ch open drain (PODCn_m = 1 or PODCEn_m = 0)
PINVn	Port output level inversion register	0	The pin output level is not inverted (active low when RSENTnCC.SOPC = 1)
		1	The pin output level is inverted (active high when RSENTnCC.SOPC = 0)
PFCn	Port function control register	—	Specify an alternative function of the pins. For details, see Table 2.9, Outline of Alternative Mode Selection (PMCn.PMCn_m = 1) .
PFCEn	Port function control expansion register	—	
PFCAEn	Port function control additional expansion register	—	
PFCEAEn	Port function control extra additional expansion register	—	

23.4.5 Interrupts and Checks

RSENT provides two interrupt lines.

The successful fast channel receive interrupt notifies the CPU that the fast channel receive buffer was updated and is holding a set of valid received data. Also, the reception status bit is set (RSENTnCS.FRS).

The status interrupt notifies the CPU that at least one of the error flags or message lost flags in the RSENTnCS register is set.

Whether or not a status flag in the RSENTnCS register contributes to the generation of an interrupt event can be set individually.

The execution of CRC checks can be disabled for the slow channel and fast channel individually. In case a check is disabled, the CRC of the received message is not checked and the related error flag is never set.

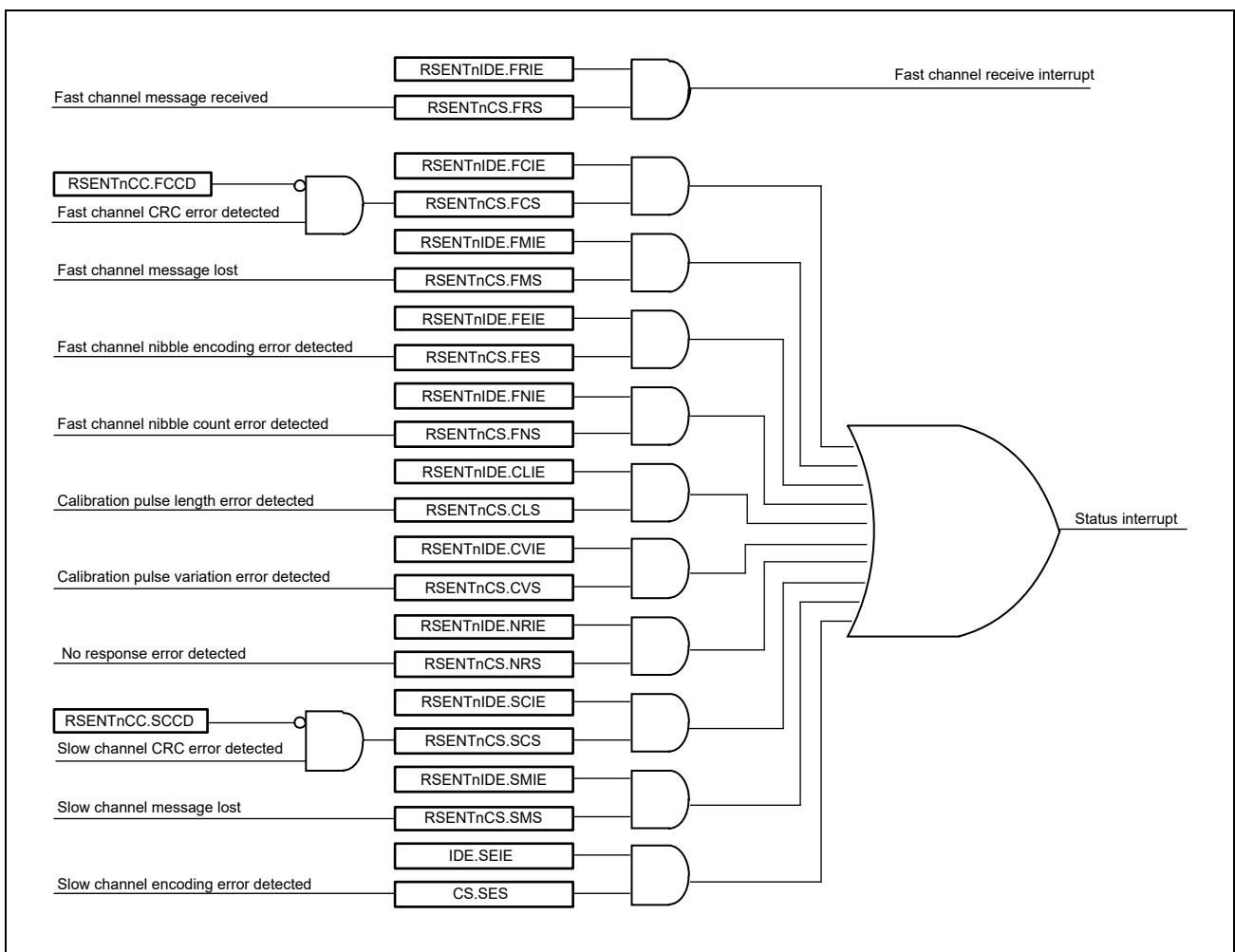


Figure 23.24 Interrupt Structure

Table 23.40 gives an overview of the relationship between set status flags and the buffer update.

Table 23.40 Status Flag Influence to Receive Buffer Behavior

RSENTnCS	Fast Channel Receive Buffer	Slow Channel Receive Buffer
FRS	Updated	Updated if all status and communication nibbles of slow channel messages are received and RSENTnCS.SES = 0 and RSENTnCS.SCS = 0
FCS	Not updated	Not impacted. Fast channel CRC does not cover the communication and status nibble
FMS	Background buffer overwritten	Not impacted
FES	Not updated	Receive process aborted. Search for new start condition
FNS	Not updated	Receive process aborted. Search for new start condition
CLS	Not updated	Receive process aborted. Search for new start condition
CVS	Not updated	Receive process aborted. Search for new start condition
NRS	Not updated	Receive process aborted. Search for new start condition
SCS	Not impacted	Not updated
SMS	Not impacted	Message lost
SES	Not impacted	Receive process aborted. Search for new start condition

Section 24 Ethernet Controller (ETNC)

This section describes the Ethernet Controller module (ETNC).

This module has an Ethernet Controller (ETHERC) and Direct Memory Access Controller (EDMAC).

The first subsection describes all the characteristics specific to ETNC such as channels, register base addresses, and input/output signal names.

The second and subsequent subsections describe the functions of ETNC.

24.1 Features of ETNC

24.1.1 Number of Units and Channels

The ETNC includes one physical channel. This physical channel features MAC layer interface ports through which the MCU can be connected to the physical layer LSI (PHY-LSI), allowing transmission and reception of frames compliant with the Ethernet/IEEE802.3 standard.

Table 24.1 Number of Channels

Product Name	RH850/E2x-FCC1	
	373 pins	292 pins
Number of Channels	1	1
Name	ETNCn (n = 0)	

Table 24.2 Number of Channels

Product Name	RH850/E2M	
	373 pins	292 pins
Number of Channels	1	1
Name	ETNCn (n = 0)	

Table 24.3 Indices

Index	Description
n	Throughout this section, chip select areas are identified by "n" (n = 0). For example, the ETHERC Status Register is described as ETNCnECSR.

24.1.2 Register Addresses

The ETNC_n register address is expressed as an offset from the base address <ETNC_n_base>.

The following table shows the base address <ETNC_n_base> of ETNC_n.

Table 24.4 Register Base Address

Base Address Name	Base Address	Bus Group
<ETNC0_base>	1002 4000 _H	H-Bus Group 2

24.1.3 Clock Supply

The ETNC_n clock supplies are shown in the following table.

Table 24.5 Clock Supply

Unit Name	Unit Clock Name	Clock Supply Name
ETNC _n	PCLKA (register access clock)	CLK_HBUS
	MII_TX_CLK (MII Transfer clock (25 MHz))	MII_TX_CLK (external input clock)
	MII_RX_CLK (MII Receipt clock (25 MHz))	MII_RX_CLK (external input clock)
	REF50CK (RMII Reference clock (50 MHz))	REF50CK (external input clock)

24.1.4 Interrupt Requests

The ETNC_n interrupt requests are listed in the following table.

Table 24.6 Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Description	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
INTETNC0	INTETNC _n (n=0)	Ethernet interrupt	485	—	—

24.1.5 Reset Source

The ETNC_n reset source is listed in the following table. ETNC_n is initialized by the following reset source.

Table 24.7 Reset Source

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
ETNC _n	All registers	✓	✓	✓	✓	—	—

24.1.6 External Input/Output Signals

The following table shows the input/output pins of ETHERC (for the MII/ RMII mode).

Table 24.8 and **Table 24.9** list the ETHERC I/O pins and **Table 24.10** lists the other PHY-LSI pins.

Table 24.8 Input/Output Pins of ETHERC in Operation MII Mode

Operating Mode	Pin Name	I/O	Description
MII mode	MII_TX_CLK	Input	MII Transmit clock Timing reference signal for outputting the MII_TX_EN, MII_TXD3 to MII_TXD0, and MII_TX_ER signals.
	MII_RX_CLK	Input	MII Receive clock Timing reference signal for inputting the MII_RX_DV, MII_RXD3 to MII_RXD0, and MII_RX_ER signals.
	MII_TX_EN	Output	MII Transmit data enable output (valid) This signal indicates that valid transmit data has been output to pins MII_TXD3 to MII_TXD0.
	MII_TXD3 MII_TXD2 MII_TXD1 MII_TXD0	Output	4-bit MII transmit data output
	MII_TX_ER*1	Output	MII Transmit error output This signal notifies the PHY-LSI that an error occurred during transmission.
	MII_RX_DV	Input	MII Receive data enable input (valid) This signal indicates that valid receive data has been input from pins MII_RXD3 to MII_RXD0.
	MII_RXD3 MII_RXD2 MII_RXD1 MII_RXD0	Input	4-bit MII receive data
	MII_RX_ER	Input	MII Receive error input This signal indicates that there is an error in a frame that is being transferred from the PHY-LSI to the ETHERC.
	MII_CRD	Input	MII Carrier detection signal
	MII_COL	Input	MII Collision detection signal
	MDC	Output	PHY Management data clock Reference clock signal for transmitting management data with the PHY-LSI .
	MDIO	I/O	PHY Management data I/O Bidirectional data signal for exchanging management data with the PHY-LSI.

Note: MII signals are compliant with IEEE802.3u.

Note 1. Always output low level.

Table 24.9 Input/Output Pins of ETHERC in Operation RMII Mode

Operating Mode	Pin Name	I/O	Description
RMII mode	REF50CK	Input	RMII Reference clock Timing reference signal for pins RMII_TX_EN, RMII_TXD1 to RMII_TXD0, RMII_CRS_DV, RMII_RXD1 to RMII_RXD0, and RMII_RX_ER.
	RMII_TX_EN	Output	RMII Transmit data enable output (valid) This signal indicates that valid transmit data has been output to pins RMII_TXD1 to RMII_TXD0.
	RMII_TXD1 RMII_TXD0	Output	2-bit RMII transmit data output
	RMII_CRS_DV	Input	RMII Carrier sense detection signal/receive data valid This signal indicates that valid receive data has been input from pins RMII_RXD1 to RMII_RXD0.
	RMII_RXD1 RMII_RXD0	Input	2-bit RMII receive data
	RMII_RX_ER	Input	RMII Receive error This signal indicates that there is an error in a frame that is being transferred from the PHY-LSI to the ETHERC.
	MDC	Output	PHY Management data clock Reference clock signal for transmitting management data with the PHY-LSI
	MDIO	I/O	PHY Management data I/O Bidirectional data signal for exchanging management data with the PHY-LSI.

Note: MII signals are compliant with IEEE802.3u.

Table 24.10 Other PHY-LSI Pins

Operating Mode	Pin Name	I/O	Description
Other signals	LINKSTA	Input	PHY Link status input from the PHY-LSI
	EXOUT	Output	External output pin
	WOL	Output	Wake-On-LAN. This signal indicates that a Magic Packet has been received.

Figure 24.1 and Figure 24.2 show examples of connecting the MCU to the PHY-LSI.

Figure 24.1 shows the connection with PHY-LSI in MII mode.

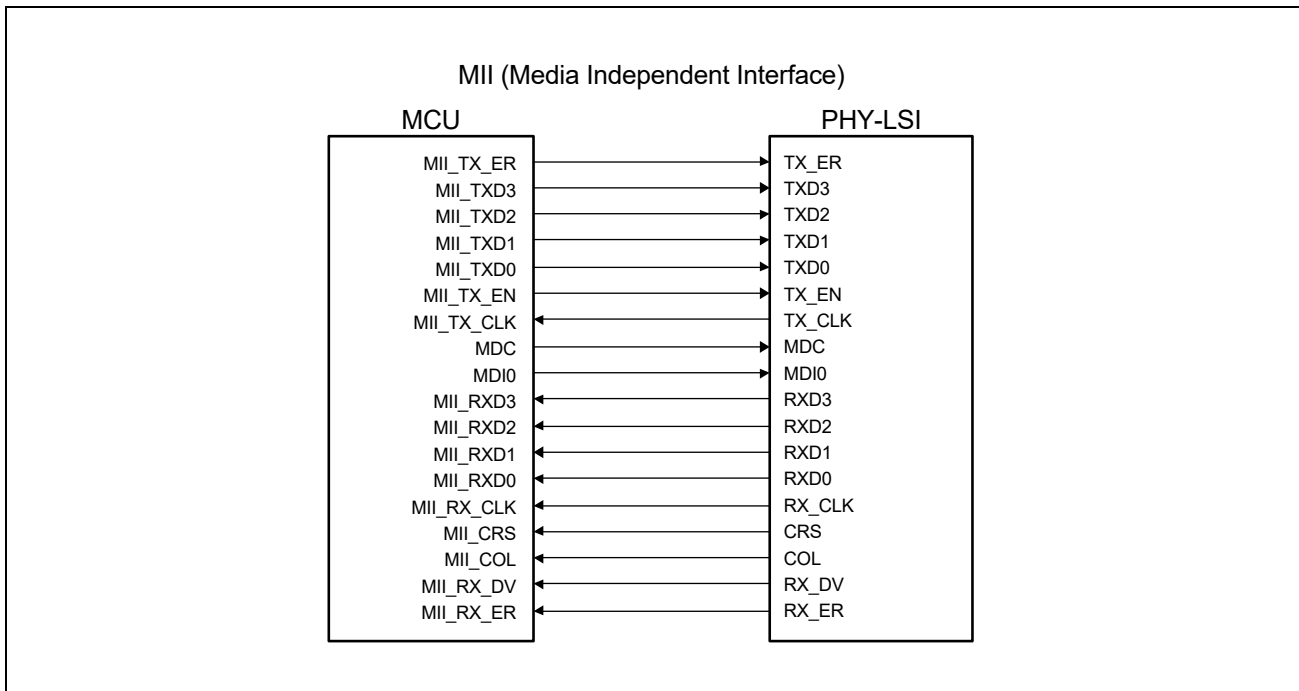


Figure 24.1 Example of Connection with the PHY-LSI in MII Mode

Figure 24.2 shows the connection with PHY-LSI in RMII mode.

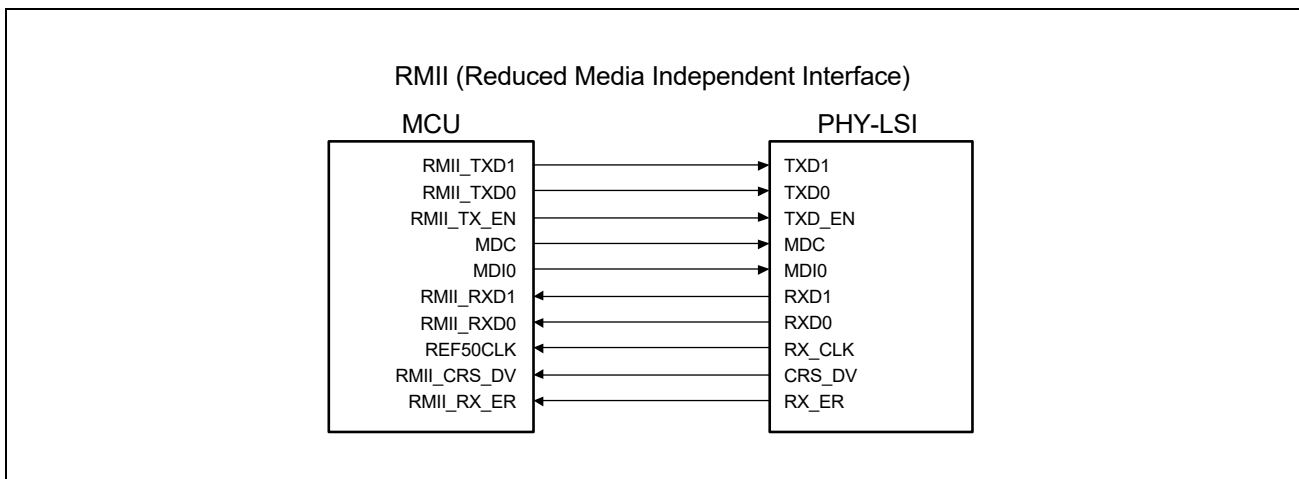


Figure 24.2 Example of Connection with PHY-LSI in RMII Mode

24.2 Overview

24.2.1 Functional Overview

This MCU has a one-channel Ethernet controller (ETHERC) compliant with the Ethernet or IEEE802.3 Media Access Control (MAC) layer protocol. The ETHERC channel features an MAC layer interface through which the MCU can be connected to the physical layer LSI (PHY-LSI), allowing transmission and reception of frames compliant with the Ethernet/IEEE802.3 standard. The ETHERC is connected to the Ethernet Controller's Direct Memory Access Controller (EDMAC), so data can be transferred without using the CPU.

The EDMAC controls most of the transmit/receive buffer management operations for communications. This reduces the load on the CPU and allows efficient data transmission and reception. The data transfers are controlled according to the information (referred to as descriptors) on the memory.

Table 24.11 lists the ETHERC and EDMAC specifications. **Figure 24.3** shows the ETHERC and EDMAC configuration.

Figure 24.4 shows the configuration of descriptors and transmit/receive buffers on the memory.

Table 24.11 ETHERC and EDMAC Specifications

Item	Description
Protocol	Flow control compliant with IEEE802.3x
Data transmission/reception	<ul style="list-style-type: none"> Frames compliant with the Ethernet/IEEE802.3 standard can be transmitted and received.
Bit rate	Supports 10 Mbps and 100 Mbps
Operation modes	Supports full-duplex and half-duplex modes
Interfaces	Media Independent Interface (MII), Reduced Media Independent Interface (RMII), compliant with the IEEE802.3u standard
Functions	<ul style="list-style-type: none"> Magic Packet™¹ detection, Wake-On-LAN (WOL) signal output
Low power consumption function	The EDMAC can be set to the module-stop state to reduce power consumption.

Note 1. Magic Packet is a trademark of Advanced Micro Devices, Inc.

24.2.2 Block Diagram

The following figures show block diagrams of the ETNC module.

The ETHERC can transfer the transmitted or received Ethernet frame data to and from the transmit/receive buffer in the memory at high speed using a dedicated direct memory access controller (EDMAC).

Figure 24.3 shows the ETHERC and EDMAC configuration.

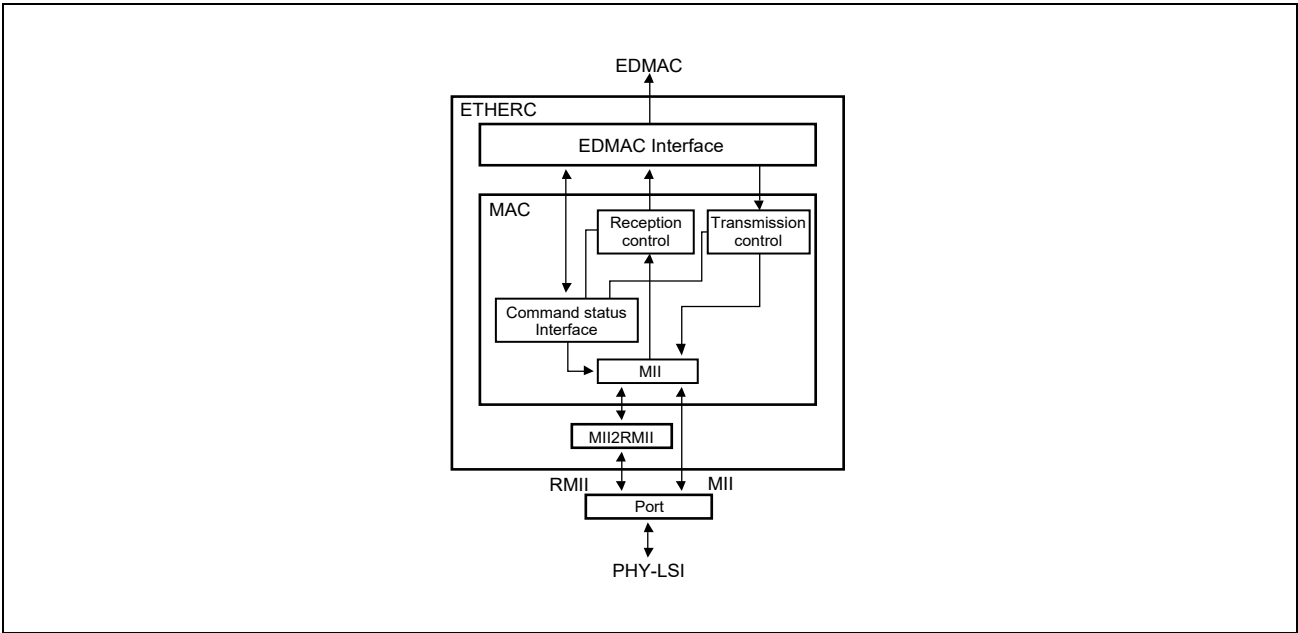


Figure 24.3 ETHERC and EDMAC Configuration

Figure 24.4 shows the Descriptors and Transmit/Receive Buffers on the Memory:

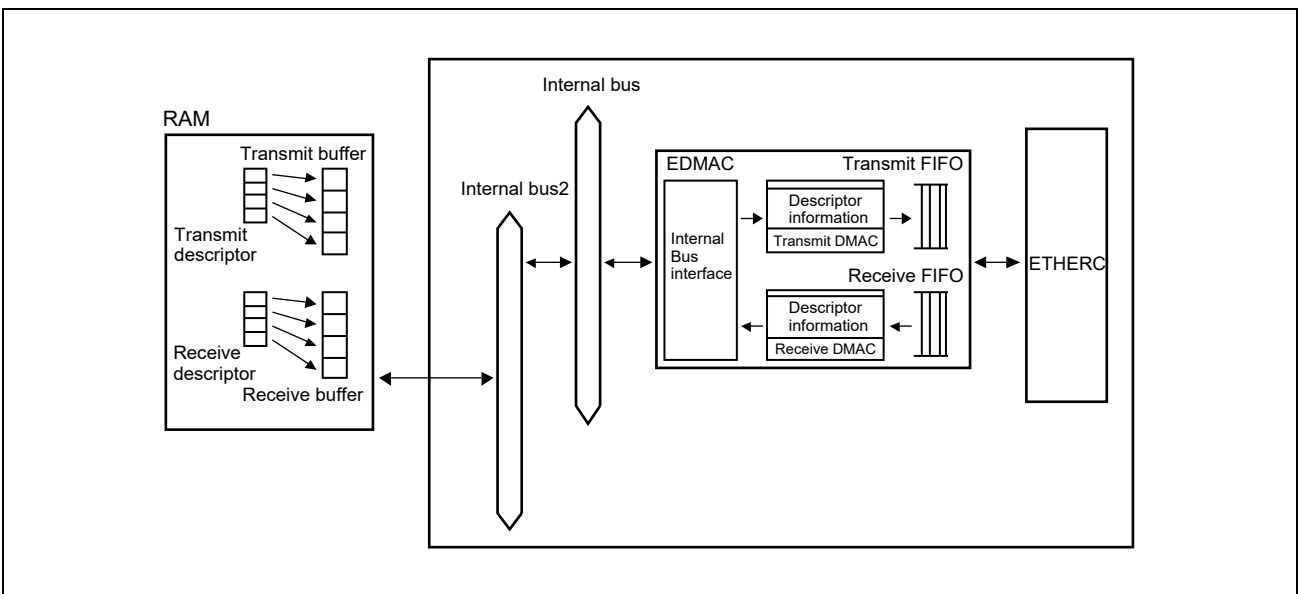


Figure 24.4 Configuration of Descriptors and Transmit/Receive Buffers on the Memory

24.3 Registers

24.3.1 List of Registers

Table 24.12 shows the registers of ETHERC.

Table 24.12 List of Registers of ETHERC

Module Name	Register Name	Symbol	Address	Access Size	Access Protection
ETNCn	ETHERC Mode Register	ETNCnECMR	<ETNCn_base> + 0100 _H	32	—
	Receive Frame Maximum Length Register	ETNCnRFLR	<ETNCn_base> + 0108 _H	32	—
	ETHERC Status Register	ETNCnECSR	<ETNCn_base> + 0110 _H	32	—
	ETHERC Interrupt Enable Register	ETNCnECSIPR	<ETNCn_base> + 0118 _H	32	—
	PHY Interface Register	ETNCnPIR	<ETNCn_base> + 0120 _H	32	—
	PHY Status Register	ETNCnPSR	<ETNCn_base> + 0128 _H	32	—
	Random Number Generation Counter Limit Setting Register	ETNCnRDMLR	<ETNCn_base> + 0140 _H	32	—
	Interpacket Gap Register	ETNCnIPGR	<ETNCn_base> + 0150 _H	32	—
	Automatic PAUSE Frame Register	ETNCnAPR	<ETNCn_base> + 0154 _H	32	—
	Manual PAUSE Frame Register	ETNCnMPR	<ETNCn_base> + 0158 _H	32	—
	Received PAUSE Frame Counter	ETNCnRFCF	<ETNCn_base> + 0160 _H	32	—
	PAUSE Frame Retransmit Count Setting Register	ETNCnTPAUSER	<ETNCn_base> + 0164 _H	32	—
	PAUSE Frame Retransmit Counter Register	ETNCnTPAUSECR	<ETNCn_base> + 0168 _H	32	—
	Broadcast Frame Receive Count Setting Register	ETNCnBCFRR	<ETNCn_base> + 016C _H	32	—
	MAC Address Upper Bit Register	ETNCnMAHR	<ETNCn_base> + 01C0 _H	32	—
	MAC Address Lower Bit Register	ETNCnMALR	<ETNCn_base> + 01C8 _H	32	—
	Transmit Retry Over Counter Register	ETNCnTROCR	<ETNCn_base> + 01D0 _H	32	—
	Late Collision Detect Counter Register	ETNCnCDCR	<ETNCn_base> + 01D4 _H	32	—
	Lost Carrier Counter Register	ETNCnLCCR	<ETNCn_base> + 01D8 _H	32	—
	Carrier Not Detected Counter Register	ETNCnCNDCR	<ETNCn_base> + 01DC _H	32	—
	CRC Error Frame Receive Counter Register	ETNCnCEFCR	<ETNCn_base> + 01E4 _H	32	—
	Frame Receive Error Counter Register	ETNCnFRECR	<ETNCn_base> + 01E8 _H	32	—
	Too-Short Frame Receive Counter Register	ETNCnTSFRCR	<ETNCn_base> + 01EC _H	32	—
	Too-Long Frame Receive Counter Register	ETNCnTLFRCR	<ETNCn_base> + 01F0 _H	32	—
Received Alignment Error Frame Counter Register	ETNCnRFCR	<ETNCn_base> + 01F4 _H	32	—	
Multicast Address Frame Receive Counter Register	ETNCnMAFCR	<ETNCn_base> + 01F8 _H	32	—	

Table 24.13 shows the registers of EDMAC.

Table 24.13 List of Registers of EDMAC

Module Name	Register Name	Symbol	Address	Access Size	Access Protection
ETNCn	EDMAC Mode Register	ETNCnEDMR	<ETNCn_base> + 0000 _H	32	—
	EDMAC Transmit Request Register	ETNCnEDTRR	<ETNCn_base> + 0008 _H	32	—
	EDMAC Receive Request Register	ETNCnEDRRR	<ETNCn_base> + 0010 _H	32	—
	Transmit Descriptor List Start Address Register	ETNCnTDLAR	<ETNCn_base> + 0018 _H	32	—
	Receive Descriptor List Start Address Register	ETNCnRDLAR	<ETNCn_base> + 0020 _H	32	—
	ETHERC/EDMAC Status Register	ETNCnEESR	<ETNCn_base> + 0028 _H	32	—
	ETHERC/EDMAC Status Interrupt Enable Register	ETNCnEESIPR	<ETNCn_base> + 0030 _H	32	—
	ETHERC/EDMAC Transmit / Receive Status Copy Enable Register	ETNCnTRSCER	<ETNCn_base> + 0038 _H	32	—
	Missed-Frame Counter Register	ETNCnRMFCR	<ETNCn_base> + 0040 _H	32	—
	Transmit FIFO Threshold Register	ETNCnTFTR	<ETNCn_base> + 0048 _H	32	—
	FIFO Depth Register	ETNCnFDR	<ETNCn_base> + 0050 _H	32	—
	Receive Method Control Register	ETNCnRMCR	<ETNCn_base> + 0058 _H	32	—
	Transmit FIFO Underflow Counter	ETNCnTFUCR	<ETNCn_base> + 0064 _H	32	—
	Receive FIFO Overflow Counter	ETNCnRFOCR	<ETNCn_base> + 0068 _H	32	—
	Independent Output Signal Setting Register	ETNCnIOSR	<ETNCn_base> + 006C _H	32	—
	Flow Control Start FIFO Threshold Setting Register	ETNCnFCFTR	<ETNCn_base> + 0070 _H	32	—
	Receive Data Padding Insert Register	ETNCnRPADIR	<ETNCn_base> + 0078 _H	32	—
	Transmit Interrupt Setting Register	ETNCnTRIMD	<ETNCn_base> + 007C _H	32	—
	Receive Buffer Write Address Register	ETNCnRBWAR	<ETNCn_base> + 00C8 _H	32	—
	Receive Descriptor Fetch Address Register	ETNCnRDFAR	<ETNCn_base> + 00CC _H	32	—
Transmit Buffer Read Address Register	ETNCnTBRAR	<ETNCn_base> + 00D4 _H	32	—	
Transmit Descriptor Fetch Address Register	ETNCnTDFAR	<ETNCn_base> + 00D8 _H	32	—	

24.3.2 ETHERC Registers

24.3.2.1 ETNCnECMR — ETHERC Mode Register

The ETNCnECMR register controls the ETHERC operations.

Set bits in the ETNCnECMR register, excluding the TE and RE bits, during initialization after a reset. When rewriting this register other than during the initialization process, set the ETNCnEDMR.SWR bit to 1 to reset the EDMAC and ETHERC, and then set this register again.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	TPC	ZPF	PFR	RXF	TXF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PRCEF	—	—	MPDE	—	—	RE	TE	—	ILB	RTM	DM	PRM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W

Table 24.14 ETNCnECMR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 21	—	Reserved The read value is 0. The write value should be 0.
20	TPC	PAUSE Frame Transmit 0: PAUSE frame is transmitted even during a PAUSE period. 1: PAUSE frame is not transmitted during a PAUSE period.
19	ZPF	0 Time PAUSE Frame Enable 0: PAUSE frame with a pause_time parameter of 0 is not used. 1: PAUSE frame with a pause_time parameter of 0 is used.
18	PFR	PAUSE Frame Receive Mode 0: PAUSE frame is not transferred to the EDMAC. 1: PAUSE frame is transferred to the EDMAC.
17	RXF	Receive Flow Control Operating Mode 0: PAUSE frame detection is disabled. 1: PAUSE frame detection is enabled.
16	TXF	Transmit Flow Control Operating Mode 0: Automatic PAUSE frame transmission is disabled. (PAUSE frame is not automatically transmitted.) 1: Automatic PAUSE frame transmission is enabled. (PAUSE frame is automatically transmitted as required.)
15 to 13	—	Reserved The read value is 0. The write value should be 0.
12	PRCEF	CRC Error Frame Receive Mode 0: EDMAC is notified of a CRC error. 1: EDMAC is not notified of a CRC error.
11 and 10	—	Reserved The read value is 0. The write value should be 0.

Table 24.14 ETNCnECMR Register Contents (2/2)

Bit Position	Bit Name	Function
9	MPDE	Magic Packet Detection Enable 0: Magic Packet detection is disabled. 1: Magic Packet detection is enabled.
8 and 7	—	Reserved The read value is 0. The write value should be 0.
6	RE	Reception Enable 0: Receive function is disabled. 1: Receive function is enabled.
5	TE	Transmission Enable 0: Transmit function is disabled. 1: Transmit function is enabled.
4	—	Reserved The read value is 0. The write value should be 0.
3	ILB	Internal Loopback Mode 0: Normal data transmission or reception is performed. 1: Data is looped back in the ETHERC when full-duplex mode is selected.
2	RTM	Bit Rate 0: 10 Mbps 1: 100 Mbps
1	DM	Duplex Mode 0: Half-duplex mode 1: Full-duplex mode
0	PRM	Promiscuous Mode 0: Promiscuous mode is disabled. 1: Promiscuous mode is enabled.

0 Time PAUSE Frame Enable (ZPF Bit)

When the ZPF bit is 1, a PAUSE frame with a `pause_time` parameter of 0 is transmitted if the PAUSE frame transmit request is canceled before the PAUSE time of the previously transmitted PAUSE frame has elapsed. After the PAUSE frame with a `pause_time` parameter of 0 is received, the ETHERC is ready for transmission.

When the ZPF bit is 0, even if the PAUSE frame transmit request from the receive FIFO is canceled, the next PAUSE frame is not transmitted until the PAUSE time of the previously transmitted PAUSE frame has elapsed. When a PAUSE frame with a `pause_time` parameter of 0 is received, the PAUSE frame is discarded.

CRC Error Frame Receive Mode (PRCEF Bit)

When the PRCEF bit is set to 1, the EDMAC is not notified that a CRC error has occurred even if the error is detected in a receive frame. Accordingly, the ETNCnEESR.CERF flag and RFS0 bit in receive descriptor 0 (RD0) do not become 1.

Reception Enable (RE Bit)

When the RE bit is set to 1, the ETHERC receive function is enabled.

When the RE bit is set to 0, the receive function is disabled after the frame being processed is completely received.

Transmission Enable (TE Bit)

When the TE bit is set to 1, the ETHERC transmit function is enabled.

When the TE bit is set to 0, the transmit function is disabled after the frame being processed is completely transmitted.

Internal Loopback Mode (ILB Bit)

When the ILB bit is set to 1, transmit frames can be looped back in the MCU. Set the DM bit to 1 (full-duplex mode) to perform a loopback test.

Bit Rate (RTM Bit)

The RTM bit sets the bit rate when the RMII is selected.

Promiscuous Mode (PRM Bit)

When the PRM bit is set to 1, the ETHERC operates in promiscuous mode where all Ethernet frames are received. In promiscuous mode, the ETHERC receives all valid frames regardless of whether or not the address matches the destination address or broadcast address and regardless of the multicast bit setting.

24.3.2.2 ETNCnRFLR — Receive Frame Maximum Length Register

The ETNCnRFLR register sets the maximum frame length that can be received by the MCU. Set the length in bytes.

Do not rewrite this register while the ETNCnECMR.RE bit is 1 (receive function is enabled).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RFL[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.15 ETNCnRFLR Register Contents

Bit Position	Bit Name	Function
31 to 12	—	Reserved The read value is 0. The write value should be 0.
11 to 0	RFL[11:0]	Receive Frame Maximum Length The set value becomes the maximum frame length. The minimum value that can be set is 1,518 bytes, and the maximum value that can be set is 2,048 bytes. Values that are less than 1,518 bytes are regarded as 1,518 bytes, and values larger than 2,048 bytes are regarded as 2,048 bytes.

Receive Frame Maximum Length (RFL[11:0] Bits)

The RFL[11:0] bits set a frame length to be checked. When the number of bytes for fields from the destination address to the frame check sequence (FCS) of the received frame exceeds the RFL[11:0] bit value, the EDMAC is notified of a frame-too-long error.

When the received frame length exceeds the RFL[11:0] bit value, the excess data is discarded.

24.3.2.3 ETNCnECSR — ETHERC Status Register

The ETNCnECSR register indicates the status of the ETHERC.

When any flag in the ETNCnECSR register becomes 1 while the corresponding bit in the ETNCnECSIPR (ETHERC Interrupt Enable Register) register is 1 (interrupt is reported), the ETNCnECSR.ECI flag becomes 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	BFR	PSRTO	—	LCHNG	MPD	ICD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R/W*1	R/W*1	R/W*1

Note 1. Write 1 to clear the flag.

Table 24.16 ETNCnECSR Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved The read value is 0. The write value should be 0.
5	BFR	Continuous Broadcast Frame Reception Flag 0: Continuous reception of broadcast frames has not been detected. 1: Continuous reception of broadcast frames has been detected.
4	PSRTO	PAUSE Frame Retransmit Over Flag 0: PAUSE frame retransmit count has not reached the upper limit. 1: PAUSE frame retransmit count has reached the upper limit.
3	—	Reserved The read value is 0. The write value should be 0.
2	LCHNG	Link Signal Change Flag 0: Change in the LINKSTA signal has not been detected. 1: Change in the LINKSTA signal has been detected (high to low, or low to high).
1	MPD	Magic Packet Detect Flag 0: Magic Packet has not been detected. 1: Magic Packet has been detected.
0	ICD	False Carrier Detect Flag 0: PHY-LSI has not detected a false carrier on the line. 1: PHY-LSI has detected a false carrier on the line.

PAUSE Frame Retransmit Over Flag (PSRTO Flag)

The PSRTO flag indicates that the number of retransmissions has reached the value set in the TPAUSER register when retransmitting a PAUSE frame while automatic PAUSE frame transmission is enabled.

Link Signal Change Flag (LCHNG Flag)

The LCHNG flag indicates that LINKSTA signal input from the PHY-LSI has changed from high to low, or from low to high.

Refer to the PSR.LMON flag for the current link status.

False Carrier Detect Flag (ICD Flag)

The ICD flag indicates that the PHY-LSI has detected a false carrier on the line.

The ICD flag becomes 1 when a receive error signal shown in **Figure 24.10** is received from the PHY-LSI. Note that the information may not be correct when signals input from the PHY-LSI change faster than software recognizes the change. Check the timing of the PHY-LSI.

24.3.2.4 ETNCnECSIPR — ETHERC Interrupt Enable Register

The ETNCnECSIPR register selects whether or not to notify the EDMAC of the status indicated by the ETNCnECSR register. Each bit corresponds to the flag in the ETNCnECSR register that has the same bit number.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	BFSIPR	PSRTOIP	—	LCHNGIP	MPDIP	ICDIP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 24.17 ETNCnECSIPR Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved The read value is 0. The write value should be 0.
5	BFSIPR	Continuous Broadcast Frame Reception Interrupt Enable 0: Notification of continuous broadcast frame reception interrupt is disabled. 1: Notification of continuous broadcast frame reception interrupt is enabled.
4	PSRTOIP	PAUSE Frame Retransmit Over Interrupt Enable 0: Notification of PAUSE frame retransmit over interrupt is disabled. 1: Notification of PAUSE frame retransmit over interrupt is enabled.
3	—	Reserved The read value is 0. The write value should be 0.
2	LCHNGIP	LINK Signal Change Interrupt Enable 0: Notification of LINKSTA signal change interrupt is disabled. 1: Notification of LINKSTA signal change interrupt is enabled.
1	MPDIP	Magic Packet Detect Interrupt Enable 0: Notification of the Magic Packet detect interrupt is disabled. 1: Notification of the Magic Packet detect interrupt is enabled.
0	ICDIP	False Carrier Detect Interrupt Enable 0: Notification of the false carrier detect interrupt is disabled. 1: Notification of the false carrier detect interrupt is enabled.

24.3.2.5 ETNCnPIR — PHY Interface Register

The ETNCnPIR register is used to access registers in the PHY-LSI via the MII or RMI. The management clock and management data are controlled by software.

Refer to **Section 24.4.1.4, Accessing MII/RMII Registers** for details on accessing MII and RMII registers.

Value after reset: 0000 000X_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	MDI	MDO	MMD	MDC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	X	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 24.18 ETNCnPIR Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved The read value is 0. The write value should be 0.
3	MDI	MII/RMII Management Data-In This bit indicates the level of the MDIOpin. The write value should be 0.
2	MDO	MII/RMII Management Data-Out The MDO bit value is output from the MDIOpin when the MMD bit is 1 (write). The value is not output when the MMD bit is 0 (read).
1	MMD	MII/RMII Management Mode 0: Read 1: Write
0	MDC	MII/RMII Management Data Clock The MDC bit value is output from the MDCpin to supply the management data clock to the MII or RMII.

24.3.2.6 ETNCnPSR — PHY Status Register

The ETNCnPSR register is used to monitor interface signals from the PHY-LSI.

Value after reset: 0000 000X_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LMON
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.19 ETNCnPSR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved The read value is 0. The write value should be 0.
0	LMON	LINKSTA Pin Status Flag The link status can be read by connecting the link signal output from the PHY-LSI to the LINKSTA pin. For details on the polarity, refer to the specifications of the connected PHY-LSI.

24.3.2.7 ETNCnRDMLR — Random Number Generation Counter Limit Setting Register

The ETNCnRDMLR register sets the maximum value for the counter used in the random number generator.

Do not rewrite this register while the ETNCnECMR.TE bit is 1 (transmit function is enabled) or while the ETNCnECMR.RE bit is 1 (receive function is enabled).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RMD[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.20 ETNCnRDMLR Register Contents

Bit Position	Bit Name	Function
31 to 20	—	Reserved The read value is 0. The write value should be 0.
19 to 0	RMD[19:0]	Random Number Generation Counter 00000 _H : Normal operation 00001 _H to FFFFF _H : Maximum value of counter.

24.3.2.8 ETNCnIPGR — Interpacket Gap Register

The ETNCnIPGR register sets the interpacket gap (IPG) value.

Do not rewrite this register while the ETNCnECMR.TE bit is 1 (transmit function is enabled) or while the ETNCnECMR.RE bit is 1 (receive function is enabled).

Refer to **Section 24.4.1.6, Adjusting Transmission Efficiency by Changing the IPG** for details on the IPG.

Value after reset: 0000 0014_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	IPG[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 24.21 ETNCnIPGR Register Contents

Bit Position	Bit Name	Function
31 to 5	—	Reserved The read value is 0. The write value should be 0.
4 to 0	IPG[4:0]	Interpacket Gap 00 _H : 16 bit times 01 _H : 20 bit times ⋮ 14 _H : 96 bit times (initial value) ⋮ 1F _H : 140 bit times

24.3.2.9 ETNCnAPR — Automatic PAUSE Frame Register

The ETNCnAPR register sets the PAUSE time of the PAUSE frame that is automatically transmitted. The value set in the ETNCnAPR register is used for the `pause_time` parameter of the PAUSE frame.

Do not rewrite this register while the ETNCnECMR.TE bit is 1 (transmit function is enabled) or while the ETNCnECMR.RE bit is 1 (receive function is enabled).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.22 ETNCnAPR Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved The read value is 0. The write value should be 0.
15 to 0	AP[15:0]	Automatic PAUSE Time Setting These bits set the value of the <code>pause_time</code> parameter for a PAUSE frame that is automatically transmitted. Transmission is not performed until the set value multiplied by 512 bit times has elapsed.

24.3.2.10 ETNCnMPR — Manual PAUSE Frame Register

The ETNCnMPR register sets the PAUSE time of the PAUSE frame that is manually transmitted. The value set in the ETNCnMPR register is used for the `pause_time` parameter of the PAUSE frame.

When a value is set to this register, a PAUSE frame is transmitted. Rewrite this register while the ETNCnECMR.TE bit is 1 (transmit function is enabled).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.23 ETNCnMPR Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved The read value is 0. The write value should be 0.
15 to 0	MP[15:0]	Manual PAUSE Time Setting These bits set the value of the <code>pause_time</code> parameter for a PAUSE frame that is manually transmitted. Transmission is not performed until the set value multiplied by 512 bit times has elapsed. The read value is undefined.

24.3.2.11 ETNCnRFCF — Received PAUSE Frame Counter

The ETNCnRFCF register is a counter indicating the number of received PAUSE frames. The counter is reset after this register is read.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RPAUSE[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.24 ETNCnRFCF Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved The read value is 0.
7 to 0	RPAUSE[7:0]	Received PAUSE Frame Count Number of received PAUSE frames.

24.3.2.12 ETNCnTPAUSER — PAUSE Frame Retransmit Count Setting Register

The ETNCnTPAUSER register selects the maximum number of times a PAUSE frame is automatically transmitted.

Do not rewrite this register while the ETNCnECMR.TE bit is 1 (transmit function is enabled).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TPAUSE[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.25 ETNCnTPAUSER Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved The read value is 0. The write value should be 0.
15 to 0	TPAUSE[15:0]	Automatic PAUSE Frame Retransmit Setting 0000 _H : Number of retransmissions is unlimited. 0001 _H : Maximum number of retransmissions is 1. ⋮ FFFF _H : Maximum number of retransmissions is 65,535.

24.3.2.13 ETNCnTPAUSECR — PAUSE Frame Retransmit Counter Register

The ETNCnTPAUSECR register is a counter indicating the number of times a PAUSE frame was automatically retransmitted. The counter is reset after this register is read.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TXP[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.26 ETNCnTPAUSECR Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved The read value is 0.
7 to 0	TXP[7:0]	PAUSE Frame Retransmit Count Number of times a PAUSE frame was retransmitted.

24.3.2.14 ETNCnBCFRR — Broadcast Frame Receive Count Setting Register

The ETNCnBCFRR register sets the number of times broadcast frames can be received continuously. When the number of received frames exceeds the ETNCnBCF[15:0] bit value, the excess broadcast frames are discarded.

Do not rewrite this register while the ETNCnECMR.RE bit is 1 (receive function is enabled).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BCF[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.27 ETNCnBCFRR Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved The read value is 0. The write value should be 0.
15 to 0	BCF[15:0]	Broadcast Frame Continuous Receive Count Setting 0000 _H : Number of receptions is unlimited. 0001 _H : Receive 1 frame. ⋮ FFFF _H : Receive 65,535 frames.

24.3.2.15 ETNCnMAHR — MAC Address Upper Bit Register

The ETNCnMAHR register sets the upper 32 bits (b47 to b16) of the 48-bit MAC address. For example, if the MAC address is 01-23-45-67-89-AB, set the register to 0123 4567_H.

Set the ETNCnMAHR register during initialization after a reset. Do not rewrite this register while the ETNCnECMR.TE bit is 1 (transmit function is enabled) or while the ETNCnECMR.RE bit is 1 (receive function is enabled). When rewriting this register, set the ETNCnEDMR.SWR bit to 1 to reset the EDMAC and ETHERC and then set this register again.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MA[47:32]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.28 ETNCnMAHR Register Contents

Bit Position	Bit Name	Function
31 to 0	MA[47:16]	MAC address [47-16] These bits set the upper 32 bits of the MAC address.

24.3.2.16 ETNCnMALR — MAC Address Lower Bit Register

The ETNCnMALR register sets the lower 16 bits of the 48-bit MAC address. For example, if the MAC address is 01-23-45-67-89-AB, set the register to 0000 89AB_H.

Set the ETNCnMALR register during initialization after a reset. Do not rewrite this register while the ETNCnECMR.TE bit is 1 (transmit function is enabled) or while the ETNCnECMR.RE bit is 1 (receive function is enabled). When rewriting this register, set the ETNCnEDMR.SWR bit to 1 to reset the EDMAC and ETHERC and then set this register again.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.29 ETNCnMALR Register Contents

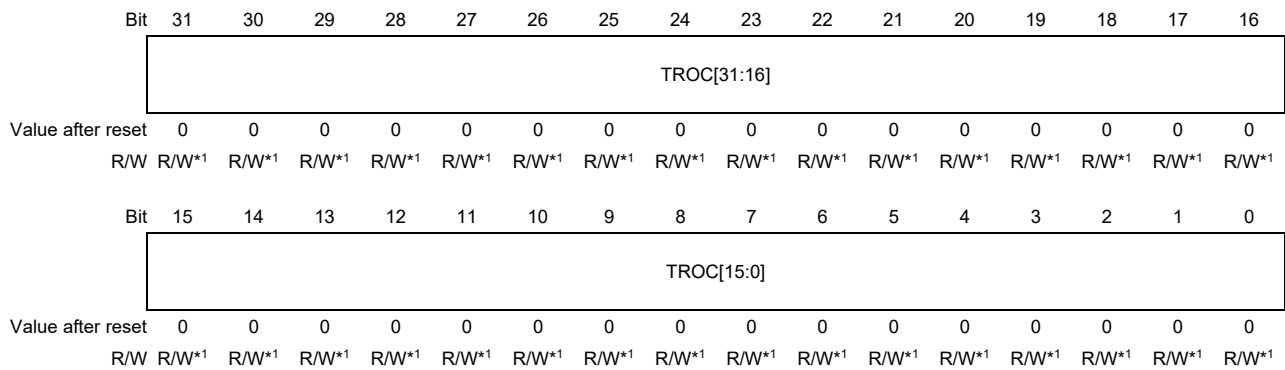
Bit Position	Bit Name	Function
31 to 16	—	Reserved The read value is 0. The write value should be 0.
15 to 0	MA[15:0]	MAC address [15-0] These bits set the lower 16 bits of the MAC address.

24.3.2.17 ETNCnTROCR — Transmit Retry Over Counter Register

The ETNCnTROCR register is a counter indicating the number of frames that fail to be retransmitted.

The ETNCnTROCR register is incremented by 1 when a frame fails to be retransmitted 15 times. The counter stops when the ETNCnTROCR register value becomes FFFF FFFF_H. The counter value becomes 0 by writing any value to the ETNCnTROCR register.

Value after reset: 0000 0000_H



Note 1. This register is cleared to 0 by writing any value to TROC.

Table 24.30 ETNCnTROCR Register Contents

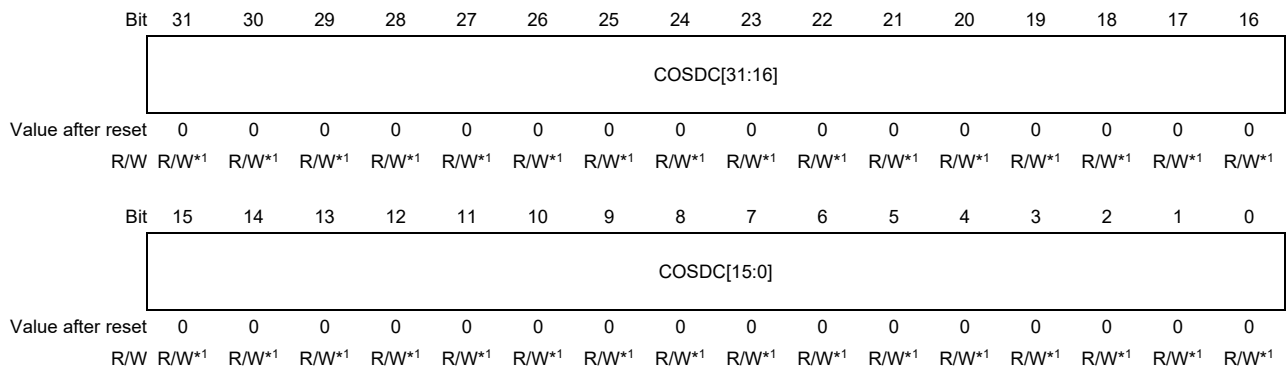
Bit Position	Bit Name	Function
31 to 0	TROC[31:0]	Transmit Retry Over Count These bits indicate the number of frames that were unable to be transmitted in 16 transmission attempts including the retransfer.

24.3.2.18 ETNCnCDCR — Late Collision Detect Counter Register

The ETNCnCDCR register is a counter indicating the number of late collisions that have been detected since transmission started.

When the ETNCnCDCR register value becomes FFFF FFFF_H, the counter stops. The counter value becomes 0 by writing any value to the ETNCnCDCR register.

Value after reset: 0000 0000_H



Note 1. This register is cleared to 0 by writing any value to CDCR.

Table 24.31 ETNCnCDCR Register Contents

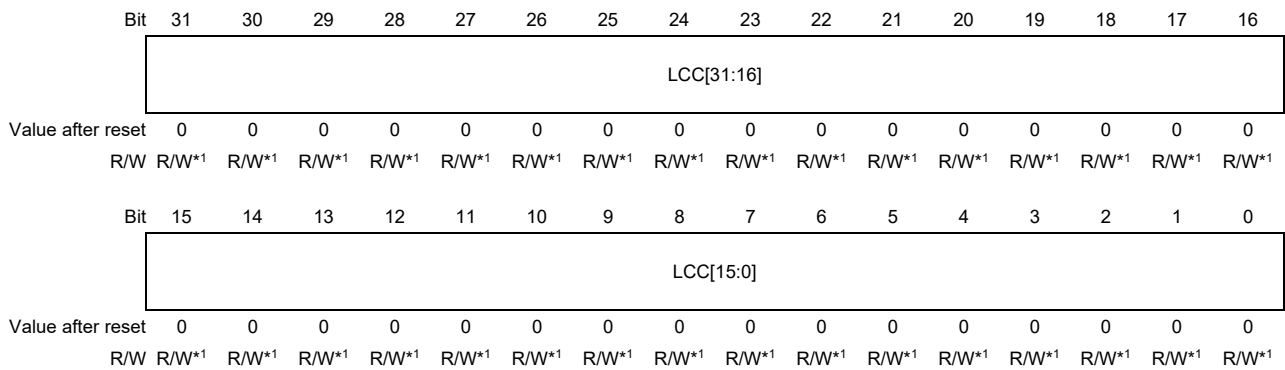
Bit Position	Bit Name	Function
31 to 0	COSDC[31:0]	Delayed Collision Detect Count These bits indicate the number of late collisions since the start of data transmission.

24.3.2.19 ETNCnLCCR — Lost Carrier Counter Register

The ETNCnLCCR register is a counter indicating the number of times a loss of carrier is detected during frame transmission.

When the ETNCnLCCR register value becomes FFFF FFFF_H, the counter stops. The counter value becomes 0 by writing any value to the ETNCnLCCR register.

Value after reset: 0000 0000_H



Note 1. This register is cleared to 0 by writing any value to LCCR.

Table 24.32 ETNCnLCCR Register Contents

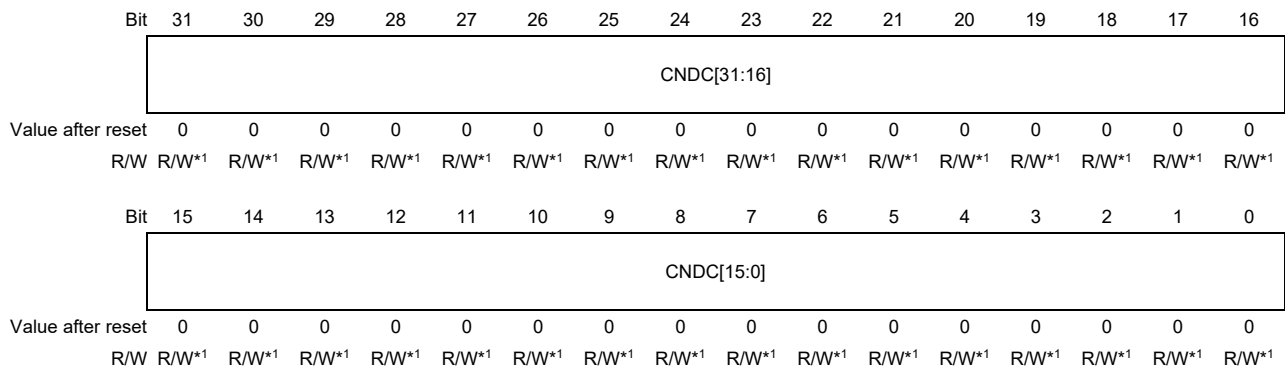
Bit Position	Bit Name	Function
31 to 0	LCC[31:0]	Lost Carrier Count These bits indicate the number of times the carrier was lost during data transmission.

24.3.2.20 ETNCnCNDCR — Carrier Not Detected Counter Register

The ETNCnCNDCR register is a counter indicating the number of times a carrier is not detected during preamble transmission.

When the ETNCnCNDCR register value becomes FFFF FFFF_H, the counter stops. The counter value becomes 0 by writing any value to the ETNCnCNDCR register.

Value after reset: 0000 0000_H



Note 1. This register is cleared to 0 by writing any value to ETNCnCNDCR.

Table 24.33 ETNCnCNDCR Register Contents

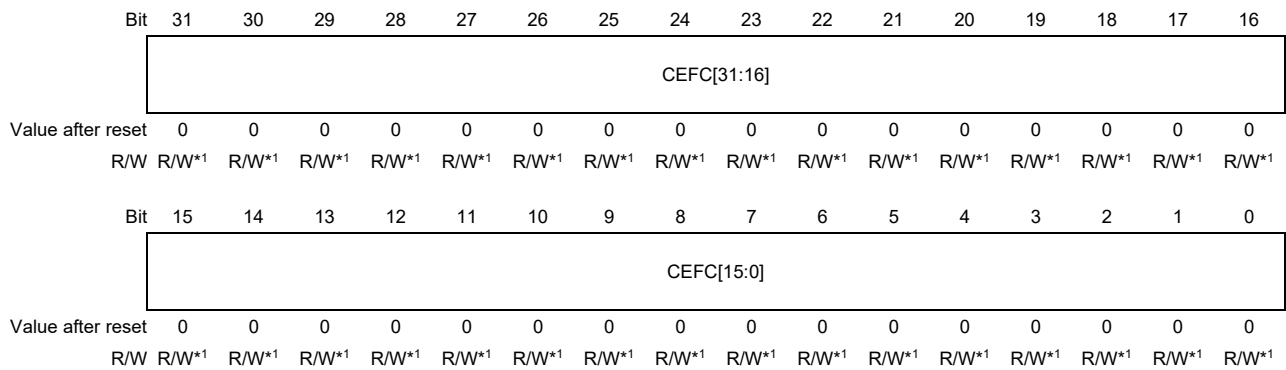
Bit Position	Bit Name	Function
31 to 0	CNDC[31:0]	Carrier Non Detected Count These bits indicate the number of times the carrier was not detected during preamble transmission.

24.3.2.21 ETNCnCEFRCR — CRC Error Frame Receive Counter Register

The ETNCnCEFRCR register is a counter indicating the number of received frames where a CRC error has been detected.

When the ETNCnCEFRCR register value becomes FFFF FFFF_H, the counter stops. The counter value becomes 0 by writing any value to the ETNCnCEFRCR register.

Value after reset: 0000 0000_H



Note 1. This register is cleared to 0 by writing any value to ETNCnCEFRCR.

Table 24.34 ETNCnCEFRCR Register Contents

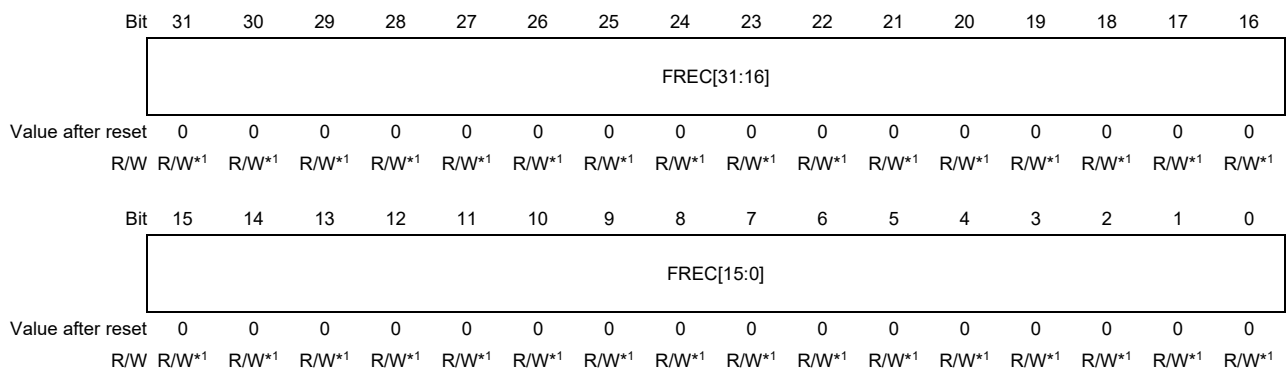
Bit Position	Bit Name	Function
31 to 0	CEFC[31:0]	CRC Error Frame Count These bits indicate the number of CRC error frames received.

24.3.2.22 ETNCnFRECR — Frame Receive Error Counter Register

The ETNCnFRECR register is a counter indicating the number of times a frame receive error has occurred. The PHY-LSI notifies the ETHERC of the frame receive error using the MII_RX_ER pin.

The ETNCnFRECR register is incremented each time the MII_RX_ER pin becomes high. When the ETNCnFRECR register value becomes FFFF FFFF_H, the counter stops. The counter value becomes 0 by writing any value to the ETNCnFRECR register.

Value after reset: 0000 0000_H



Note 1. This register is cleared to 0 by writing any value to ETNCnFRECR.

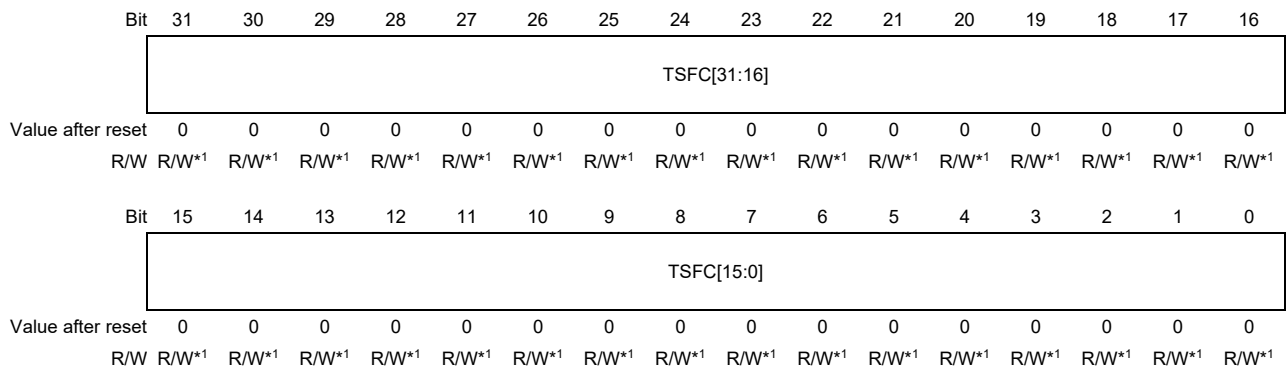
Table 24.35 ETNCnFRECR Register Contents

Bit Position	Bit Name	Function
31 to 0	FRECC[31:0]	Frame Receive Error Count These bits indicate the number of errors during frame reception.

24.3.2.23 ETNCnTSFRCR — Too-Short Frame Receive Counter Register

The ETNCnTSFRCR register is a counter indicating the number of times a short frame that is shorter than 64 bytes has been received. When the ETNCnTSFRCR register value becomes FFFF FFFF_H, the counter stops. The counter value becomes 0 by writing any value to the ETNCnTSFRCR register.

Value after reset: 0000 0000_H



Note 1. This register is cleared to 0 by writing any value to ETNCnTSFRCR.

Table 24.36 ETNCnTSFRCR Register Contents

Bit Position	Bit Name	Function
31 to 0	TSFC[31:0]	Too-Short Frame Receive Count These bits indicate the number of frames received with a length of less than 64 bytes.

24.3.2.24 ETNCnTLFRCR — Too-Long Frame Receive Counter Register

The ETNCnTLFRCR register is a counter indicating the number of times a long frame that is longer than the ETNCnRFLR register value has been received.

When the ETNCnTLFRCR register value becomes FFFF FFFF_H, the counter stops. The counter value becomes 0 by writing any value to the ETNCnTLFRCR register.

Note that the ETNCnTLFRCR register is not incremented when a frame is received with the alignment error. In this case, the ETNCnRFCR register is incremented.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TLFC[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TLFC[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Note 1. This register is cleared to 0 by writing any value to ETNCnTLFRCR.

Table 24.37 ETNCnTLFRCR Register Contents

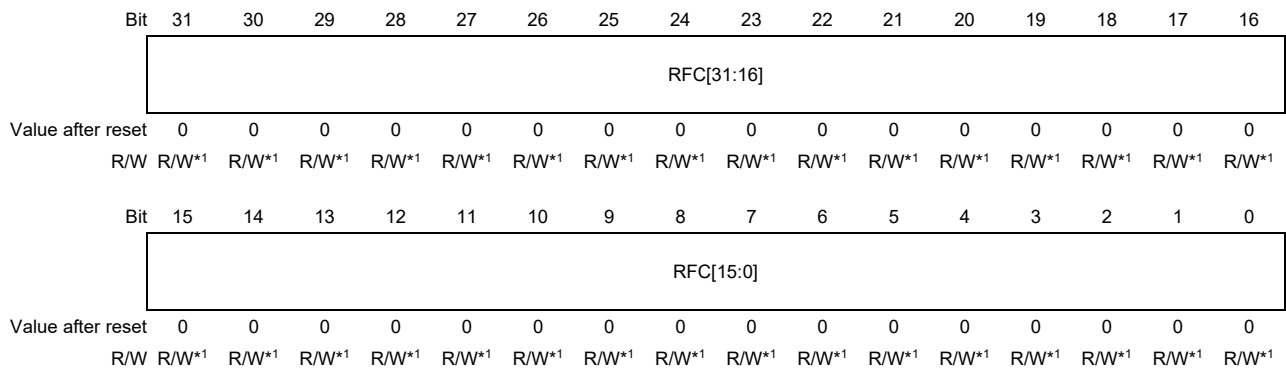
Bit Position	Bit Name	Function
31 to 0	TLFC[31:0]	Too-Long Frame Receive Count These bits indicate the number of frames received with a length exceeding the value of the ETNCnRFLR register.

24.3.2.25 ETNCnRFCR — Received Alignment Error Frame Counter Register

The ETNCnRFCR register is a counter indicating the number of times a frame has been received with the alignment error (frame is not an integral number of octets).

When the ETNCnRFCR register value becomes FFFF FFFF_H, the counter stops. The counter value becomes 0 by writing any value to the ETNCnRFCR register.

Value after reset: 0000 0000_H



Note 1. This register is cleared to 0 by writing any value to ETNCnRFCR.

Table 24.38 ETNCnRFCR Register Contents

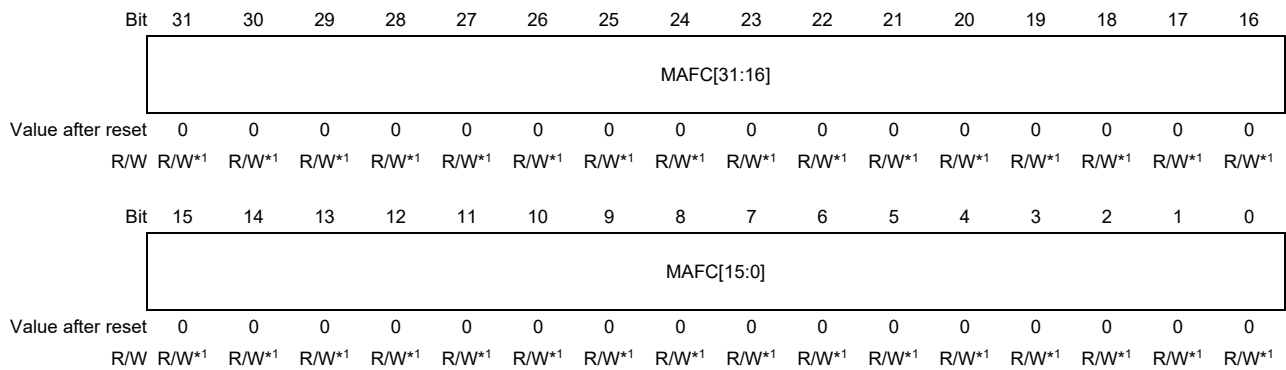
Bit Position	Bit Name	Function
31 to 0	RFC[31:0]	Residual-Bit Frame Receive Count These bits indicate the number of frames received containing residual bits.

24.3.2.26 ETNCnMAFCR — Multicast Address Frame Receive Counter Register

The ETNCnMAFCR register is a counter indicating the number of times a frame with a multicast address set has been received.

When the ETNCnMAFCR register value becomes FFFF FFFF_H, the counter stops. The counter value becomes 0 by writing any value to the ETNCnMAFCR register.

Value after reset: 0000 0000_H



Note 1. This register is cleared to 0 by writing any value to ETNCnMAFCR.

Table 24.39 ETNCnMAFCR Register Contents

Bit Position	Bit Name	Function
31 to 0	MAFC[31:0]	Multicast Address Frame Count These bits indicate the number of multicast frames received.

24.3.3 EDMAC Registers

24.3.3.1 ETNCnEDMR — EDMAC Mode Register

The ETNCnEDMR register controls EDMAC operations.

Set the ETNCnEDMR register during the initialization process after a reset. When rewriting this register other than during the initialization process, set the SWR bit to 1 to reset the EDMAC and ETHERC, and then set this register again. If the ETHERC and EDMAC are reset during data transmission or reception, abnormal data may be sent on the line. Do not rewrite this register while the ETHERC transmit or receive function is enabled. It takes 64 cycles of the peripheral module clock (PCLKA) to initialize the ETHERC and EDMAC. Complete the initialization process before accessing registers in the ETHERC and EDMAC.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	DE	DL[1:0]	—	—	—	—	SWR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R/W

Table 24.40 ETNCnEDMR Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved The read value is 0. The write value should be 0.
6	DE	Big Endian Mode/Little Endian Mode*1 0: Big endian mode 1: Little endian mode
5 and 4	DL[1:0]	Transmit/Receive Descriptor Length b5 b4 0 0: 16 bytes 0 1: 32 bytes 1 0: 64 bytes 1 1: 16 bytes
3 to 1	—	Reserved The read value is 0. The write value should be 0.
0	SWR	Software Reset When 1 is written, the corresponding channels of the EDMAC and ETHERC are reset. The ETNCnTDLAR, ETNCnRDLAR, ETNCnRMFCR, ETNCnTFUCR, and ETNCnRFOCR registers are not reset by this bit. The read value is 0.

Note 1. This setting applies to data for the transmit / receive buffer. It does not apply to transmit / receive descriptors and registers.

24.3.3.2 ETNCnEDTRR — EDMAC Transmit Request Register

The ETNCnEDTRR register controls EDMAC transmission.

After the EDMAC has transmitted one frame, it reads the next descriptor. When the TD0.TACT bit in the descriptor is 1, the EDMAC continues transmission. When the TD0.TACT bit is 0, the EDMAC sets the TR bit to 0 and stops transmission.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 24.41 ETNCnEDTRR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved The read value is 0. The write value should be 0.
0	TR	Transmit Request When 1 is written, the EDMAC reads the corresponding descriptor and transmits frames whose TD0.TACT bit is 1. The TR bit becomes 0 after all the valid frames are transmitted. Writing 0 to this bit has no effect.

24.3.3.3 ETNCnEDRRR — EDMAC Receive Request Register

The ETNCnEDRRR register controls EDMAC reception.

When the RR bit becomes 1, the EDMAC reads the receive descriptor. When the RD0.RACT bit is 1, the EDMAC waits for a receive request from the ETHERC.

When the EDMAC has received data equivalent to the receive buffer size, it reads the next descriptor and waits to receive a frame. If the RD0.RACT bit is 0, the EDMAC sets the RR bit to 0 and stops reception.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 24.42 ETNCnEDRRR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved The read value is 0. The write value should be 0.
0	RR	Receive Request 0: Receive function is disabled.*1 1: Receive descriptor is read, and the receive function is enabled.

Note 1. If the receive function is disabled during frame reception, write-back to the receive descriptor is not performed successfully. Subsequent pointers for reading a receive descriptor become abnormal and the EDMAC cannot operate normally. In this case, to enable the EDMAC receive function again, execute a software reset by setting the ETNCnEDMR.SWR bit to 1. To disable the EDMAC receive function without resetting the EDMAC, set the ETNCnECMR.RE bit to 0. Next, after the EDMAC has completed reception and write-back to the receive descriptor has been confirmed, set the RR bit to 0.

24.3.3.4 ETNCnTDLAR — Transmit Descriptor List Start Address Register

The ETNCnTDLAR register sets the start address of the transmit descriptor list.

Allocate each descriptor on the corresponding boundary according to the descriptor length selected by the ETNCnEDMR.DL[1:0] bits.

Do not rewrite the TDLAR register during transmission. Rewrite the ETNCnTDLAR register while the ETNCnEDTRR.TR bit is 0.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TDLA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDLA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.43 ETNCnTDLAR Register Contents

Bit Position	Bit Name	Function
31 to 0	TDLA [31:0]	Transmit Descriptor Start Address These bits set start address of the transmit descriptor list. Set the start address according to the descriptor length selected by the ETNCnEDMR.DL[1:0] bits. 16-byte boundary: Lower 4 bits = 0000 _B 32-byte boundary: Lower 5 bits = 00000 _B 64-byte boundary: Lower 6 bits = 000000 _B

24.3.3.5 ETNCnRDLAR — Receive Descriptor List Start Address Register

The ETNCnRDLAR register sets the start address of the receive descriptor list.

Allocate each descriptor on the corresponding boundary according to the descriptor length selected by the ETNCnEDMR.DL[1:0] bits.

Do not rewrite the ETNCnRDLAR register during reception. Rewrite the ETNCnRDLAR register while the ETNCnEDRRR.RR bit is 0.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDLA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDLA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.44 ETNCnRDLAR Register Contents

Bit Position	Bit Name	Function
31 to 0	RDLA[31:0]	Receive Descriptor Start Address These bits set the start address of the receive descriptor list. Set the start address according to the descriptor length selected by the ETNCnEDMR.DL[1:0] bits. 16-byte boundary: Lower 4 bits = 0000 _B 32-byte boundary: Lower 5 bits = 00000 _B 64-byte boundary: Lower 6 bits = 000000 _B

24.3.3.6 ETNCnEESR — ETHERC/EDMAC Status Register

The ETNCnEESR register indicates the ETHERC and EDMAC communication status.

Each flag in the ETNCnEESR register can be output as an interrupt request signal (INTETNCn) from the EDMAC. Each flag, excluding the ECI flag, becomes 0 by writing 1. The value of each flag is not changed by writing 0. Each interrupt source can be enabled by the corresponding bit in the ETNCnEESIPR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	TWB	—	—	—	TABT	RABT	RFCOF	ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CND	DLC	CD	TRO	RMAF	—	—	RRF	RTLFL	RTSF	PRE	CERF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Table 24.45 ETNCnEESR Register Contents (1/2)

Bit Position	Bit Name	Function
31	—	Reserved The read value is 0. The write value should be 0.
30	TWB	Write-Back Complete Flag 0: Write-back has not been completed, or no transmission has been requested. 1: Write-back to the transmit descriptor has been completed.
29 to 27	—	Reserved The read value is 0. The write value should be 0.
26	TABT	Transmit Abort Detect Flag 0: Frame transmission has not been aborted or no transmission has been requested. 1: Frame transmission has been aborted.
25	RABT	Receive Abort Detect Flag 0: Frame reception has not been aborted or no reception has been requested. 1: Frame reception has been aborted.
24	RFCOF	Receive Frame Counter Overflow Flag 0: Receive frame counter has not overflowed. 1: Receive frame counter has overflowed.
23	ADE	Address Error Flag 0: Invalid memory address has not been detected (normal operation). 1: Invalid memory address has been detected.*2
22	ECI	ETHERC Status Register Source Flag 0: ETHERC status interrupt source has not been detected. 1: ETHERC status interrupt source has been detected.
21	TC	Frame Transfer Complete Flag 0: Transfer has not been completed, or no transfer has been requested. 1: All frames indicated by the transmit descriptor have been completely transferred to the transmit FIFO.

Table 24.45 ETNCnEESR Register Contents (2/2)

Bit Position	Bit Name	Function
20	TDE	Transmit Descriptor Empty Flag 0: The EDMAC detects that the transmit descriptor valid bit (TD0.TACT) is 1. 1: The EDMAC detects that the transmit descriptor valid bit (TD0.TACT) is 0.
19	TFUF	Transmit FIFO Underflow Flag 0: Underflow has not occurred. 1: Underflow has occurred.
18	FR	Frame Receive Flag 0: Frame has not been received. 1: Frame has been received. Update of the receive descriptor is complete.
17	RDE	Receive Descriptor Empty Flag 0: The EDMAC detects that the receive descriptor valid bit (RD0.RACT) is 1. 1: The EDMAC detects that the receive descriptor valid bit (RD0.RACT) is 0.
16	RFOF	Receive FIFO Overflow Flag 0: Overflow has not occurred. 1: Overflow has occurred.
15 to 12	—	Reserved The read value is 0. The write value should be 0.
11	CND	Carrier Not Detected Flag 0: A carrier was detected before the start of transmission. 1: A carrier was not detected during preamble transmission.
10	DLC	Loss of Carrier Detected Flag 0: Loss of carrier has not been detected. 1: Loss of carrier was detected during frame transmission.
9	CD	Late Collision Detected Flag 0: Late collision has not been detected. 1: Late collision was detected during frame transmission.
8	TRO	Transmit Retry Over Flag 0: Transmit retry-over condition has not been detected. 1: Transmit retry-over condition has been detected.
7	RMAF	Multicast Address Frame Receive Flag 0: Multicast address frame has not been received. 1: Multicast address frame has been received.
6 and 5	—	Reserved The read value is 0. The write value should be 0.
4	RRF	Alignment Error Flag 0: Alignment error has not been detected. 1: Alignment error has been detected.
3	RTLF	Frame-Too-Long Error Flag 0: Frame-too-long error has not been detected. 1: Frame-too-long error has been detected.
2	RTSF	Frame-Too-Short Error Flag 0: Frame-too-short error has not been detected. 1: Frame-too-short error has been detected.
1	PRE	PHY-LSI Receive Error Flag 0: PHY-LSI receive error has not been detected. 1: PHY-LSI receive error has been detected.
0	CERF	CRC Error Flag 0: CRC error has not been detected. 1: CRC error has been detected.

Note 1. The ECI flag is read only. When the source in the ETNCnECSR register is cleared, the ECI flag is also cleared.

Note 2. When an address error is detected, the EDMAC halts the current process. To resume the operation, set the EDMR.SWR bit to 1 (EDMAC and ETHERC are reset) and then configure the EDMAC and ETHERC again.

Write-Back Complete Flag (TWB Flag)

The TWB flag indicates that EDMAC has completed writing back to the descriptor after frame transmission. Note that this flag becomes 1 after each frame transmission when the ETNCnTRIMD.TIM bit is 0. This flag becomes 1 only when the ETNCnTRIMD.TIS bit is 1.

Transmit Abort Detect Flag (TABT Flag)

The TABT flag indicates that the ETHERC has aborted frame transmission due to transmit retry over, loss of carrier, no carrier detection, or another error.

Receive Abort Detect Flag (RABT Flag)

The RABT flag indicates that the ETHERC has aborted frame reception due to a CRC error, PHY-LSI receive error, frame-too-short error, frame-too-long error, or another error.

Receive Frame Counter Overflow Flag (RFCOF Flag)

The RFCOF flag indicates that the next frame reception has started even though the number of frames stored in the receive FIFO has reached the maximum number of frames (16 frames). Note that the received frame is discarded while the RFCOF flag is 1.

Address Error Flag (ADE Flag)

The ADE flag indicates that the memory address that the EDMAC tried to transfer is invalid.

ETHERC Status Register Source Flag (ECI Flag)

The ECI flag becomes 1 when an interrupt request is generated by the ETNCnECSR register.

Frame Transfer Complete Flag (TC Flag)

The TC flag indicates that all the data specified by the transmit descriptor has been transmitted from the ETHERC. This flag becomes 1 when one frame has been transmitted in single-buffer frame transmission or when the last data of a frame is transmitted in multi-buffer frame transmission and the TD0.TACT bit in the next transmit descriptor is 0. After frame transmission is completed, the EDMAC writes the transfer status back to the descriptor.

Transmit Descriptor Empty Flag (TDE Flag)

The TDE flag indicates that the TD0.TACT bit of the transmit descriptor is 0 while the previous transmit descriptor is indicating that the frame is not complete (TD0.TFP[1:0] bits are 10_B or 00_B) in multi-buffer frame transmission. As a result, an incomplete frame may be sent.

When this flag becomes 1, perform a software reset and then set the EDTRR.TR bit to 1 to resume transmission.

Transmission starts from the address stored in the TDLAR register.

Transmit FIFO Underflow Flag (TFUF Flag)

The TFUF flag indicates that no data remains in the transmit FIFO during frame transmission. Incomplete data has been sent on the line.

Frame Receive Flag (FR Flag)

The FR flag indicates that a frame has been received and the receive descriptor has been updated. The FR flag becomes 1 every time a frame is received.

Receive Descriptor Empty Flag (RDE Flag)

The RDE flag indicates that the read receive descriptor is invalid.

When this flag becomes 1, set the RD0.RACT bit in the receive descriptor to 1 and set the ETNCnEDRRR.RR bit to 1 to resume reception.

Receive FIFO Overflow Flag (RFOF Flag)

The RFOF flag indicates that the receive FIFO has overflowed during frame reception.

Carrier Not Detected Flag (CND Flag)

The CND flag becomes 1 when a carrier was not detected during preamble transmission.

Loss of Carrier Detected Flag (DLC Flag)

The DLC flag indicates that a loss of carrier was detected during frame transmission.

Late Collision Detected Flag (CD Flag)

The CD flag indicates that a late collision was detected during frame transmission.

Transmit Retry Over Flag (TRO Flag)

The TRO flag indicates that a collision occurred again during the 15th retry of frame transmission.

Multicast Address Frame Receive Flag (RMAF Flag)

The RMAF flag indicates that a multicast frame has been received.

Alignment Error Flag (RRF Flag)

The RRF flag indicates that a frame is not an integral number of octets. The last word that is not an integral number of octets is not transferred.

Frame-Too-Long Error Flag (RTLF Flag)

The RTLF flag indicates that a received frame is greater than the upper limit of the receive frame length set in the ETNCnRFLR register. The excess data is discarded.

Frame-Too-Short Error Flag (RTSF Flag)

The RTSF flag indicates that a received frame was less than 64 bytes.

PHY-LSY Receive Error Flag (PRE Flag)

The PRE flag indicates the RX_ER signal output from the PHY-LSI is high.

CRC Error Flag (CERF Flag)

The CERF flag becomes 1 when an error is detected in checking the frame check sequence (FCS) field of the receive frame.

24.3.3.7 ETNCnEESIPR — ETHERC/EDMAC Status Interrupt Enable Register

The ETNCnEESIPR register enables interrupt requests corresponding to bits in the ETNCnEESR register. When a bit in this register is 1, the corresponding interrupt request is enabled.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	TWBIP	—	—	—	TABTIP	RABTIP	RFCOFIP	ADEIP	ECIIP	TCIP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CNDIP	DLCIP	CDIP	TROIP	RMAFIP	—	—	RRFIP	RTLFIIP	RTSFIP	PREIP	CERFIIP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Table 24.46 ETNCnEESIPR Register Contents (1/2)

Bit Position	Bit Name	Function
31	—	Reserved The read value is 0. The write value should be 0.
30	TWBIP	Write-Back Complete Interrupt Request Enable 0: Write-back complete interrupt request is disabled. 1: Write-back complete interrupt request is enabled.
29 to 27	—	Reserved The read value is 0. The write value should be 0.
26	TABTIP	Transmit Abort Detected Interrupt Request Enable 0: Transmit abort detected interrupt request is disabled. 1: Transmit abort detected interrupt request is enabled.
25	RABTIP	Receive Abort Detected Interrupt Request Enable 0: Receive abort detected interrupt request is disabled. 1: Receive abort detected interrupt request is enabled.
24	RFCOFIP	Receive Frame Counter Overflow Interrupt Request Enable 0: Receive frame counter overflow interrupt request is disabled. 1: Receive frame counter overflow interrupt request is enabled.
23	ADEIP	Address Error Interrupt Request Enable 0: Address error interrupt request is disabled. 1: Address error interrupt request is enabled.
22	ECIIP	ETHERC Status Register Source Interrupt Request Enable 0: ETHERC status interrupt request is disabled. 1: ETHERC status interrupt request is enabled.
21	TCIP	Frame Transfer Complete Interrupt Request Enable 0: Frame transmission complete interrupt request is disabled. 1: Frame transmission complete interrupt request is enabled.
20	TDEIP	Transmit Descriptor Empty Interrupt Request Enable 0: Transmit descriptor empty interrupt request is disabled. 1: Transmit descriptor empty interrupt request is enabled.
19	TFUFIP	Transmit FIFO Underflow Interrupt Request Enable 0: Underflow interrupt request is disabled. 1: Underflow interrupt request is enabled.

Table 24.46 ETNCnEESIPR Register Contents (2/2)

Bit Position	Bit Name	Function
18	FRIP	Frame Receive Interrupt Request Enable 0: Frame reception interrupt request is disabled. 1: Frame reception interrupt request is enabled.
17	RDEIP	Receive Descriptor Empty Interrupt Request Enable 0: Receive descriptor empty interrupt request is disabled. 1: Receive descriptor empty interrupt request is enabled.
16	RFOFIP	Receive FIFO Overflow Interrupt Request Enable 0: Overflow interrupt request is disabled. 1: Overflow interrupt request is enabled.
15 to 12	—	Reserved The read value is 0. The write value should be 0.
11	CNDIP	Carrier Not Detected Interrupt Request Enable 0: Carrier not detected interrupt request is disabled. 1: Carrier not detected interrupt request is enabled.
10	DLCIP	Loss of Carrier Detected Interrupt Request Enable 0: Loss of carrier detected interrupt request is disabled. 1: Loss of carrier detected interrupt request is enabled.
9	CDIP	Late Collision Detected Interrupt Request Enable 0: Late collision detected interrupt request is disabled. 1: Late collision detected interrupt request is enabled.
8	TROIP	Transmit Retry Over Interrupt Request Enable 0: Transmit retry over interrupt request is disabled. 1: Transmit retry over interrupt request is enabled.
7	RMAFIP	Multicast Address Frame Receive Interrupt Request Enable 0: Multicast address frame receive interrupt request is disabled. 1: Multicast address frame receive interrupt request is enabled.
6 and 5	—	Reserved The read value is 0. The write value should be 0.
4	RRFIP	Alignment Error Interrupt Request Enable 0: Alignment error interrupt request is disabled. 1: Alignment error interrupt request is enabled.
3	RTLFIIP	Frame-Too-Long Error Interrupt Request Enable 0: Frame-too-long error interrupt request is disabled. 1: Frame-too-long error interrupt request is enabled.
2	RTSFIIP	Frame-Too-Short Error Interrupt Request Enable 0: Frame-too-short error interrupt request is disabled. 1: Frame-too-short error interrupt request is enabled.
1	PREIP	PHY-LSI Receive Error Interrupt Request Enable 0: PHY-LSI receive error interrupt request is disabled. 1: PHY-LSI receive error interrupt request is enabled.
0	CERFIIP	CRC Error Interrupt Request Enable 0: CRC error interrupt request is disabled. 1: CRC error interrupt request is enabled.

24.3.3.8 ETNCnTRSCER — ETHERC/EDMAC Transmit/Receive Status Copy Enable Register

The ETNCnTRSCER register selects whether the receive status indicated by the ETNCnEESR.RMAF and RRF flags is reflected in the RFE bit of the receive descriptor as a summary. The bits in the ETNCnTRSCER register correspond to bits in the ETNCnEESR register that have the same number. When setting the RMAFCE or RRFCE bit to 0, the corresponding receive status (ETNCnEESR.EESR.RMAF or RRF flag) is reflected in the RFE bit of the receive descriptor. When setting the RMAFCE or RRFCE bit to 1, the corresponding receive status is not reflected in the RFE bit.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RMAFCE	—	—	RRFCE	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R	R	R	R

Table 24.47 ETNCnTRSCER Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved The read value is 0. The write value should be 0.
7	RMAFCE	RMAF Flag Copy Enable 0: The ETNCnEESR.EESR.RMAF flag status is reflected in the RD0.RFE bit of the receive descriptor. 1: The ETNCnEESR.EESR.RMAF flag status is not reflected in the RD0.RFE bit of the receive descriptor.
6 and 5	—	Reserved The read value is 0. The write value should be 0.
4	RRFCE	RRF Flag Copy Enable 0: The ETNCnEESR.EESR.RRF flag status is reflected in the RD0.RFE bit of the receive descriptor. 1: The ETNCnEESR.EESR.RRF flag status is not reflected in the RD0.RFE bit of the receive descriptor.
3 to 0	—	Reserved The read value is 0. The write value should be 0.

24.3.3.9 ETNCnRMFCR — Missed-Frame Counter Register

The ETNCnRMFCR register indicates that the number of frames that could not be stored in the receive FIFO and were therefore discarded during reception. When the receive FIFO overflows, it stops receiving data, and the remaining frames are discarded. At the same time, the ETNCnRMFCR register value is incremented. When the ETNCnRMFCR register value reaches FFFF_H, incrementing is halted. When any value is written to the ETNCnRMFCR register, the counter value becomes 0.

For the frame that has not been completely received, after data in the receive FIFO is transferred to the receive buffer, the RACT bit in the receive descriptor 0 (RD0) becomes 0 (descriptor disabled), the RFS9 bit becomes 1 (receive FIFO overflow), and the ETNCnEESR.RFOF flag becomes 1 (overflow has occurred).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MFC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.48 ETNCnRMFCR Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved The read value is 0. The write value should be 0.
15 to 0	MFC[15:0]	Missed-Frame Counter These bits indicate the number of frames that were discarded and not transferred to the receive buffer during reception.

24.3.3.10 ETNCnTFTR — Transmit FIFO Threshold Register

The ETNCnTFTR register sets the transmit FIFO threshold at which the first transmission starts. The actual threshold is the set value multiplied by 4.

The ETHERC starts transmission when the amount of data in the transmit FIFO exceeds the number of bytes set in the ETNCnTFTR register, when the transmit FIFO is full, or when one frame of data has been completely written. Set the ETNCnTFTR register while the ETNCnEDTRR.TR bit is 0.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TFT[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.49 ETNCnTFTR Register Contents

Bit Position	Bit Name	Function
31 to 11	—	Reserved The read value is 0. The write value should be 0.
10 to 0	TFT[10:0]	Transmit FIFO Threshold 000 _H : Store and forward mode 001 _H to 00C _H : Setting prohibited 00D _H to 200 _H : The threshold is the set value multiplied by 4. Example: 00D _H : 52 bytes 040 _H : 256 bytes 100 _H : 1024 bytes 200 _H : 2048 bytes 201 _H to 7FF _H : Setting prohibited

NOTES

- When starting transmission before one frame of data has been completely written, take care to prevent an underflow.
- To prevent a transmit underflow, using the initial value (store and forward mode) is recommended.

24.3.3.11 ETNCnFDR — FIFO Depth Register

The ETNCnFDR register sets the transmit and receive FIFO depths.

Set this register to 0000 0707_H before starting transmission and reception.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	TFD[4:0]				—	—	—	RFD[4:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 24.50 ETNCnFDR Register Contents

Bit Position	Bit Name	Function
31 to 13	—	Reserved The read value is 0. The write value should be 0.
12 to 8	TFD[4:0]	Transmit FIFO Depth b12 b8 00111: 2048 bytes Settings other than above are prohibited.
7 to 5	—	Reserved The read value is 0. The write value should be 0.
4 to 0	RFD[4:0]	Receive FIFO Depth b4 b0 00111: 1968 bytes Settings other than above are prohibited.

24.3.3.12 ETNCnRMCR — Receive Method Control Register

The ETNCnRMCR register sets how to control the ETNCnEDRRR.RR bit when receiving a frame.

Since the ETNCnEDRRR.RR bit becomes 0 when one frame has been received while the RNR bit is 0, the RR bit needs to be set to 1 by software to receive the subsequent frame.

Since the ETNCnEDRRR.RR bit does not become 0 when one frame has been received while the RNR bit is 1, the EDMAC reads the next receive descriptor and continues receiving frames. It is recommended to set the RNR bit to 1 when receiving data continuously.

Set the ETNCnRMCR register while the ETNCnEDRRR.RR bit is 0.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RNR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 24.51 ETNCnRMCR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved The read value is 0. The write value should be 0.
0	RNR	Receive Request Reset 0: ETNCnEDRRR.RR bit (receive request bit) is set to 0 when one frame has been received. 1: ETNCnEDRRR.RR bit (receive request bit) is not set to 0 when one frame has been received.

24.3.3.13 ETNCnTFUCR — Transmit FIFO Underflow Counter

The ETNCnTFUCR register indicates how many times the transmit FIFO has underflowed.

The counter value becomes 0 by writing any value to the ETNCnTFUCR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UNDER[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.52 ETNCnTFUCR Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved The read value is 0. The write value should be 0.
15 to 0	UNDER[15:0]	Transmit FIFO Underflow Count These bits indicate how many times the transmit FIFO has underflowed. The counter stops when the counter value reaches FFFF _H .

24.3.3.14 ETNCnRFOCR — Receive FIFO Overflow Counter

The ETNCnRFOCR register indicates how many times the receive FIFO has overflowed.

The counter value becomes 0 by writing any value to the ETNCnRFOCR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OVER[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.53 ETNCnRFOCR Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved The read value is 0. The write value should be 0.
15 to 0	OVER[15:0]	Receive FIFO Overflow Count These bits indicate how many times the receive FIFO has overflowed. The counter stops when the counter value reaches FFFF _H .

24.3.3.15 ETNCnIOSR — Independent Output Signal Setting Register

The ETNCnIOSR register selects the output level of the ETHERC external output (EXOUT) pin in external loopback mode.

The ELB bit value is output to the EXOUT pin of the MCU as is, and can be used to set loopback mode for the PHY-LSI. To use the loopback function of the PHY-LSI through this register, the PHY-LSI needs to be provided with the pin to be connected to the EXOUT pin.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ELB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 24.54 ETNCnIOSR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved The read value is 0. The write value should be 0.
0	ELB	External Loopback Mode 0: The EXOUT pin outputs low. 1: The EXOUT pin outputs high.

24.3.3.16 ETNCnFCFTR — Flow Control Start FIFO Threshold Setting Register

The ETNCnFCFTR register sets the ETHERC flow control (sets the threshold for automatically transmitting a PAUSE frame).

The threshold can be set using the data size (RFDO[2:0] bits) and the number of frames (RFFO[2:0] bits) stored in the receive FIFO. The flow control starts when the stored data size or number of stored frames reaches the threshold.

Value after reset: 0007 0007_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFO[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RFDO[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 24.55 ETNCnFCFTR Register Contents

Bit Position	Bit Name	Function
31 to 19	—	Reserved The read value is 0. The write value should be 0.
18 to 16	RFFO[2:0]	Receive FIFO Frame PAUSE Output Threshold b18 b17 b16 0 0 0: When 2 receive frames have been stored in the receive FIFO. 0 0 1: When 4 receive frames have been stored in the receive FIFO. 0 1 0: When 6 receive frames have been stored in the receive FIFO. ⋮ 1 1 0: When 14 receive frames have been stored in the receive FIFO. 1 1 1: When 16 receive frames have been stored in the receive FIFO.
15 to 3	—	Reserved The read value is 0. The write value should be 0.
2 to 0	RFDO[2:0]	Receive FIFO Data PAUSE Output Threshold b2 b1 b0 0 0 0: When 224 (256 - 32) bytes of data are stored in the receive FIFO. 0 0 1: When 480 (512 - 32) bytes of data are stored in the receive FIFO. ⋮ 1 1 0: When 1760 (1792 - 32) bytes of data are stored in the receive FIFO. 1 1 1: When 2016 (2048 - 32) bytes of data are stored in the receive FIFO.

24.3.3.17 ETNCnRPADIR — Receive Data Padding Insert Register

The ETNCnRPADIR register sets insertion of padding for received data. The padding value is 00_H.

Set the ETNCnEDMR.SWR bit to 1 to reset before rewriting the ETNCnPRADIR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PADS[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	PADR[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.56 ETNCnRPADIR Register Contents

Bit Position	Bit Name	Function															
31 to 18	—	Reserved The read value is 0. The write value should be 0.															
17 and 16	PADS[1:0]	Padding Size <table border="0"> <tr> <td>b17</td> <td>b16</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>No padding is inserted.</td> </tr> <tr> <td>0</td> <td>1</td> <td>1 byte is inserted.</td> </tr> <tr> <td>1</td> <td>0</td> <td>2 bytes are inserted.</td> </tr> <tr> <td>1</td> <td>1</td> <td>3 bytes are inserted.</td> </tr> </table>	b17	b16		0	0	No padding is inserted.	0	1	1 byte is inserted.	1	0	2 bytes are inserted.	1	1	3 bytes are inserted.
b17	b16																
0	0	No padding is inserted.															
0	1	1 byte is inserted.															
1	0	2 bytes are inserted.															
1	1	3 bytes are inserted.															
15 to 6	—	Reserved The read value is 0. The write value should be 0.															
5 to 0	PADR[5:0]	Padding Slot 00 _H : Padding is inserted at the head of received data. 01 _H : Padding is inserted between the 1st byte and 2nd byte of received data. ⋮ 3E _H : Padding is inserted between the 62nd byte and 63rd byte of received data. 3F _H : Padding is inserted between the 63rd byte and 64th byte of received data															

24.3.3.18 ETNCnTRIMD — Transmit Interrupt Setting Register

The ETNCnTRIMD register sets transmit interrupt mode and enables/disables the transmit interrupt.

When the condition set in this register is satisfied, the ETNCnEESR.TWB flag becomes 1, and an interrupt request is output when the ETNCnEESIPR.TWBIP bit is 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	TIM	—	—	—	TIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Table 24.57 ETNCnTRIMD Register Contents

Bit Position	Bit Name	Function
31 to 5	—	Reserved The read value is 0. The write value should be 0.
4	TIM	Transmit Interrupt Mode 0: Transmission complete interrupt mode An interrupt occurs when a frame has been transmitted. 1: Write-back complete interrupt mode An interrupt occurs when write-back to the transmit descriptor has been completed while the TWBI bit is 1.
3 to 1	—	Reserved The read value is 0. The write value should be 0.
0	TIS	Transmit Interrupt Enable 0: Transmit Interrupt is disabled. 1: Transmit Interrupt is enabled. Set the ETNCnEESR.TWB flag to 1 in the mode selected by the TIM bit to report an interrupt.

24.3.3.19 ETNCnRBWAR — Receive Buffer Write Address Register

The ETNCnRBWAR register indicates the last address that the EDMAC wrote data to when writing to the receive buffer.

Refer to the address indicated by the ETNCnRBWAR register to find out which address in the receive buffer the EDMAC is writing data to. Note that the address that the EDMAC is outputting to the receive buffer may not match the read value of the ETNCnRBWAR register during data reception.

The ETNCnRBWAR register is read only. Do not write to this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RBWA[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBWA[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.58 ETNCnRBWAR Register Contents

Bit Position	Bit Name	Function
31 to 0	RBWA[31:0]	The ETNCnRBWAR register is read only. Do not write to this register.

24.3.3.20 ETNCnRDFAR — Receive Descriptor Fetch Address Register

The ETNCnRDFAR register indicates the start address of the last fetched receive descriptor when the EDMAC fetches descriptor information from the receive descriptor.

Refer to the address indicated by the ETNCnRDFAR register to find out which receive descriptor information the EDMAC is using for the current processing. Note that the address of the receive descriptor that the EDMAC fetches may not match the read value of the ETNCnRDFAR register during data reception.

The ETNCnRDFAR register is read only. Do not write to this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDFAR[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDFAR[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.59 ETNCnRDFAR Register Contents

Bit Position	Bit Name	Function
31 to 0	RDFAR[31:0]	Receive Descriptor Fetch Address The ETNCnRDFAR register is read only. Do not write to this register.

24.3.3.21 ETNCnTBRAR — Transmit Buffer Read Address Register

The ETNCnTBRAR register indicates the last address that the EDMAC has read data from when reading data from the transmit buffer.

Refer to the address indicated by the ETNCnTBRAR register to find out which address in the transmit buffer the EDMAC is reading from. Note that the address that the EDMAC is outputting to the transmit buffer may not match the read value of the ETNCnTBRAR register.

The ETNCnTBRAR register is read only. Do not write to this register.

Value after reset:		0000 0000 _H																								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	TBRA[31:16]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TBRA[15:0]									
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	TBRA[15:0]									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	TBRA[15:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TBRA[15:0]									
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	TBRA[15:0]									

Table 24.60 ETNCnTBRAR Register Contents

Bit Position	Bit Name	Function
31 to 0	TBRA[31:0]	Transmit Buffer Read Address The ETNCnTBRAR register is read only. Do not write to this register.

24.3.3.22 ETNCnTDFAR — Transmit Descriptor Fetch Address Register

The ETNCnTDFAR register indicates the start address of the last fetched transmit descriptor when the EDMAC fetches descriptor information from the transmit descriptor.

Refer to the address indicated by the ETNCnTDFAR register to find out which transmit descriptor information the EDMAC is using for the current processing.

Note that the address of the transmit descriptor that the EDMAC fetches may not match the read value of the ETNCnTDFAR register.

The ETNCnTDFAR register is read only. Do not write to this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TDFAR[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDFAR[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.61 ETNCnTDFAR Register Contents

Bit Position	Bit Name	Function
31 to 0	TDFAR[31:0]	Transmit Descriptor Fetch Address The ETNCnTDFAR register is read only. Do not write to this register.

24.4 Operation

24.4.1 ETHERC Operation

This section is an overview of the ETHERC operations. The ETHERC supports flow control compliant with IEEE802.3x, and can transmit and receive PAUSE frames.

24.4.1.1 Transmission

The ETHERC transmitter assembles transmit data into a frame and outputs it to the MII/RMII when a transmit request is received from the EDMAC. The frame transmitted via the MII/RMII is transmitted on the line by the PHY-LSI.

Figure 24.5 shows the state transitions of the ETHERC transmitter.

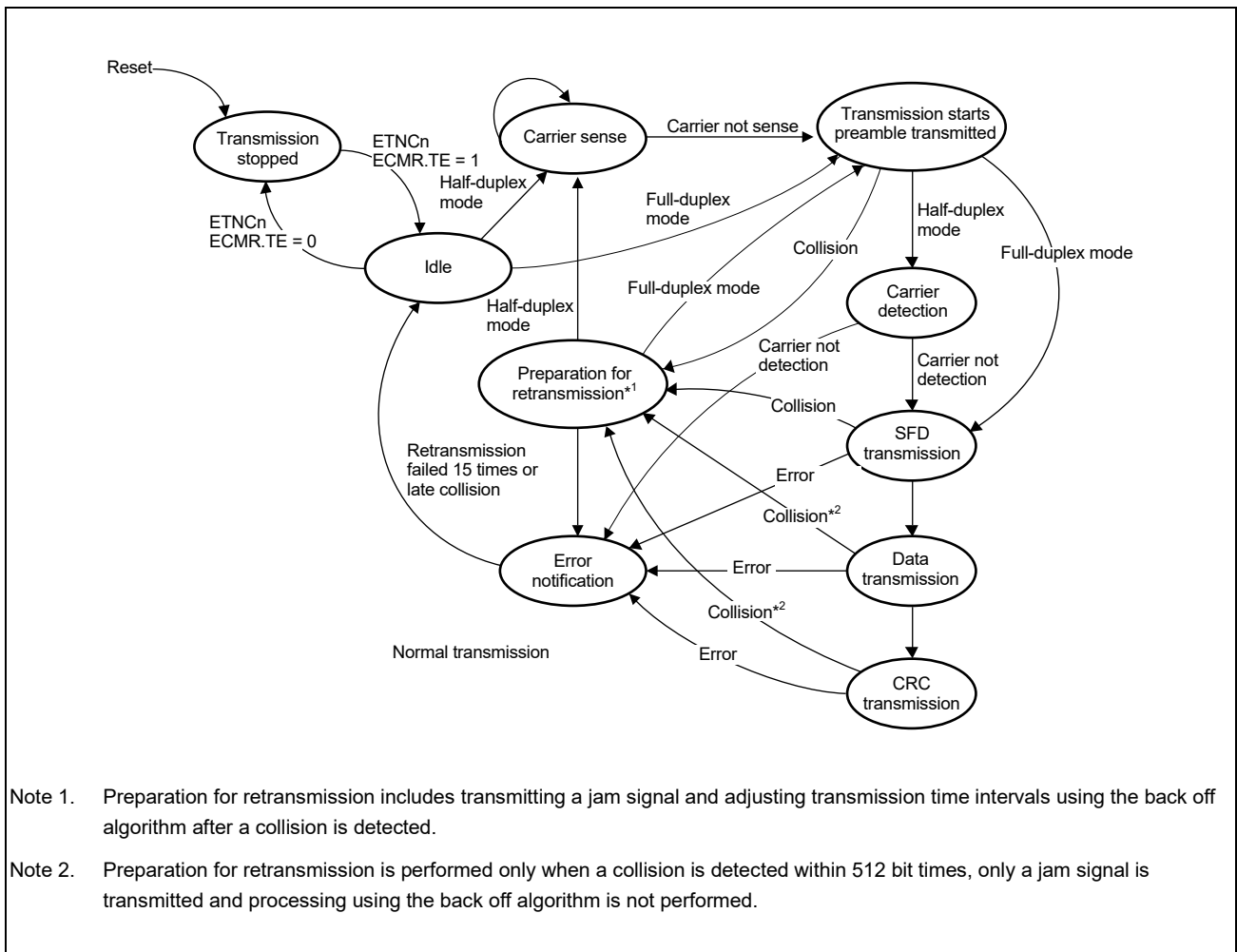


Figure 24.5 ETHERC Transmitter State Transitions

1. When setting the ETNCnECMR.TE bit to 1, the ETHERC enters the transmit idle state.
2. When a transmit request is received from the EDMAC, the ETHERC enters the carrier sense state. The ETHERC waits for the interpacket gap and then transmits a preamble to the MII/RMII. When full-duplex mode is selected, it is not required to sense a carrier, so the ETHERC transmits a preamble immediately after receiving a transmit request from the EDMAC.
3. The ETHERC transmits the Start Frame Delimiter (SFD), transmit data, and CRC sequentially. When the transmission is completed successfully, the ETHERC notifies the EDMAC of successful completion, and the EDMAC sets the ETNCnEESR.TC flag to 1. When a late collision or loss of carrier is detected during data transmission, the ETHERC stops the transmission and notifies the EDMAC of the error.
4. After the time for the interpacket gap has elapsed, the ETHERC enters the idle state and then continues transmission if transmit data remains.

24.4.1.2 Reception

The ETHERC receiver separates the frame input from the MII/RMII into the preamble, SFD, receive data, and CRC, and transmits only receive data (destination address, source address, type/length, data/LLC).

Figure 24.6 shows the state transitions of the ETHERC receiver.

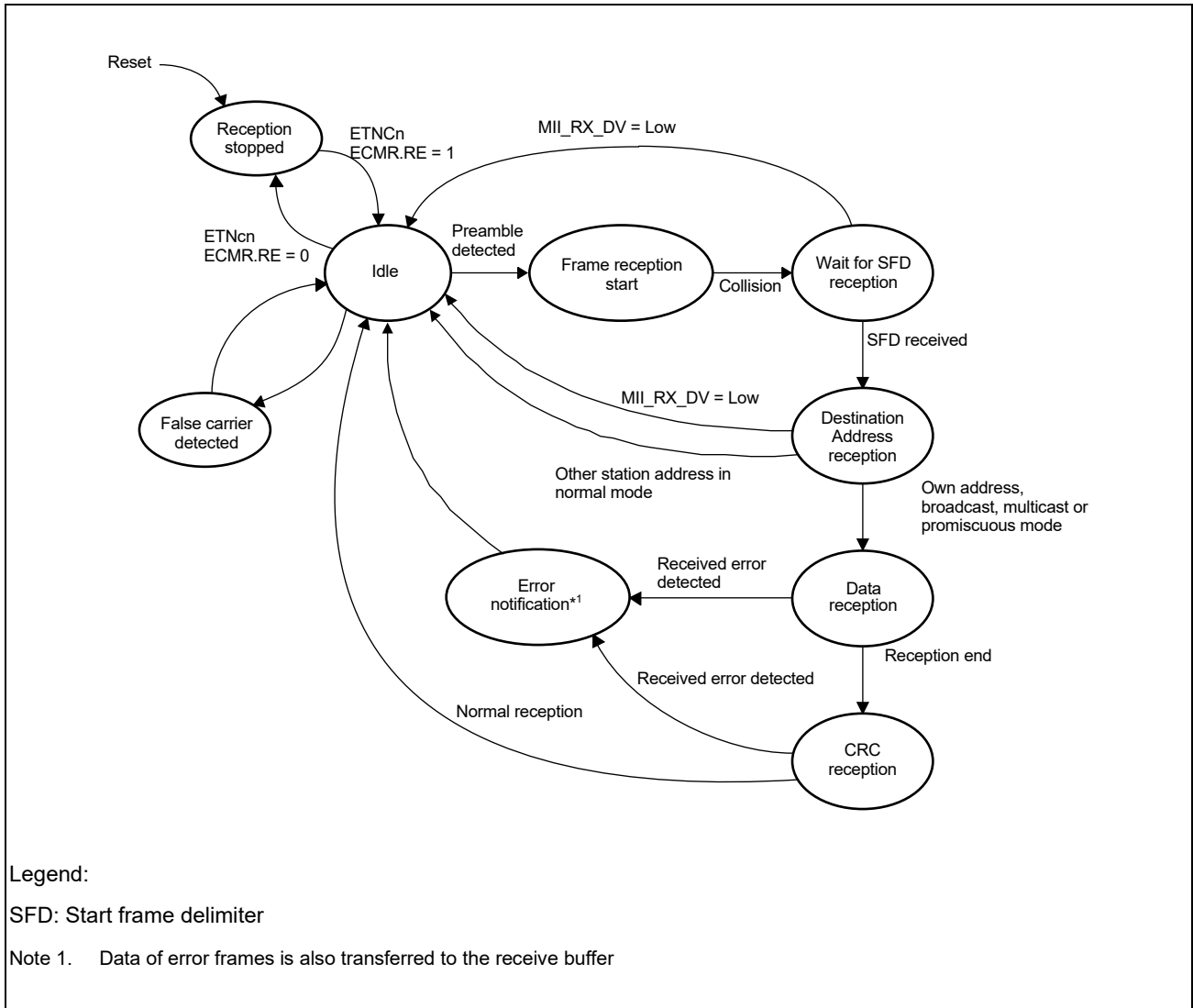


Figure 24.6 ETHERC Receiver State Transitions

1. ETHERC enters the receive idle state by setting the ETNCnECMR.RE bit to 1.
2. When the SFD following the preamble of the receive packet is detected, the ETHERC starts reception. If the received SFD is invalid, the ETHERC discards the frame.
3. In normal mode, the ETHERC starts data reception when the destination address of the receive frame is the address of the MCU or the receive frame is broadcast frame or multicast frame. In promiscuous mode, the ETHERC starts data reception regardless of the receive frame type.
4. After receiving data from the MII/RMII, the ETHERC performs the CRC check. The ETHERC notifies the EDMAC of the CRC check result. After the received data is transferred to the receive buffer, the CRC check result is written back to the receive descriptor as a status. The result is also reflected in the ETNCnEESR.CERF flag.
5. If the ETNCnECMR.RE bit is 1 after one frame has been received, the ETHERC prepares to receive the next frame.

24.4.1.3 Frame Timings

(1) MII Frame Timings

Figure 24.7 to Figure 24.11 show the MII frame timings.

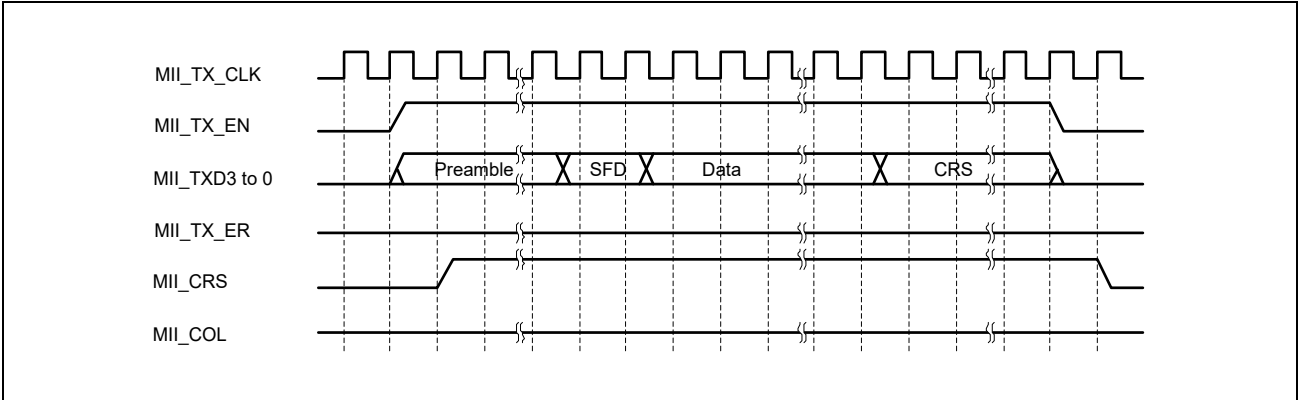


Figure 24.7 MII Frame Transmit Timing during Normal Transmission

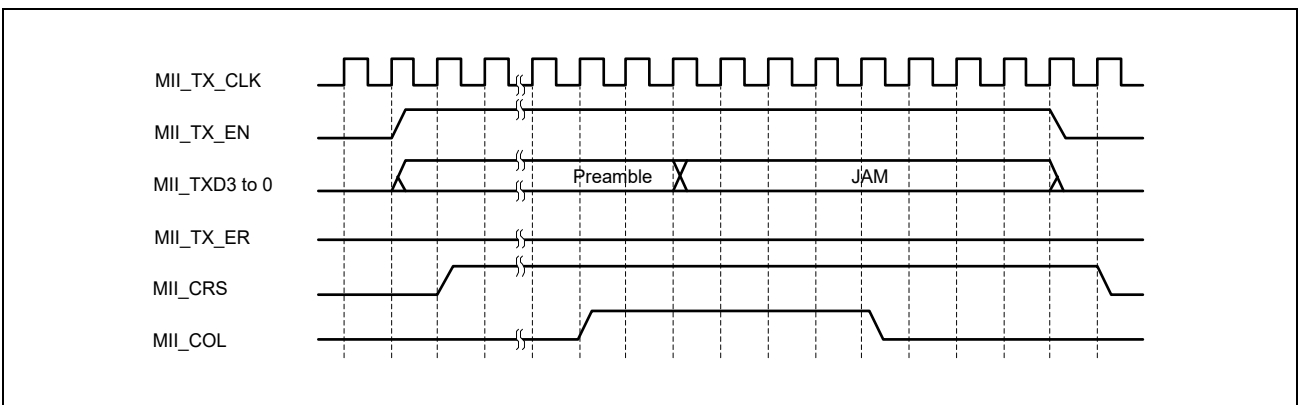


Figure 24.8 MII Frame Transmit Timing When Collision Occurs

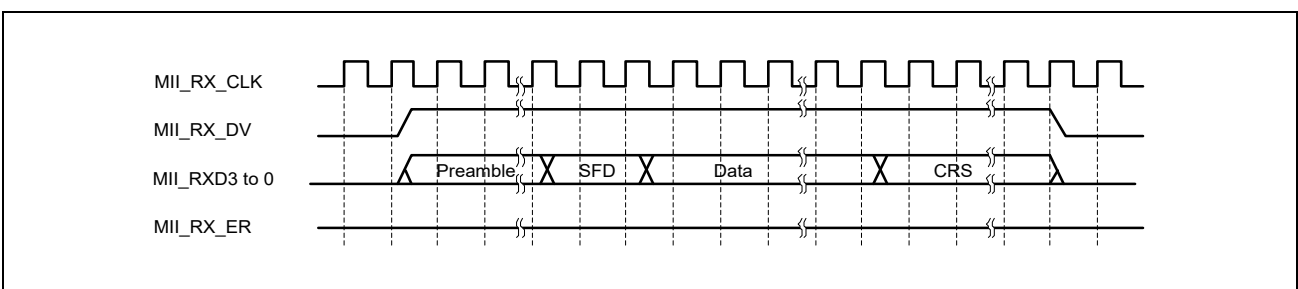


Figure 24.9 MII Frame Receive Timing during Normal Reception

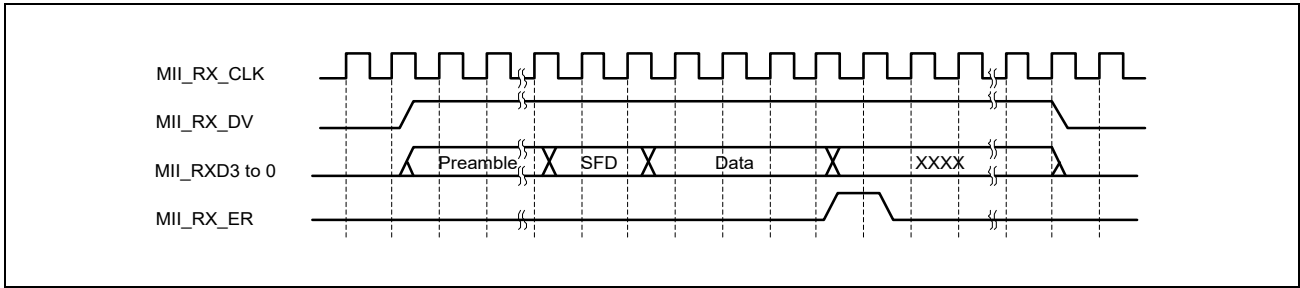


Figure 24.10 MII Frame Receive Timing for Receive Error Notification

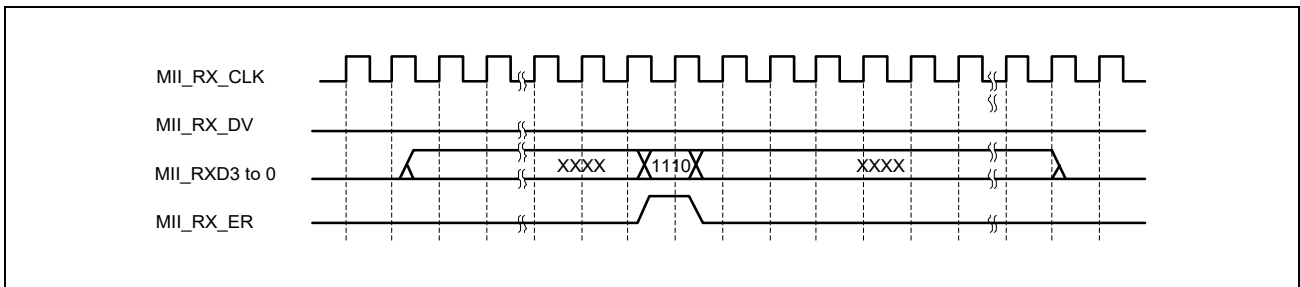


Figure 24.11 MII Frame Receive Timing for False Carrier Notification

(2) RMII Frame Timings

Figure 24.12 to Figure 24.14 show the RMII frame timings.

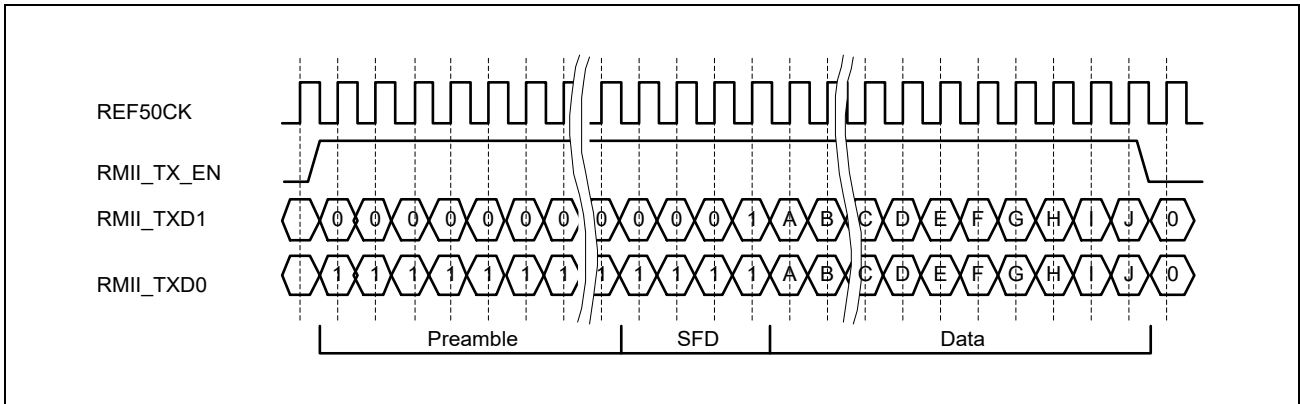


Figure 24.12 RMII Frame Transmit Timing during Normal Transmission

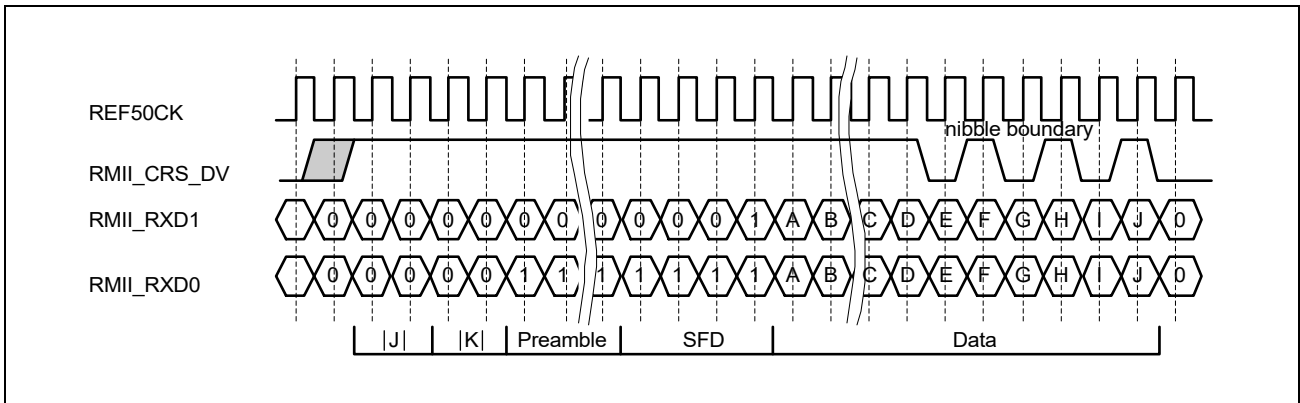


Figure 24.13 RMII Frame Receive Timing during Normal Reception

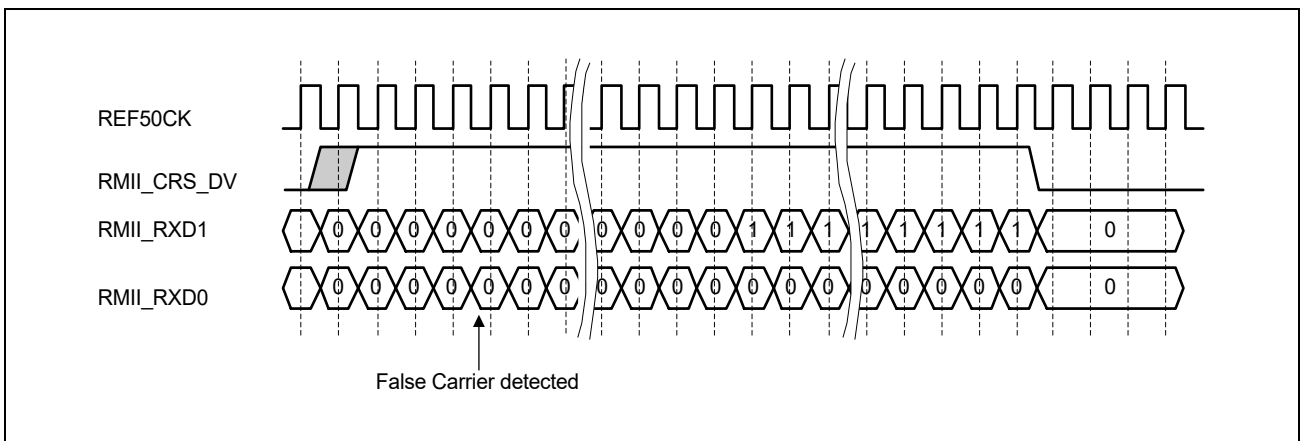


Figure 24.14 RMII Frame Receive Timing When False Carrier Is Detected

24.4.1.4 Accessing MII/RMII Registers

Use the ETNCnPIR register to access the MII/RMII registers in the PHY-LSI. Serial data in the MII/RMII management frame format is transmitted and received via the MDC and MDIO pins controlled by software.

(1) MII/RMII Management Frame Format

Table 24.62 lists the MII/RMII management frame format.

Table 24.62 MII/RMII Management Frame Format

Access Type		MII/RMII Management Frame							
	Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
	Number of bits	32	2	2	5	5	2	16	1
Read		1...1	01	10	00001	RRRRR	Z0	DDDDDDDDDDDDDD DDD	Z
Write		1...1	01	01	00001	RRRRR	10	DDDDDDDDDDDDDD DDD	Z

- PRE (preamble): Send 32 consecutive 1s.
- ST (start of frame): Send 01_B.
- OP (operation code): Send 10_B for read or 01_B for write.
- PHYAD (PHY address): Up to 32 PHY-LSIs can be connected to one MAC. PHY-LSIs are selected with these 5 bits.
 - When the PHY-LSI address is 1, send 00001_B.
- REGAD (register address): One register is selected from up to 32 registers in the PHY-LSI.
 - When the register address is 1, send 00001_B.
- TA (turnaround): 2-bit turnaround time to avoid contention between the register address and data during a read operation
 - Send 10_B during a write operation.
 - Release the bus for 1 bit during a read operation (Z is output) (indicated as Z0 because 0 is output from the PHY-LSI in the next clock cycle).
- DATA (data): 16-bit data. Sequentially send or receive starting from the MSB.
- IDLE (IDLE condition): Wait time before inputting the next MII / RMII management format
 - Release the bus during a write operation (Z is output).
 - During a read operation, no control is required since a bus has already been released.

(2) MII/RMII Register Access Procedure

Access to the MII/RMII registers includes writing data in 1-bit units, reading data in 1-bit units, and releasing the bus.

Figure 24.15 to **Figure 24.18** show examples of the MII/RMII register access timing. The access timing differs depending on the PHY-LSI type.

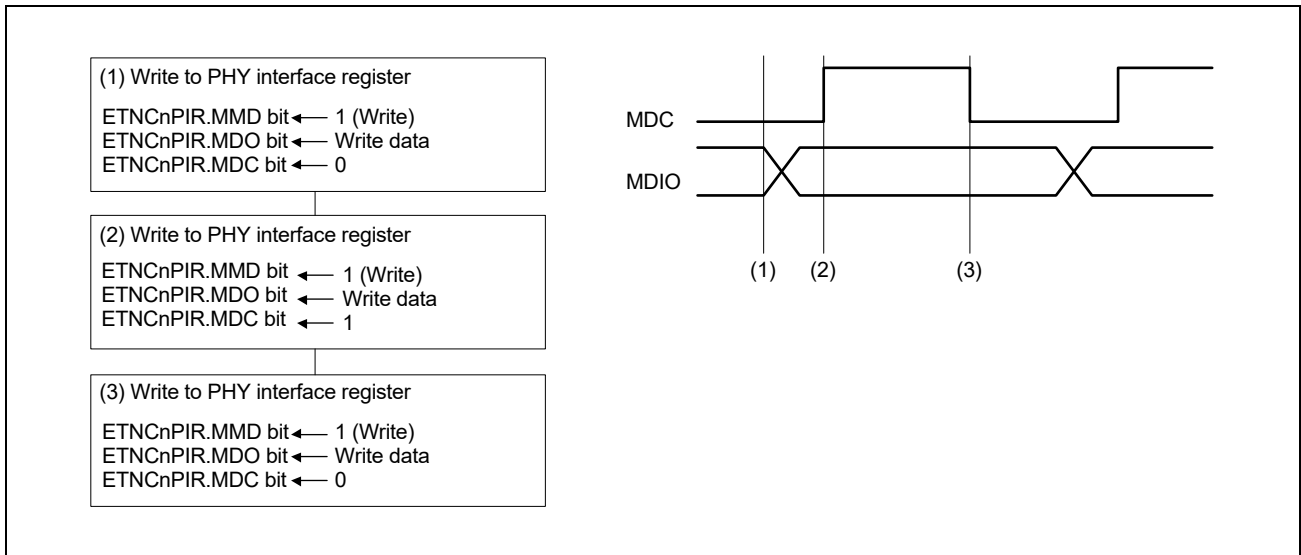


Figure 24.15 1-Bit Data Write Flow

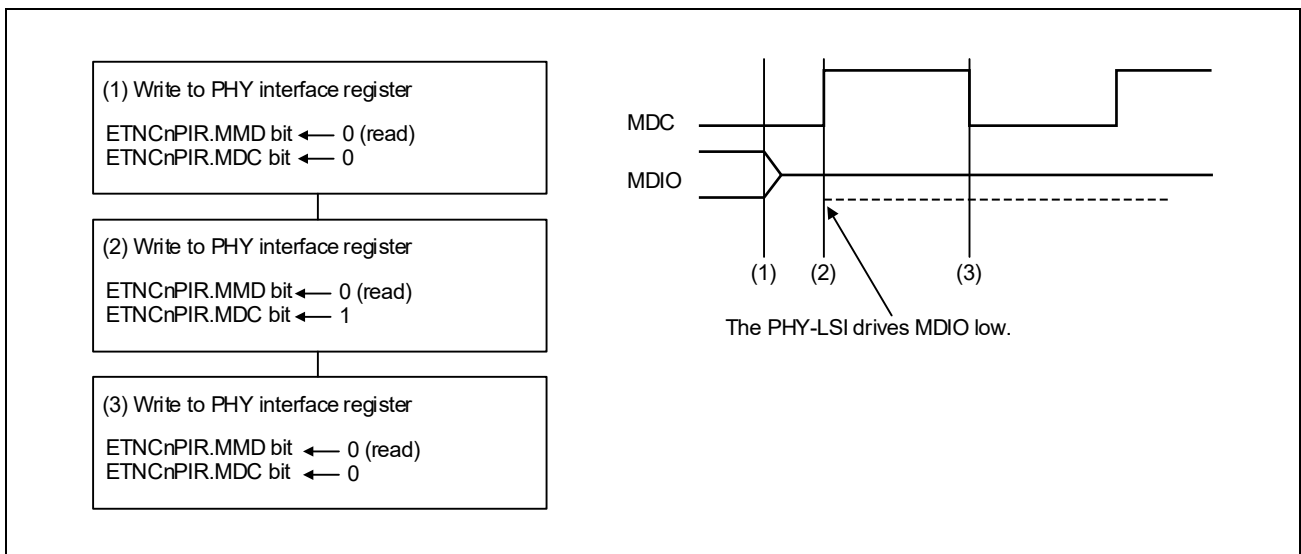


Figure 24.16 Bus Release Flow (TA in Read Operation in **Table 24.62**)

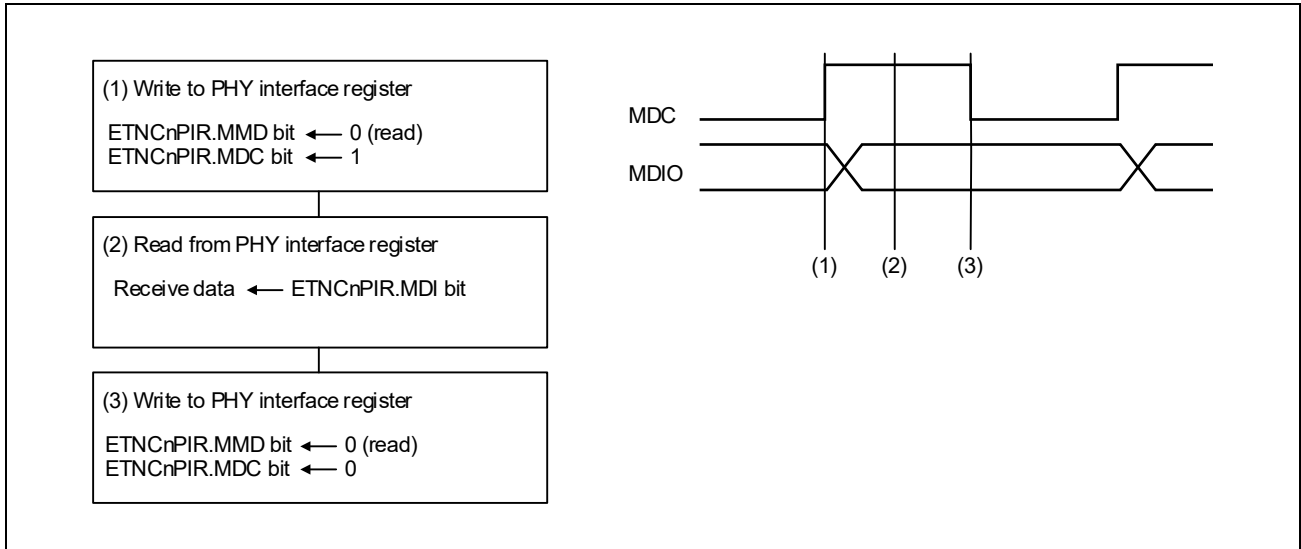


Figure 24.17 1-Bit Data Read Flow

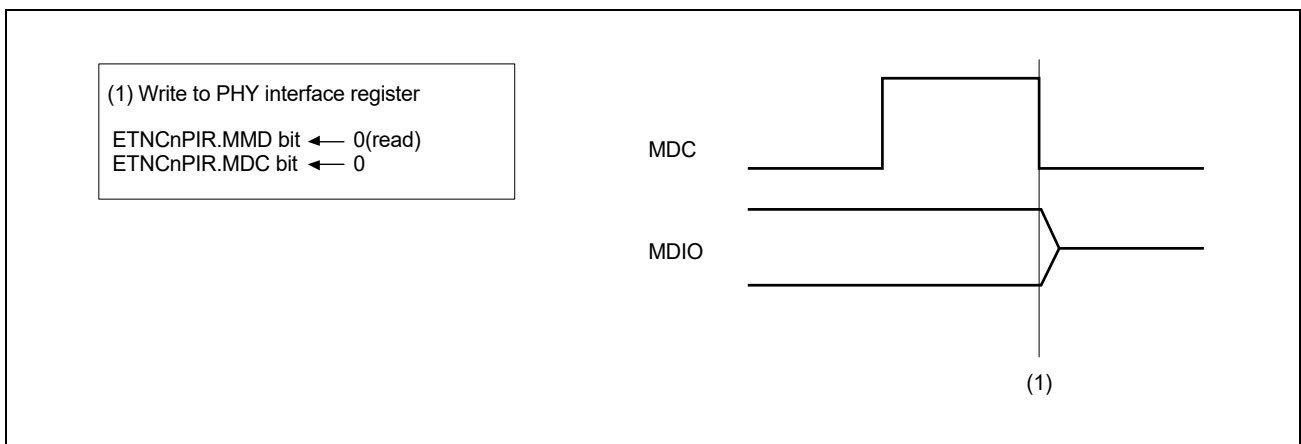


Figure 24.18 Bus Release Flow (IDLE in Write Operation in Table 24.62)

24.4.1.5 Magic Packet Detection

The ETHERC supports Wake-On-LAN (WOL). WOL is a function used to detect a Magic Packet transmitted from a host device or other device and exit a low power consumption state such as HALT mode. When the ETHERC detects a Magic Packet, high level is output from the WOL pin. Write 1 to the ETNCnEDMR.SWR bit to set the WOL pin to low.

Since a Magic Packet is transmitted in broadcast mode, the Magic Packet is received regardless of the destination MAC address selected in the format. The ETHERC outputs a high level from the WOL pin only when the destination MAC address matches its own MAC address. Refer to the technical documentation provided by Advanced Micro Devices, Inc. for details on the Magic Packet. The following describes an example of the procedure to use WOL in the MCU.

1. Set the CPU to disable the INTETNCn interrupt request.
2. Set the ETNCnECMR.MPDE bit to 1 to enable Magic Packet detection. Set the ETNCnECMR.RE bit to 1 to enable reception.
3. Set the ETNCnECSIPR.MPDIP bit to 1 to enable notification of the Magic Packet detect interrupt.
4. Set the ETNCnEESIPR.ECIIP bit to 1 to enable the ETHERC status register source interrupt.
5. Set the CPU to enable the INTETNCn interrupt request.
6. Change the CPU operating mode to low power mode.
7. When a Magic Packet is detected, an interrupt request is sent to the CPU. A high level is output from the WOL pin to notify peripheral devices that the Magic Packet has been detected.

(1) Notes on Magic Packet Detection

The ETHERC receives packets, including broadcast packets, even when waiting to receive a Magic Packet. Therefore, receive data may have been stored in the receive FIFO of the EDMAC when a Magic Packet is detected. Also, flags in ETNCnECSR and ETNCnEESR registers may have been changed. When returning to normal operation by detecting a Magic Packet, set the ETNCnEDMR.SWR bit to 1 to reset the ETHERC and EDMAC.

24.4.1.6 Adjusting Transmission Efficiency by Changing the IPG

The IPG is a non-transmit period between transmit frames. The ETHERC can change the value of the IPG. Transmission efficiency can be increased or decreased by setting the ETNCnIPGR register. The typical value of the IPG is specified by the IEEE802.3 standard. When changing the setting, confirm that all devices operate normally in the same network.

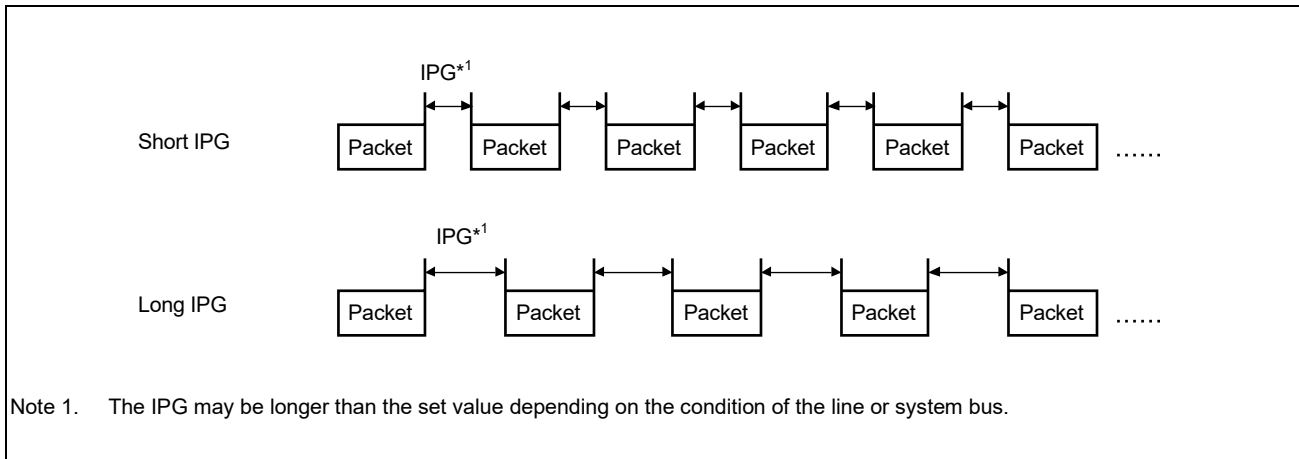


Figure 24.19 Differences in Transmission Efficiency Based on Changes in the IPG

24.4.1.7 Flow Control

The ETHERC can perform flow control compliant with IEEE802.3x in full-duplex mode, and the receiver and transmitter can be set individually. PAUSE frames can be transmitted automatically or manually.

(1) Automatic PAUSE Frame Transmission

When the ETNCnECMR.TXF bit is set to 1, automatic PAUSE frame transmission is enabled. A PAUSE frame is automatically transmitted by a PAUSE frame transmit request from the EDMAC. The ETNCnAPR.AP[15:0] bit value is used for the pause_time parameter of the PAUSE frame.

After a PAUSE frame is transmitted, if the EDMAC is still requesting PAUSE frame transmission when the PAUSE time elapses, a PAUSE frame is transmitted again. The maximum number of PAUSE frame retransmissions can be set in the ETNCnTPAUSER.TPAUSE[15:0] bits. Once the maximum number of retransmissions is reached, subsequent PAUSE frames are not transmitted.

Figure 24.20 shows the procedure to set automatic PAUSE frame transmission.

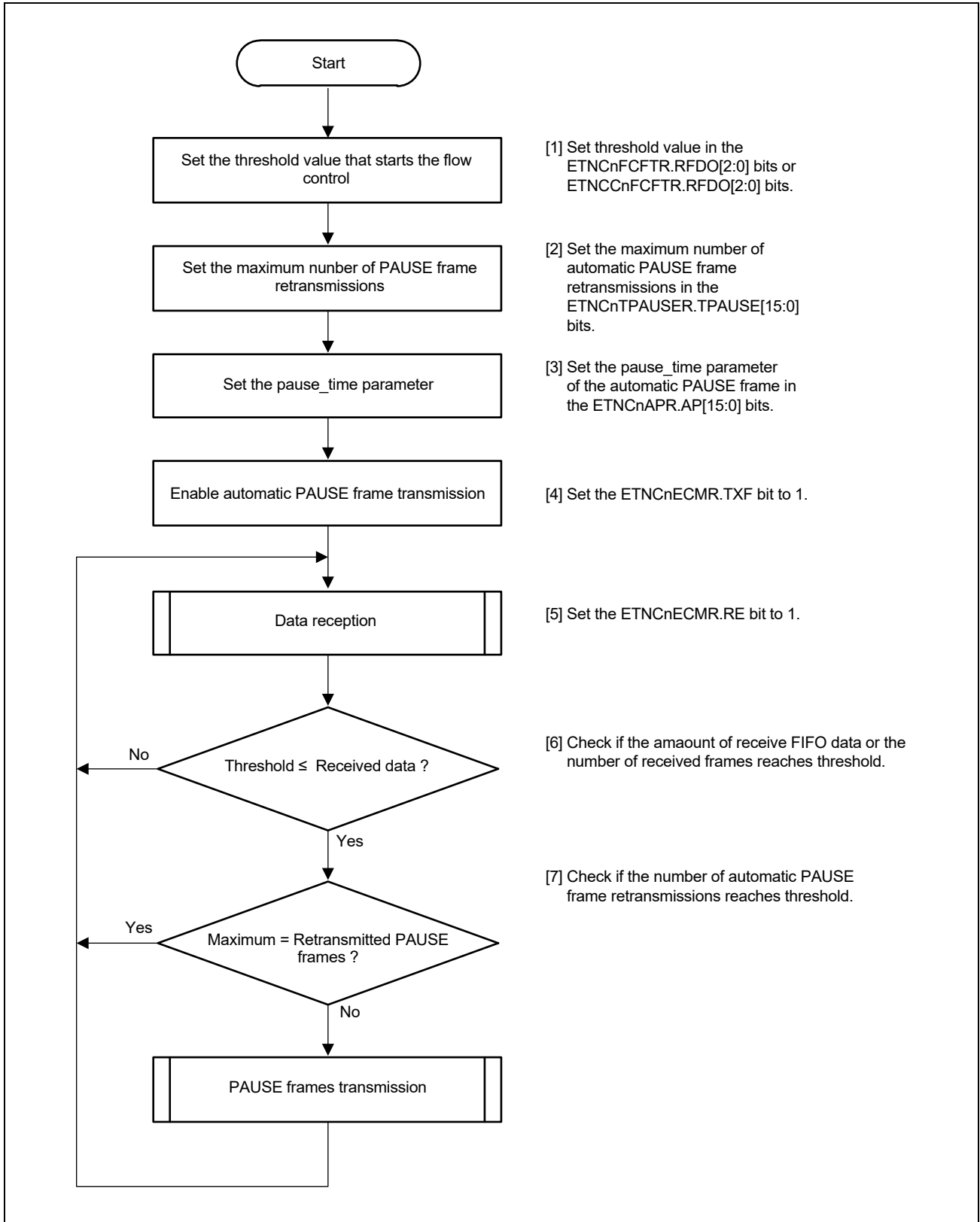


Figure 24.20 Example of Procedure to Set Automatic PAUSE Frame Transmission

(2) Manual PAUSE Frame Transmission

A PAUSE frame can be manually transmitted at any time. Writing the `pause_time` parameter of the PAUSE frame to the `ETNCnMPR.MP[15:0]` bits by software causes the ETHERC to transmit a PAUSE frame once. When transmitting a PAUSE frame more than once, write to the `ETNCnMPR.MP[15:0]` bits for each transmission.

(3) PAUSE Frame Reception

When setting the `ETNCnECMR.RXF` bit to 1, PAUSE frame detection is enabled. After a PAUSE frame is received, the ETHERC completes transmitting the current frame and waits for the PAUSE time of the received PAUSE frame to elapse before the next frame can be transmitted. Also, the ETHERC increments the `ETNCnRFCF.RPAUSE[7:0]` bit value.

However, while waiting for the PAUSE time to elapse, if a PAUSE frame that contains a `pause_time` parameter of 0 is received and the `ETNCnECMR.ZPF` bit is 1, the ETHERC becomes ready to transmit.

24.4.1.8 Interrupts

When a flag in the `ECSR` register becomes 1 and the corresponding bit in the `ETNCnECSIPR` register is 1, the ETHERC notifies the EDMAC of the status as an interrupt source. After receiving the notification, the EDMAC sets the `ETNCnEESR.ECI` flag to 1. When the `ETNCnEESIPR.ECIIP` bit is 1, the EDMAC sends an `INTETNCn` interrupt request to the CPU. See **Section 24.4.2, EDMAC Operations** for details.

24.4.2 EDMAC Operations

This section provides an overview of the EDMAC operations.

24.4.2.1 Descriptor Lists and Data Buffers

To transfer data using the EDMAC, create transmit and receive descriptor lists on the memory, set the start address of the transmit descriptor list to the ETNCnTDLAR register, and set the start address of the receive descriptor list to the RDLAR register. Also, transmit and receive buffers corresponding to each descriptor are required.

Align the descriptor list on the appropriate address boundary according to the descriptor length set by the ETNCnEDMR.DL[1:0] bits. The transmit buffer can be aligned on a long word boundary, word boundary, or byte boundary. When the valid transmit buffer size is 16 bytes or less, align it on a 32-byte boundary. Align the receive buffer on a 32-byte boundary. Set different addresses for the transmit and receive descriptors and buffers for EDMAC.

(1) Transmit Descriptor

Figure 24.21 shows the relationship between a transmit descriptor and the transmit buffer. A transmit descriptor consists of TD0 to TD2. The transmit frame and transmit buffer configuration can be selected from one buffer per frame (single-buffer frame transmission) or multiple buffers per frame (multi-buffer frame transmission) by setting the transmit descriptor.

Do not set transmission buffer length (TBL) to 32 bytes or less.

The value after reset of each transmission descriptor is 0000_H. If the transmission buffer length (TBL) is set to 0 to 32 bytes, operation indicated by the descriptor is not guaranteed.

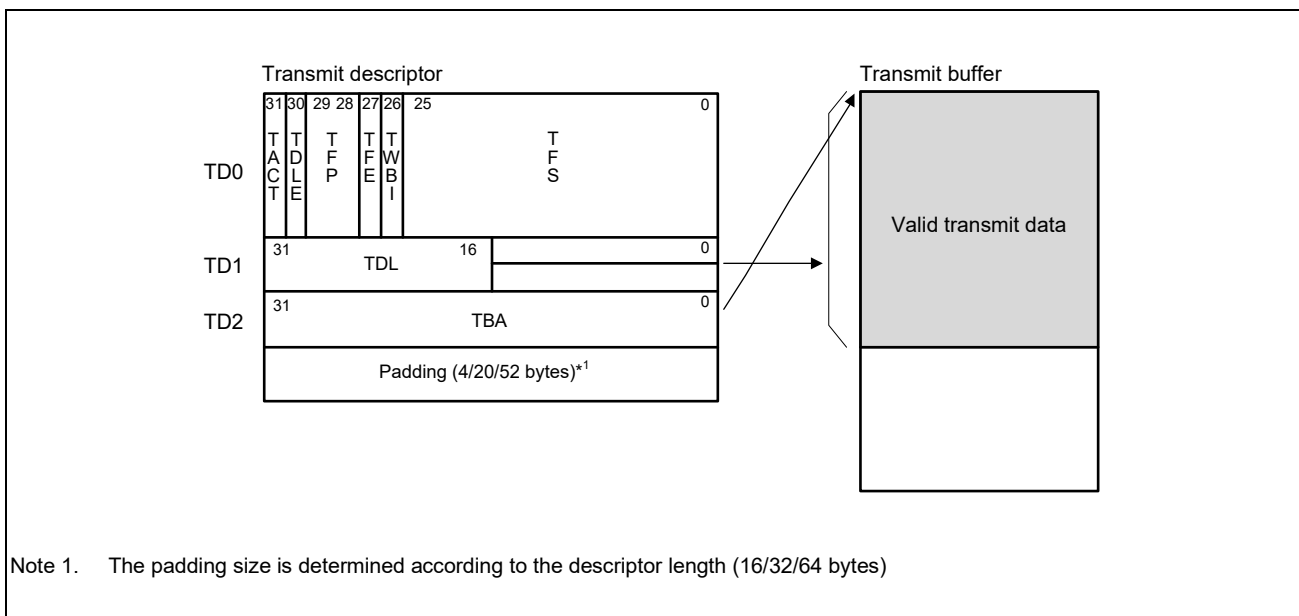


Figure 24.21 Relationship between Transmit Descriptor and Transmit Buffer

NOTE

A transmit descriptor ring has to be prepared for at least five buffers. If transmit descriptor rings are set to less than five buffers, the transmit descriptor rings may be looped two times by one transmit request, because fetching all descriptors in the ring finishes before the write back process of the first buffers transmission completion.

(a) Transmit Descriptor 0 (TD0)

Bit	Symbol	Bit Name	Description	R/W
b25 to b0	TFS	Transmit Frame Status	Set all bits to 0 when creating a descriptor. After write-back, the bits indicate the following: TFS26 to TFS9: Reserved TFS8: Transmit abort is detected (the value is equivalent to the ETNCnEESR.TABT flag). TFS7 to TFS4: Reserved TFS3: No carrier is detected (the value is equivalent to the ETNCnEESR.CND flag). TFS2: Loss of carrier is detected (the value is equivalent to the ETNCnEESR.DLC flag). TFS1: Late collision during transmission is detected (the value is equivalent to the ETNCnEESR.CD flag). TFS0: Transmit retry over (the value is equivalent to the ETNCnEESR.TRO flag). When each bit becomes 1, it indicates that the corresponding error has occurred during frame transmission. When any of the TFS bits becomes 1, the TFE bit also becomes 1. When any bit from TFS3 to TFS0 becomes 1, the TFS8 bit also becomes 1.	R/W
b26	TWBI	Write-Back Complete Interrupt Enable	(Enabled by setting TRIMD). 0: Interrupt does not occur when write-back to this descriptor is complete. 1: Interrupt occurs when write-back to this descriptor is complete.	R/W
b27	TFE	Transmit Frame Error	0: Frame transmission is successfully completed. 1: An error occurred during frame transmission (transmission aborted).	R/W
b29, b28	TFP[1:0]	Transmit Frame Position	b29 b28 0 0: Transmit buffer indicated by this descriptor is the middle of a transmit frame (frame information is incomplete). 0 1: Transmit buffer indicated by this descriptor is the end of a transmit frame (frame information is complete). 1 0: Transmit buffer indicated by this descriptor is the head of a transmit frame (frame information is incomplete). 1 1: Transmit buffer indicated by this descriptor is all of a transmit frame (one buffer per frame).	R/W
b30	TDLE	Transmit Descriptor List End	When this bit is 1, it indicates that this descriptor is the last descriptor of the descriptor list.	R/W
b31	TACT	Transmit Descriptor Valid	This bit indicates that this descriptor is valid.	R/W

Note: Bits for write-back are underlined.

TD0 indicates the transmit frame settings, and the status after transmission.

Transmit Frame Error (TFE Bit)

When the TFE bit is 1, it indicates that one or more of the TFS bits is 1.

Transmit Frame Position (TFP[1:0] Bits)

The TFP[1:0] bits indicate which part of a transmit frame corresponds to the transmit buffer indicated by this descriptor. The TFP[1:0] and TD1.TBL bit settings must be logically consistent in the previous and next descriptors.

Transmit Descriptor Valid (TACT Bit)

The TACT bit indicates that this descriptor is valid. The TACT bit is set to 1 by software. This bit becomes 0 when the transmit frame has been transferred or when the transmission is aborted.

(b) Transmit Descriptor 1 (TD1)

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	The read value is 0. The write value should be 0.	R
b31 to b16	TBL	Transmit Buffer Length	Set a valid byte length of the corresponding transmit buffer. Set a value equal to or greater than 1.	R/W

Note: TD1 is used to set a valid byte length of the transmit buffer.

(c) Transmit Descriptor 2 (TD2)

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TBA	Transmit Buffer Address	Set the start address of the transmit buffer. When the TD1.TBL bit value is 1 to 16 bytes, align it on a 32-byte boundary.	R/W

Note: TD2 is used to set the start address of the transmit buffer.

(2) Receive Descriptor

Figure 24.22 shows the relationship between a receive descriptor and the receive buffer. The receive frame and receive buffer configuration can be selected from one buffer per frame (single-buffer frame transmission) or multiple buffers per frame (multi-buffer frame transmission) by setting the receive descriptor. If the receive buffer length (RBL) is set to 0, operation indicated by the descriptor is not guaranteed.

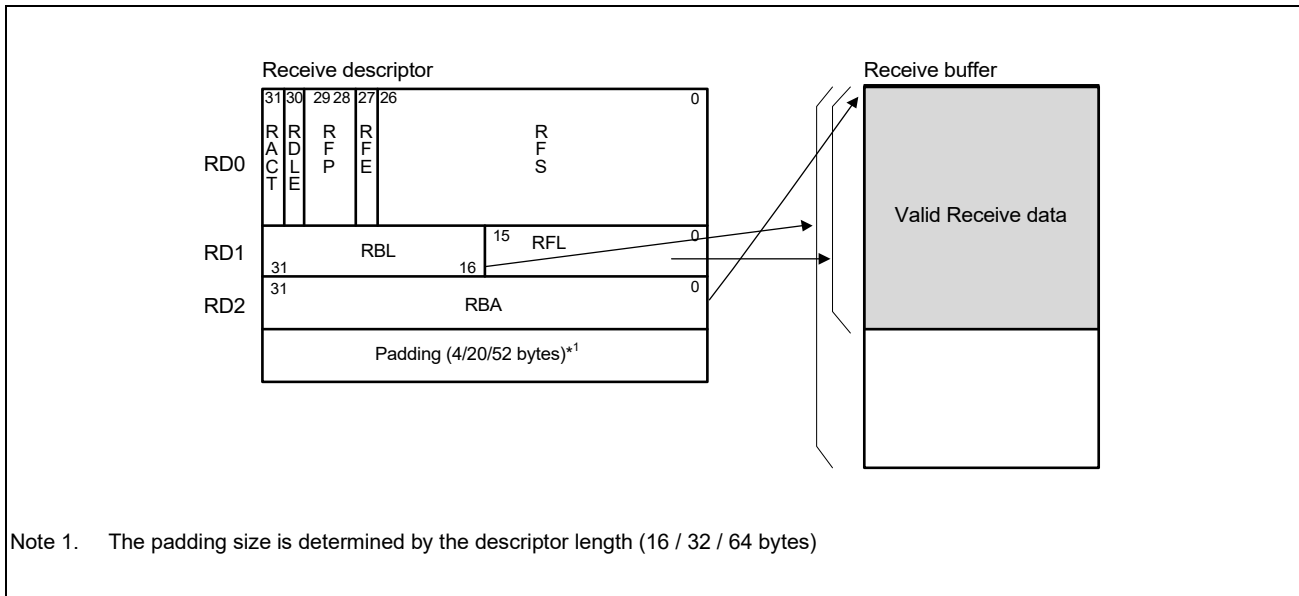


Figure 24.22 Relationship between Receive Descriptor and Receive Buffer

(a) Receive Descriptor 0 (RD0)

Bit	Symbol	Bit Name	Description	R/W
b26 to b0	RFS	Receive Frame Status	Set all bits to 0 when creating a descriptor. After write-back, the bits indicate the following: RFS26 to RFS10: Reserved RFS9: Receive FIFO overflow (the value is equivalent to the ETNCnEESR.RFOF flag). RFS8: Receive abort is detected (the value is equivalent to the ETNCnEESR.RABT flag). RFS7: Multicast address frame is received (the value is equivalent to the ETNCnEESR.RMAF flag). RFS6 and RFS5: Reserved RFS4: Alignment error is detected (the value is equivalent to the ETNCnEESR.RRF flag). RFS3: Frame-too-long error (the value is equivalent to the ETNCnEESR.RTLF flag) RFS2: Frame-too-short error (the value is equivalent to the ETNCnEESR.RTSF flag) RFS1: PHY-LSI receive error (the value is equivalent to the ETNCnEESR.PRE flag) RFS0: CRC error (the value is equivalent to the ETNCnEESR.CERF flag) When each bit becomes 1, it indicates that the corresponding error has occurred during frame reception. When any of the RFS bits becomes 1, the RFE bit also becomes 1 (set the ETNCnTRSCER register to select whether bits RFS7 and RFS4 are reflected in the RFE bit). When any bit from RFS3 to RFS0 becomes 1, the RFS8 bit also becomes 1.	R/W
b27	RFE	Receive Frame Error	0: No error has occurred in the received frame. 1: An error has occurred in the received frame.	R/W
b29, b28	RFP[1:0]	Receive Frame Position	b29 b28 0 0: Receive buffer indicated by this descriptor is the middle of a receive frame (frame information is incomplete). 0 1: Receive buffer indicated by this descriptor is the end of a receive frame (frame information is complete). 1 0: Receive buffer indicated by this descriptor is the head of a receive frame (frame information is incomplete). 1 1: Receive buffer indicated by this descriptor is all of a receive frame (one buffer per frame).	R/W
b30	RDLE	Receive Descriptor List End	When this bit is 1, it indicates that this descriptor is the last one of the descriptor list.	R/W
b31	RACT	Receive Descriptor Valid	Indicates that this descriptor is valid.	R/W

Note: Bits for write-back are underlined. RD0 indicates the receive frame status.

Receive Frame Error (RFE Bit)

When the RFE bit is 1, it indicates that one or more of the RFS bits is 1 (set the ETNCnTRSCER register to select whether bits RFS7 and RFS4 of ETNCn are reflected in the RFE bit).

Receive Frame Position (RFP[1:0] Bits)

The RFP[1:0] bits indicate which part of a receive frame corresponds to the receive buffer indicated by this descriptor.

Receive Descriptor Valid (RACT Bit)

The RACT bit indicates that this descriptor is valid. The RACT bit is set to 1 by software. This bit becomes 0 when all data has been transferred to the receive buffer indicated by RD2 or when the receive buffer becomes full.

(b) Receive Descriptor 1 (RD1)

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RFL	Receive Frame Length	These bits indicate the length (number of bytes) of the receive frame stored in the buffer. The number of bytes for padding set by the ETNCnRPADIR register is not included. These bits are written back to the descriptor corresponding to the end of a frame.	R/W
b31 to b16	RBL	Receive Buffer Length	These bits indicate the byte length of the corresponding receive buffer. Set an integral multiple of 32 as the buffer length.	R/W

Note: Bits for write-back are underlined.

RD1 indicates the receive buffer length. When reception is completed, the receive frame length is written back.

(c) Receive Descriptor 2 (RD2)

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RBA	Receive Buffer Address	These bits indicate the start address of the receive buffer. Align the buffer address on a 32-byte boundary.	R/W

Note: RD2 indicates the start address of the receive buffer.

24.4.2.2 Transmission

When the ETNCnEDTRR.TR bit is set to 1 while the ETNCnECMR.TE bit is 1, the EDMAC reads the descriptor following the previously used descriptor in the transmit descriptor list (or the descriptor indicated by the ETNCnTDLAR register after a reset). When the TACT bit is 1 in the transmit descriptor (TD0), the EDMAC sequentially reads transmit data from the start address of the transmit buffer indicated by transmit descriptor 2 (TD2) and transfers it to the ETHERC via the transmit FIFO. The ETHERC creates a transmit frame and starts transmission to the MII / RMII. When all data indicated by the TD1.TBL bit is transferred, write-back is performed according to the TD0.TFP[1:0] bits as follows:

- When the TD0.TFP[1:0] bits are 00_B or 10_B (frame information is incomplete), the TD0.TACT bit is written back.
- When the TD0.TFP[1:0] bits are 01_B or 11_B (frame information is complete), the TD0.TACT, TD0.TFS, and TD0.TFE bits are written back.

When the TD0.TACT bit in the read descriptor is 1, the EDMAC continues reading descriptors and transmits frames. When the TD0.TACT bit in the read descriptor is 0, the EDMAC sets the ETNCnEDTRR.TR bit to 0 and stops transmission.

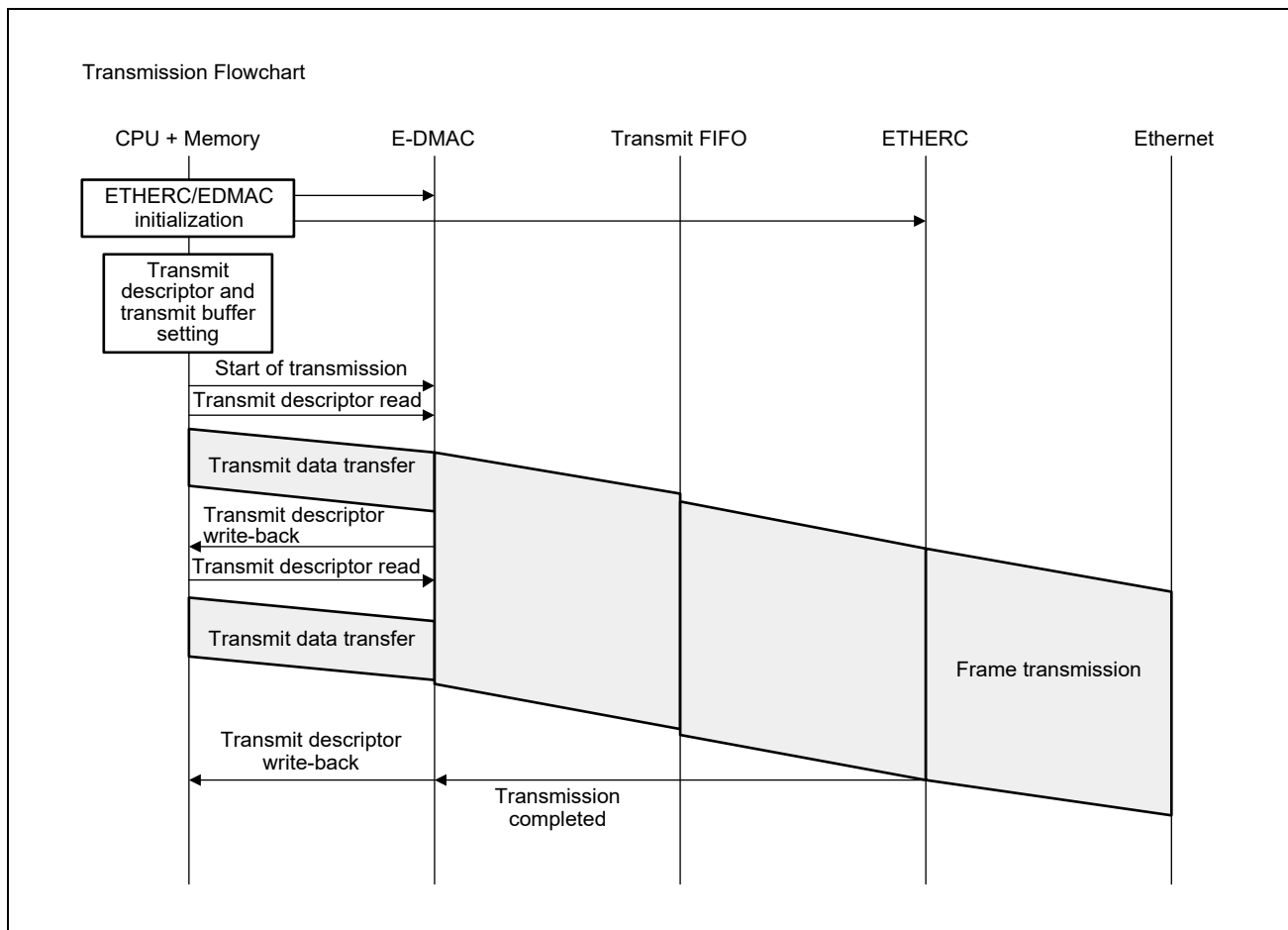


Figure 24.23 Example of Transmission Flow

24.4.2.3 Reception

When the ETNCnEDRRR.RR bit is set to 1 while the ETNCnECMR.RE bit is 1, the EDMAC reads the receive descriptor following the previously used descriptor (or descriptor indicated by the RDLAR register after a reset) and then waits for reception. While the RD0.RACT bit is 1, if the data stored in the receive FIFO is 32 bytes or more, or if the end byte of the frame is stored in the receive buffer, the EDMAC transfers data from the receive FIFO to the receive buffer indicated by receive descriptor 2 (RD2). If the data length of the received frame is longer than the buffer length set by the RBL bit in receive descriptor 1 (RD1), the EDMAC writes back 10_B or 00_B to the RD0.RFP[1:0] bits and 0 to the RD0.RACT bit when the receive buffer becomes full, and then the EDMAC reads the next data. After that, the EDMAC transfers data to another receive buffer. When the frame reception is complete or when the frame reception is aborted by an error, the EDMAC writes back 11_B or 01_B to the RD0.RFP[1:0] bits, 0 to the RD0.RACT bit, and the receive frame length to the RD1.RFL bit. When the ETNCnRMCR.RNR bit is 1, the EDMAC reads the next descriptor and waits for reception. When the RNR bit is 0, the EDMAC sets the ETNCnEDRRR.RR bit to 0 and stops reception.

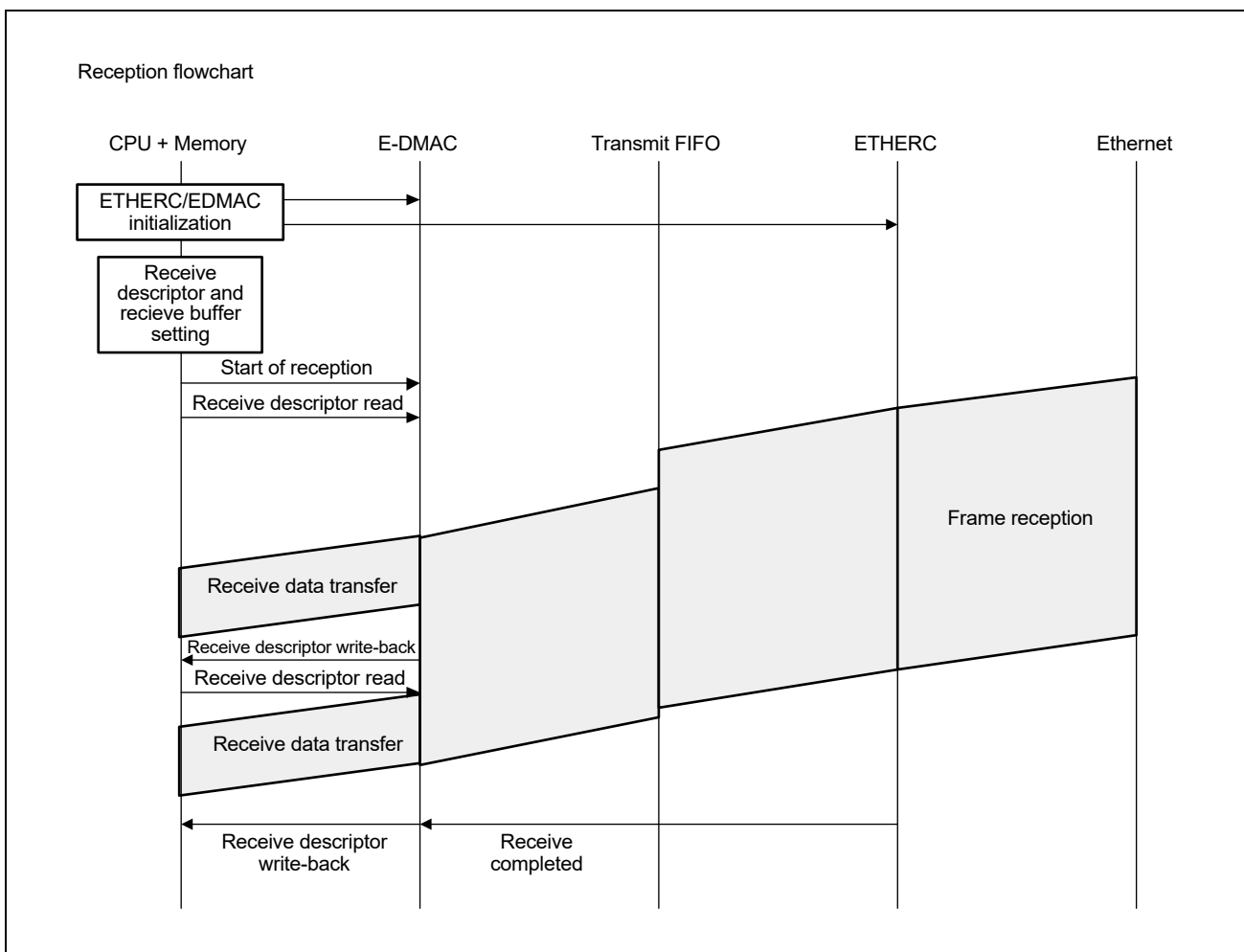


Figure 24.24 Example of Reception Flow

24.4.2.4 Multi-Buffer Frame Transmission

(1) Error Processing While Transmitting a Multi-Buffer Frame

If an error occurs during multi-buffer frame transmission, the EDMAC performs the processing shown in **Figure 24.25**.

In the figure, when the TACT bit of transmit descriptor 0 (TD0) is 0, the descriptor indicates that all data in the buffer has been successfully transmitted. When the TACT bit is 1, the descriptor indicates that data in the buffer has not yet been transmitted. If a frame transmit error*1 occurs in the head or middle of the frame while the TD0.TACT bit is 1, the EDMAC stops data transmission from the transmit FIFO and EDMAC data transfer, and sets the TD0.TACT bit to 0. After that, the EDMAC reads the next descriptor to see if the descriptor indicates the middle of the frame (TD0.TFP[1:0] bits are 00_B) or the end of the frame (TD0.TFP[1:0] bits are 01_B). When the descriptor indicates the middle of the frame, the EDMAC sets the TD0.TACT bit to 0 and reads the next descriptor. When the descriptor indicates the end of the frame, the EDMAC not only sets the TD0.TACT bit to 0, but also writes back to the TD0.TFE and TD0.TFS bits. After an error occurs, data in the buffer is not transmitted until write-back to the descriptor for the end of the frame. When the corresponding transmit error interrupt is enabled in the ETNCnEESIPR register, an interrupt request is generated immediately after write-back to the descriptor for the end of the frame.

Note 1. For ETNCn, these errors are a transmit retry-over condition error, late collision error, loss of carrier error, and a carrier not detected error.

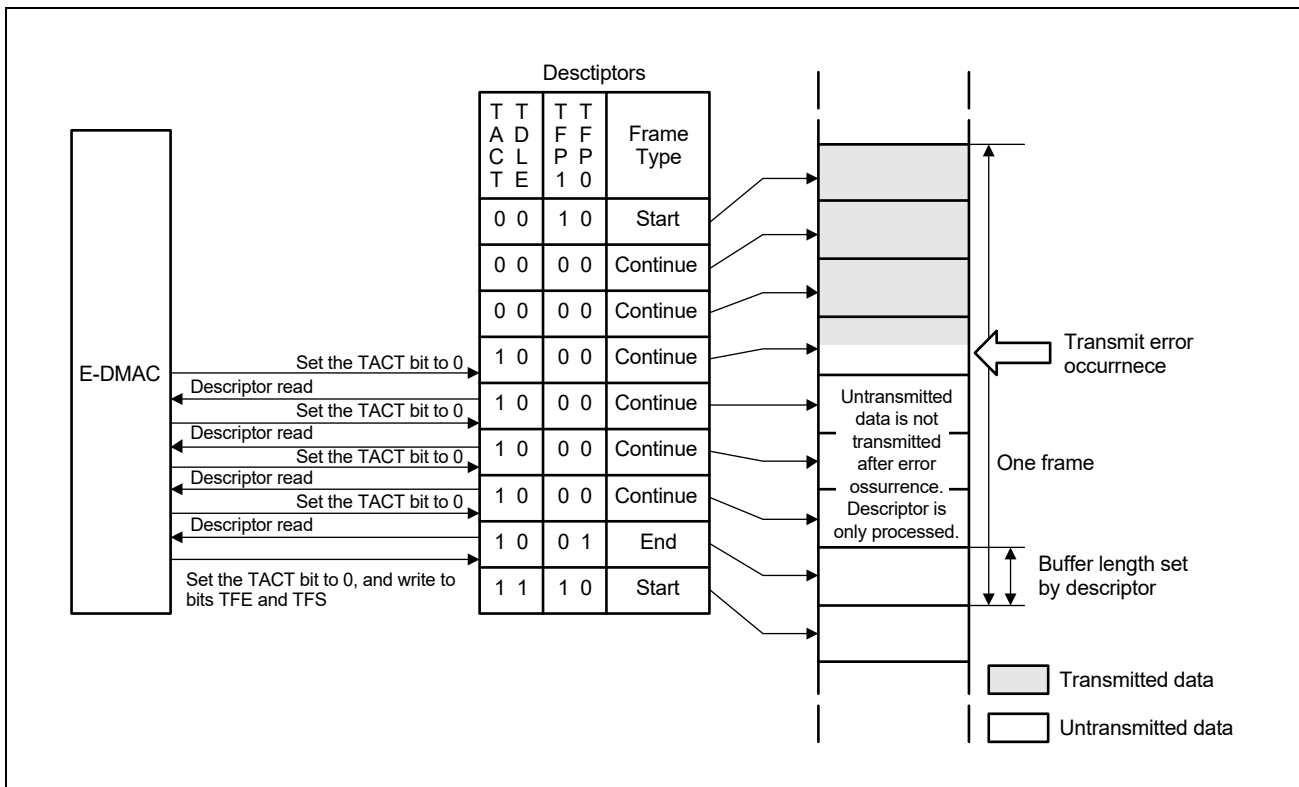


Figure 24.25 EDMAC Operation After a Transmit Error Occurs

(2) Error Processing While Receiving a Multi-Buffer Frame

If an error occurs during multi-buffer frame reception, the EDMAC performs the processing shown in **Figure 24.26**.

In the figure, when the RACT bit of receive descriptor 0 (RD0) is 0, the descriptor indicates that data has been successfully received in the buffer. When the RACT bit is 1, the descriptor indicates that data has not yet been received in the buffer. If a frame receive error*¹ occurs, the EDMAC stops receiving data of the frame, but it transfers data that has already been stored in the receive FIFO to the receive buffer.

When the receive buffer becomes full during transfer, the EDMAC sets the RACT bit to 0 and the RFP[1:0] bits to 10_B or 00_B and reads the next descriptor. After all data in the receive FIFO has been transferred, the EDMAC writes back the status to the descriptor.

When the corresponding receive error interrupt is enabled in the ETNCnEESIPR register, an interrupt request is generated immediately after write-back to the descriptor. When there is a request to receive a new frame, the EDMAC continues reception using the descriptor following the descriptor where the error occurred.

Note 1. For ETNCn, these errors are a CRC error, PHY-LSI receive error, frame-too-short error, frame-too-long error, and alignment error.

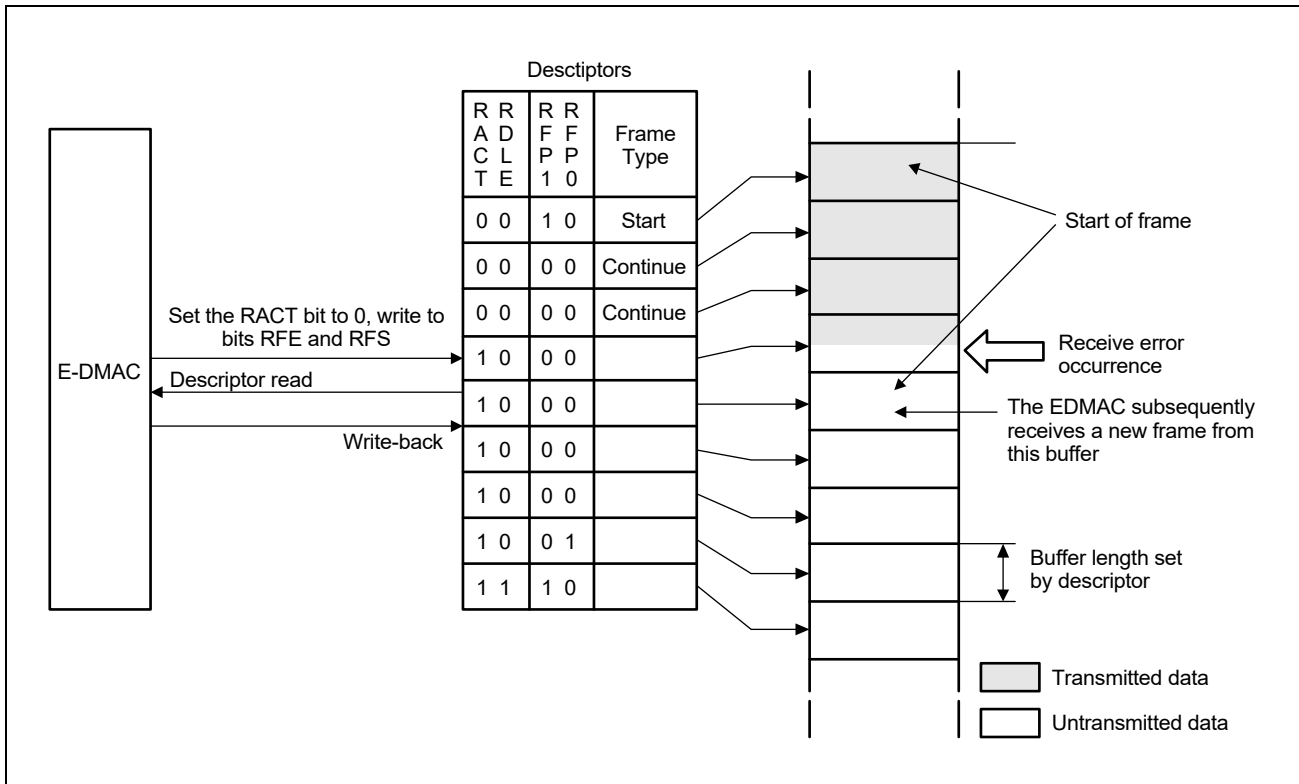


Figure 24.26 EDMAC Operation after Receive Error Occurs

24.4.2.5 Interrupts

When any of the status flags in the ETNCnEESR register becomes 1 while the corresponding interrupt request enable bit in the ETNCnEESIPR register is 1, ETNCn outputs an INTETNCn interrupt request to the CPU.

24.5 Usage Notes

24.5.1 Operation in Each Communication Mode

The bridge function of MII-to-RMII and RMII-to-MII is realized by converting the receive signal from RMII to MII and the transmit signal from MII to RMII by using the MII2RMII block. The operation is performed in multiple different communication modes depending on the setting of the mode signal.

Table 24.63 lists the mode settings in each communication mode.

Table 24.63 Operation State in Each Communication Mode

Communication Mode	I/F specifications	II	RMII	RMII	RMII	RMII
	Communication rate	10/100 Mbps	10 Mbps	10 Mbps	100 Mbps	100 Mbps
Communication system	Full/half duplex	Half duplex	Half duplex	Full duplex	Half duplex	Full duplex
Register setting	OPBT8.ETN_RMII_SEL	0	1			
	ETNCnECMR.RTM bit	—	0		1	
	ETNCnECMR.DM bit	—	0	1	0	1

24.5.2 Conditions for the LCHNG Flag to Become 1

The ETNCnECSR.LCHNG flag may become 1 even when the input level of the LINKSTA pin remains the same. In this case, a high level is input to the LINKSTA pin when setting the Pin function to assign the LINKSTA signal to a port or when releasing the ETHERC and EDMAC software reset using the ETNCnEDMR.SWR bit. The ETNCnECSR.LCHNG flag becomes 1 because the LINKSTA signal in the ETHERC is fixed to low level regardless of the level input to the external pin while the Pin function does not assign the LINKSTA signal or during the ETHERC and EDMAC software reset.

To avoid wrongly generating a link signal change interrupt, clear the ETNCnECSR.LCHNG flag and then set the ETNCnECSIPR.LCHNGIP bit to 1.

24.5.3 Input to the RMII_RX_ER Pin While the RMII is Selected

When the width of a reception error signal received from the PHY-LSI is only one cycle of the REF50CK clock (50 MHz) while the RMII is selected, the signal is not recognized as an error signal.

24.5.4 Stopping the EDMAC during Operations

When stopping EDMAC operations by using a sleep instruction or module-stop function while the EDMAC is running, confirm that the ETNCnEDTRR.TR and ETNCnEDRRR.RR bits are 0. If the EDMAC is stopped while the ETNCnEDTRR.TR or ETNCnEDRRR.RR bit is 1, the data for the frame that is being transmitted or received may not be complete, and EDMAC operation after exiting the sleep mode or module-stop state will not be guaranteed.

Section 25 Peripheral Sensor Interface 5 S (PSI5-S)

This section contains a generic description of peripheral sensor interface 5 S (PSI5-S).

The first part of this section describes the specific properties of this product, such as the number of channels and register base addresses.

The remainder of this section describes the functions and registers of PSI5-S.

25.1 Features of PSI5-S

25.1.1 Units and Channels

This product contains a PSI5-S with the number of channels shown below.

Table 25.1 Number of Channels

Product Name	RH850/E2x-FCC1 Series	
	373 Pins	292 Pins
Number of Channels	1	1
Name	PSI5S	

25.1.2 Register Base Address

The base address <PSI5S_base> of PSI5-S is shown listed in **Table 25.2**.

Table 25.2 Register Base Address <PSI5S_base>

PSI5-S Channel	PSI5-S Base Address	Bus Group
<PSI5S_base>	FFD5 0000 _H	Peripheral Group 3

25.1.3 Clock Supply

The clocks supplied by and to PSI5-S are listed in the following table.

Table 25.3 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
PSI5S	PCLK (bus clock)	CLK_HSB
	UART communication clock	CLK_HSB
	UART communication multiply clock	CLK_UHSB

25.1.4 Interrupt Requests

The PSI5-S interrupt requests are listed in the following table.

Table 25.4 Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number
INTPSI5SINTCH0	int_psis_ch0	Communication interrupt for CH0	396
INTPSI5SINTCH1	int_psis_ch1	Communication interrupt for CH1	397
INTPSI5SINTCH2	int_psis_ch2	Communication interrupt for CH2	398
INTPSI5SINTCH3	int_psis_ch3	Communication interrupt for CH3	399
INTPSI5SINTCH4	int_psis_ch4	Communication interrupt for CH4	400
INTPSI5SINTCH5	int_psis_ch5	Communication interrupt for CH5	401
INTPSI5SINTCH6	int_psis_ch6	Communication interrupt for CH6	402
INTPSI5SINTCH7	int_psis_ch7	Communication interrupt for CH7	403

The PSI5-S DMA requests are listed in the following table.

Table 25.5 DMA Requests

Interrupt Symbol Name	Unit Signal	Outline	sDMA Trigger Number	DTS Trigger Number
INTPSI5SRXDMACH0	dma_psis_ch0_rx	PSI5-S DMA request (CH0 RX)	group0-138	group1-46
INTPSI5SRXDMACH1	dma_psis_ch1_rx	PSI5-S DMA request (CH1 RX)	group0-139	group1-47
INTPSI5SRXDMACH2	dma_psis_ch2_rx	PSI5-S DMA request (CH2 RX)	group0-140	group1-48
INTPSI5SRXDMACH3	dma_psis_ch3_rx	PSI5-S DMA request (CH3 RX)	group0-141	group1-49
INTPSI5SRXDMACH4	dma_psis_ch4_rx	PSI5-S DMA request (CH4 RX)	group0-142	group1-50
INTPSI5SRXDMACH5	dma_psis_ch5_rx	PSI5-S DMA request (CH5 RX)	group0-143	group1-51
INTPSI5SRXDMACH6	dma_psis_ch6_rx	PSI5-S DMA request (CH6 RX)	group0-144	group1-52
INTPSI5SRXDMACH7	dma_psis_ch7_rx	PSI5-S DMA request (CH7 RX) / UART RX	group0-145	group1-53
INTPSI5STXDMACH1	dma_psis_ch1_tx	PSI5-S DMA request (CH1 TX)	group0-146	group1-54
INTPSI5STXDMACH2	dma_psis_ch2_tx	PSI5-S DMA request (CH2 TX)	group0-147	group1-55
INTPSI5STXDMACH3	dma_psis_ch3_tx	PSI5-S DMA request (CH3 TX)	group0-148	group1-56
INTPSI5STXDMACH4	dma_psis_ch4_tx	PSI5-S DMA request (CH4 TX)	group0-149	group1-57
INTPSI5STXDMACH5	dma_psis_ch5_tx	PSI5-S DMA request (CH5 TX)	group0-150	group1-58
INTPSI5STXDMACH6	dma_psis_ch6_tx	PSI5-S DMA request (CH6 TX)	group0-151	group1-59
INTPSI5STXDMACH7	dma_psis_ch7_tx	PSI5-S DMA request (CH7 TX) / UART TX	group0-152	group1-60

25.1.5 Reset Sources

Table 25.6 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
PSI5S	All registers	✓	✓	✓	✓	—	—

25.1.6 External Input/Output Signals

PSI5-S has the input/output pins listed in **Table 25.7**.

Table 25.7 Pin Configuration

Unit Signal Name	Outline	Alternative Port Pin Signal
PSI5S		
psis_rx_data	UART Rx data	PSISRX
psis_tx_data	UART Tx data	PSISTX
psis_tx_sclk	UART clock	PSISCLK

CAUTION

This AC specification is only applied to the specific pin groups. For details, refer to Appendix file "*Limited_conditions_for_AC_specification.xlsx*".

25.2 Outline of Functions

25.2.1 Functional Overview

PSI5-S performs UART communication with an external transceiver (PSI5 transceiver).

PSI5-S implements a peripheral sensor interface compliant with PSI5 Version 2.2 via the external transceiver.

This document describes the general specifications of PSI5-S.

The following table lists the features of PSI5-S.

Table 25.8 Features

Item	Specification
UART communication	Sampling clock output(*1): 6.67 MHz to 26.67 MHz Baud rate(*2): 1.333 Mbps to 5.333 Mbps Frame format (total of 10 or 11 bits) <ul style="list-style-type: none"> • Start bit: 1 bit • Data: 8 bits • Parity: None, even parity, 0 parity, or odd parity (can be specified separately for reception and transmission) • Stop bit: 1 bit
Sensor-to-ECU communication	<ul style="list-style-type: none"> • Possible reception of eight channels of frame data • Possible reception of 8-bit to 28-bit payload • Automatic calculation of CRC and parity values from payload • Possible storage of the CRC and parity bits attached to the data in received frames • Monitoring of the number of packets in received frames • Timestamp function for received messages. • Monitoring of the received frames by WDT.
ECU-to-sensor communication	<ul style="list-style-type: none"> • Automatic calculation of the CRC value to be added to frame data • Output format selectable from frame 1 to frame 4
Other interface	Interrupt output <ul style="list-style-type: none"> • Eight channels (Ch0 to Ch7) DMA request output <ul style="list-style-type: none"> • Eight channels (Ch0 to Ch7) for reception and seven channels (Ch1 to Ch7) for transmission

Note 1. For details about the clock specifications, see **Section 25.5.1, Common Setting**.

Note 2. For details about the baud rate of UART communication, **Section 25.5.1, Common Setting**.

25.2.1.1 Terms

Table 25.9 Abbreviations

HW	Hardware
SW	Software
PSI5	Peripheral Sensor Interface 5
UART	Universal Asynchronous Receiver Transmitter
Mbps	Megabits per second
SFR	Special Function Register
ECU	Engine Control Unit
Ch	Channel
Frm	Frame
Sync	Synchronous
Async	Asynchronous
Rx	Receive
Tx	Transmit
WDT	Watchdog Timer
MB	MailBox
INT	Interrupt
DDSR	Downstream Data Shift Register
GTM	Generic Timer Module

A “packet” means a UART frame.

A “packet frame” means a group of three to six packets (UART frames) that are received from a transceiver.

A “PSI5 frame” means data that is reconstructed from a packet frame after removing start, stop, and parity bits from the packet frame.

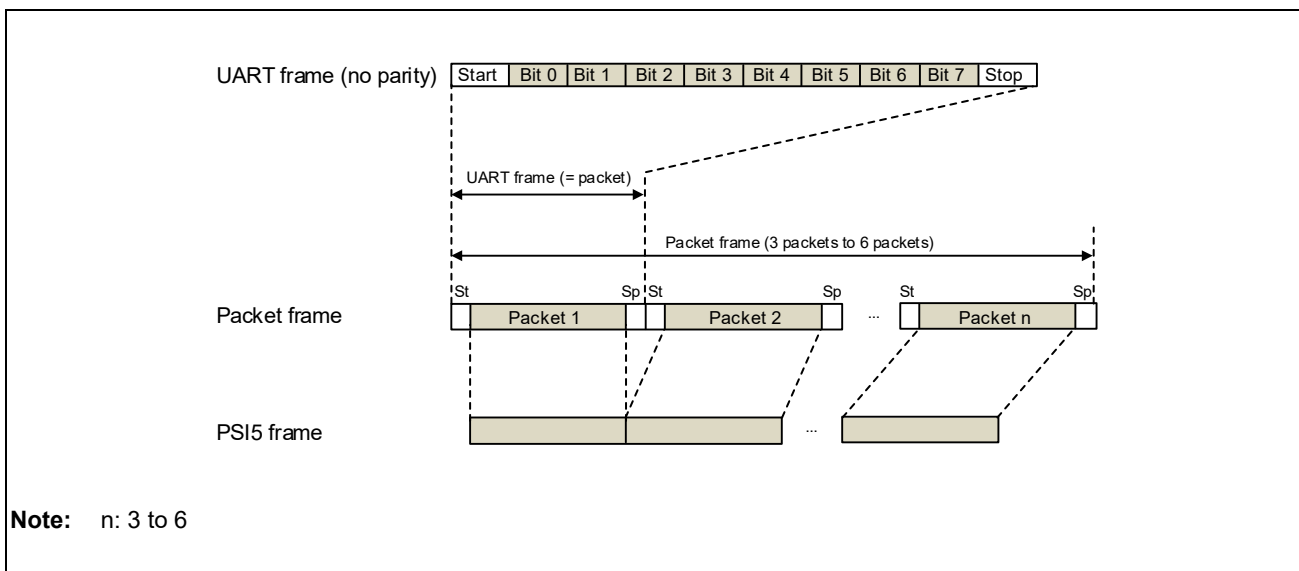


Figure 25.1 Definitions of Packet and Frames (without Parity)

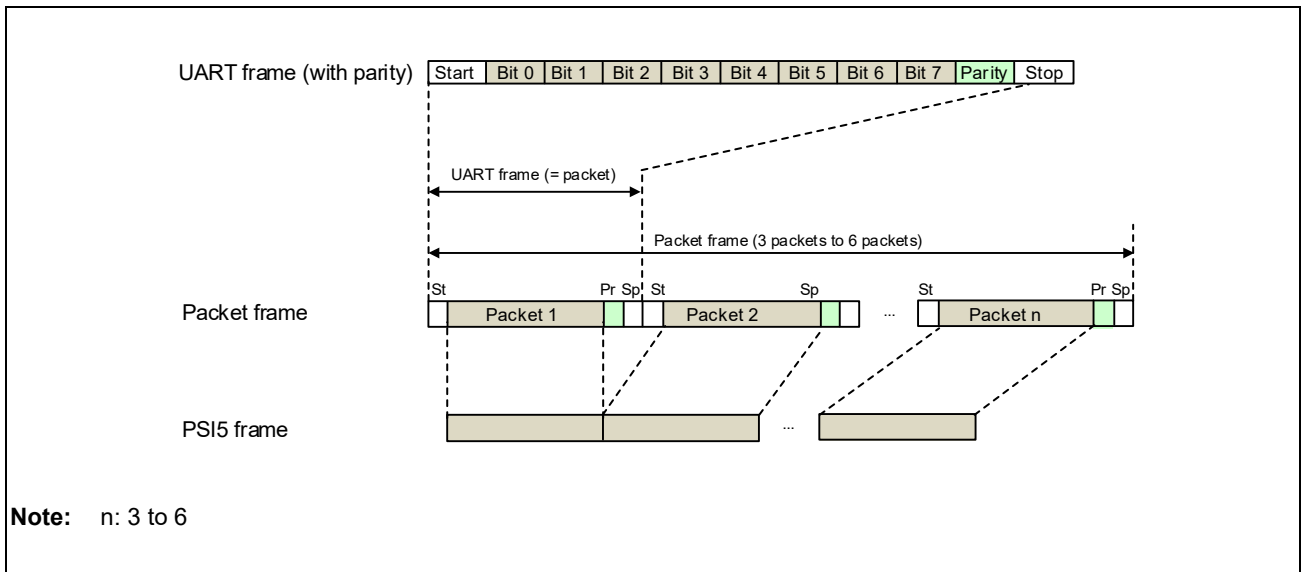


Figure 25.2 Definitions of Packet and Frames (with Parity)

The PSI5 frame format is as follows.

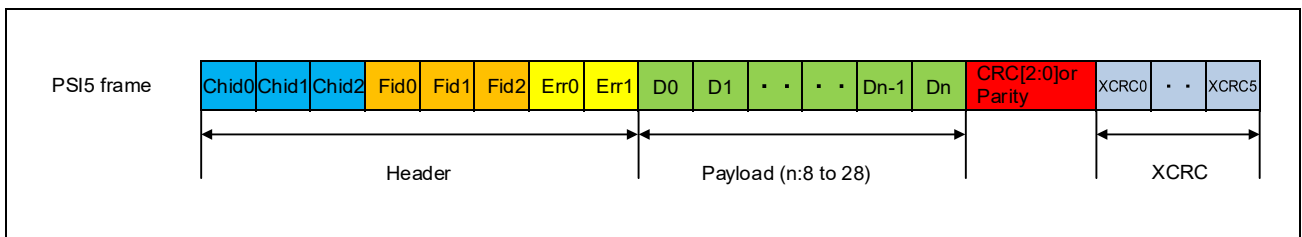


Figure 25.3 PSI5 Frame Format

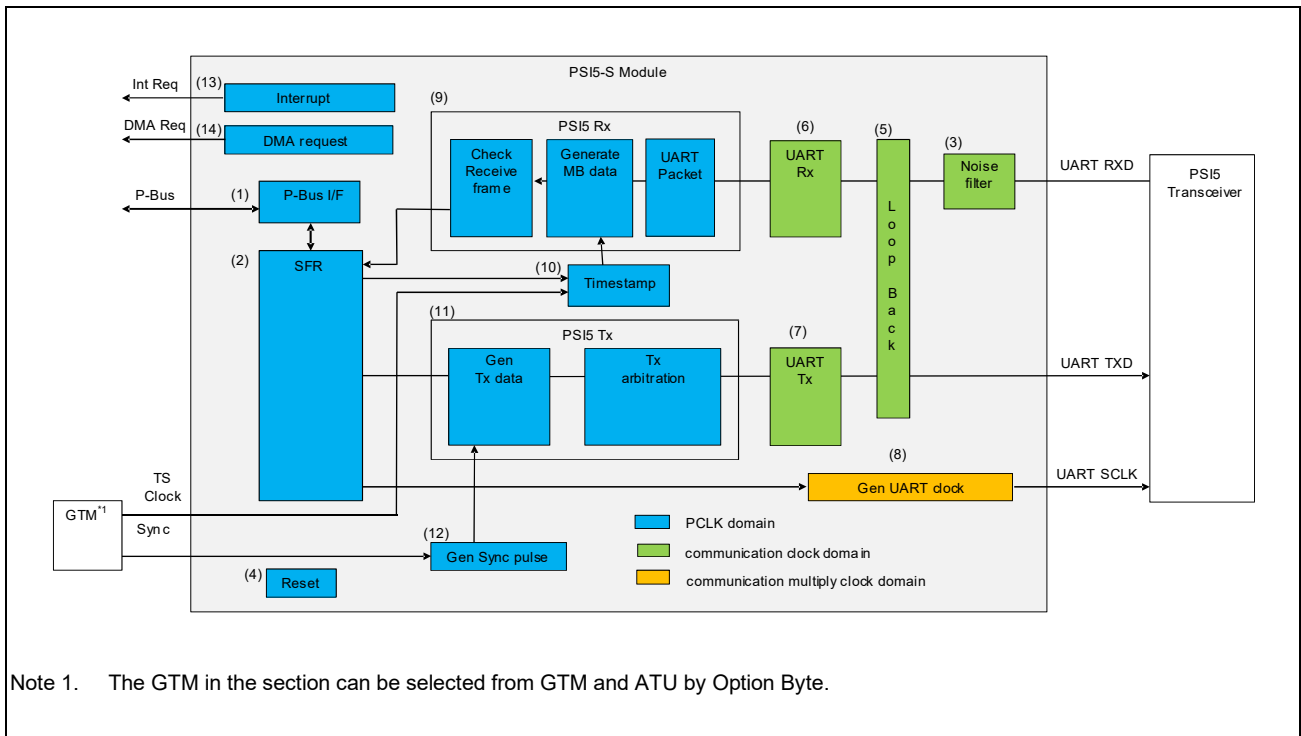


Figure 25.4 PSI5-S Block Diagram

The following table lists the functions of PSI5-S.

Table 25.10 PSI5-S Function List

No.	Name	Function
(1)	P-Bus interface	P-Bus interface (write/read access)
(2)	SFR	Special function registers (detect some kinds of errors)
(3)	Noise filter	Removes signals of less than 37.5 ns.
(4)	Reset	Generates an asynchronous reset via input reset signal or software setting.
(5)	Loopback	UART loopback setting
(6)	UART Rx	UART receiver
(7)	UART Tx	UART transmitter
(8)	Gen UART clock	Generates the UART communication clock.
(9)	PSI5 Rx	Restores PSI5 frame data from UART reception data. Checks for errors in reception data. PSI5 frame data is watched by WDT.
(10)	Timestamp	Generates a timestamp via the GTM or according to internal timing. When a header or synchronization pulse is received, the timestamp value is captured.
(11)	PSI5 Tx	Generates ECU-to-sensor data. When a sync pulse is input to the transceiver as command data, UART transmission is performed.
(12)	Gen Sync pulse	Generates a sync pulse when the input sync trigger is detected by the GTM.
(13)	Interrupt	Generates an interrupt signal when an interrupt factor occurs.
(14)	DMA request	Generates a DMA request signal when a DMAC request factor occurs.

25.3 Registers

25.3.1 List of Registers

25.3.1.1 Common Register/Config

The PSI5-S registers are listed in the following table.

For details on <PSI5S_base >, see **Section 25.1.2, Register Base Address**.

Table 25.11 List of Registers (1/13)

Module Name	Register Name	Symbol	Address	Access	Access Protection
PSI5S	PSI5-S/UART Operation Enable register	PSI5SPUOEB	<PSI5S_base> + 000 _H	8, 16, 32	—
	PSI5-S/UART Operation Mode register	PSI5SPUOMD	<PSI5S_base> + 004 _H	8, 16, 32	—
	PSI5-S/UART Operation Status register	PSI5SPUOS	<PSI5S_base> + 008 _H	8, 16, 32	—
	PSI5-S/UART Noise Filter Set register	PSI5SPUNFST	<PSI5S_base> + 00C _H	8, 16, 32	—
	PSI5-S/UART Software Reset register	PSI5SPUSWR	<PSI5S_base> + 010 _H	8, 16, 32	—
	PSI5-S Receive MailBox Data Clear register	PSI5SPRMBC	<PSI5S_base> + 014 _H	8, 16, 32	—
	PSI5-S/UART Communication Loop Back register	PSI5SPUCLB	<PSI5S_base> + 020 _H	8, 16, 32	—
	PSI5-S/UART Rx/Tx Parity Set register	PSI5SPUPTS	<PSI5S_base> + 024 _H	8, 16, 32	—
	PSI5-S/UART Baud Rate Clock Enable register	PSI5SPUBCE	<PSI5S_base> + 028 _H	8, 16, 32	—
	PSI5-S/UART Baud Rate Parameter register	PSI5SPUBPR	<PSI5S_base> + 02C _H	16, 32	—
	PSI5-S Timestamp Prescaler register	PSI5SPTPS	<PSI5S_base> + 030 _H	32	—
	PSI5-S Timestamp Counter A Select register	PSI5SPTCAS	<PSI5S_base> + 034 _H	8, 16, 32	—
	PSI5-S Timestamp Counter B Select register	PSI5SPTCBS	<PSI5S_base> + 038 _H	8, 16, 32	—
	PSI5-S Timestamp Counter A Enable register	PSI5SPTCAE	<PSI5S_base> + 040 _H	8, 16, 32	—
	PSI5-S Timestamp Counter A Clear register	PSI5SPTCAC	<PSI5S_base> + 044 _H	8, 16, 32	—
	PSI5-S Timestamp Counter B Enable register	PSI5SPTCBE	<PSI5S_base> + 048 _H	8, 16, 32	—
	PSI5-S Timestamp Counter B Clear register	PSI5SPTCBC	<PSI5S_base> + 04C _H	8, 16, 32	—
	PSI5-S All Timestamp Counter Enable register	PSI5SPATCE	<PSI5S_base> + 050 _H	8, 16, 32	—
	PSI5-S All Timestamp Counter Clear register	PSI5SPATCC	<PSI5S_base> + 054 _H	8, 16, 32	—
	UART Communication Rx Interrupt Enable	PSI5SUCRIE	<PSI5S_base> + 058 _H	8, 16, 32	—
	UART Communication Tx Interrupt Enable	PSI5SUCTIE	<PSI5S_base> + 05C _H	8, 16, 32	—

Table 25.11 List of Registers (2/13)

Module Name	Register Name	Symbol	Address	Access	Access Protection
PSI5S	UART Communication DMA Request Enable register	PSI5SUCDRE	<PSI5S_base> + 060 _H	8, 16, 32	—
	Access inhibit	—	—	—	—
	UART Communication Rx Data register	PSI5SUCRD	<PSI5S_base> + 070 _H	8, 16, 32	—
	UART Communication Rx Status register	PSI5SUCRS	<PSI5S_base> + 074 _H	8, 16, 32	—
	UART Communication Rx Status Clear register	PSI5SUCRSC	<PSI5S_base> + 078 _H	8, 16, 32	—
	PSI5-S Tx Frame Start register	PSI5SPTFST	<PSI5S_base> + 080 _H	8, 16, 32	—
	PSI5-S Tx Frame Number register	PSI5SPTFNM	<PSI5S_base> + 084 _H	8, 16, 32	—
	PSI5-S Tx Frame Data1 register	PSI5SPTFD1	<PSI5S_base> + 088 _H	8, 16, 32	—
	PSI5-S Tx Frame Data2 register	PSI5SPTFD2	<PSI5S_base> + 08C _H	8, 16, 32	—
	PSI5-S Tx Frame Status register	PSI5SPTFS	<PSI5S_base> + 090 _H	8, 16, 32	—
	PSI5-S Tx FIFO Status register	PSI5SPTFIS	<PSI5S_base> + 094 _H	8, 16, 32	—
	UART Communication Tx Data register	PSI5SUCTD	<PSI5S_base> + 0A0 _H	8, 16, 32	—
	UART Communication Tx Monitoring register	PSI5SUCTM	<PSI5S_base> + 0A4 _H	8, 16, 32	—
	UART Communication Tx Status register	PSI5SUCTS	<PSI5S_base> + 0A8 _H	8, 16, 32	—
	UART Communication Tx Status Clear register	PSI5SUCTSC	<PSI5S_base> + 0AC _H	8, 16, 32	—
	PSI5-S Receive Config1 ch0 register	PSI5SPRCF10	<PSI5S_base> + 100 _H	32	—
	PSI5-S Receive Config2 ch0 register	PSI5SPRCF20	<PSI5S_base> + 104 _H	32	—
	PSI5-S WDT Enable ch0 register	PSI5SPWDE0	<PSI5S_base> + 108 _H	8, 16, 32	—
	PSI5-S WDT Prescaler ch0 register	PSI5SPWDP0	<PSI5S_base> + 10C _H	16, 32	—
	PSI5-S WDT Expiration Value ch0 register	PSI5SPWDEV0	<PSI5S_base> + 110 _H	32	—
	PSI5-S CPU Interrupt Enable ch0	PSI5SPCIE0	<PSI5S_base> + 118 _H	16, 32	—
	PSI5-S DMA Transfer Request Enable ch0 register	PSI5SPDRE0	<PSI5S_base> + 11C _H	32	—
	PSI5-S Receive Error Status ch0 register	PSI5SPRES0	<PSI5S_base> + 130 _H	8, 16, 32	—
	PSI5-S Receive Error Status Clear ch0 register	PSI5SPRESC0	<PSI5S_base> + 134 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data ch0 register	PSI5SPTCDT0	<PSI5S_base> + 138 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data Clear ch0 register	PSI5SPTCDC0	<PSI5S_base> + 13C _H	8, 16, 32	—
	PSI5-S CPU Interrupt Status ch0 register	PSI5SPCIS0	<PSI5S_base> + 150 _H	16, 32	—
	PSI5-S CPU Interrupt Status Clear ch0 register	PSI5SPCISC0	<PSI5S_base> + 154 _H	16, 32	—
	PSI5-S Receive Config1 ch1 register	PSI5SPRCF11	<PSI5S_base> + 180 _H	32	—
	PSI5-S Receive Config2 ch1 register	PSI5SPRCF21	<PSI5S_base> + 184 _H	32	—
	PSI5-S WDT Enable ch1 register	PSI5SPWDE1	<PSI5S_base> + 188 _H	8, 16, 32	—

Table 25.11 List of Registers (3/13)

Module Name	Register Name	Symbol	Address	Access	Access Protection
PSI5S	PSI5-S WDT Prescaler ch1 register	PSI5SPWDP1	<PSI5S_base> + 18C _H	16, 32	—
	PSI5-S WDT Expiration Value ch1 register	PSI5SPWDEV1	<PSI5S_base> + 190 _H	32	—
	PSI5-S Tx Command Data ch1 register	PSI5SPTCD1	<PSI5S_base> + 194 _H	16, 32	—
	PSI5-S CPU Interrupt Enable ch1 register	PSI5SPCIE1	<PSI5S_base> + 198 _H	16, 32	—
	PSI5-S DMA Transfer Request Enable ch1 register	PSI5SPDRE1	<PSI5S_base> + 19C _H	32	—
	PSI5-S Sync Trigger Prescaler ch1 register	PSI5SPSTP1	<PSI5S_base> + 1A4 _H	16, 32	—
	PSI5-S Sync Trigger Expiration Value ch1 register	PSI5SPSTEV1	<PSI5S_base> + 1A8 _H	32	—
	PSI5-S Sync Trigger Select ch1 register	PSI5SPSTS1	<PSI5S_base> + 1AC _H	8, 16, 32	—
	PSI5-S Receive Error Status ch1 register	PSI5SPRES1	<PSI5S_base> + 1B0 _H	8, 16, 32	—
	PSI5-S Receive Error Status Clear ch1 register	PSI5SPRESC1	<PSI5S_base> + 1B4 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data ch1 register	PSI5SPTCDT1	<PSI5S_base> + 1B8 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data Clear ch1 register	PSI5SPTCDC1	<PSI5S_base> + 1BC _H	8, 16, 32	—
	PSI5-S DDSR Type ch1 register	PSI5SPDDTP1	<PSI5S_base> + 1C0 _H	8, 16, 32	—
	PSI5-S DDSR Data ch1 register	PSI5SPDDD1	<PSI5S_base> + 1C4 _H	32	—
	PSI5-S DDSR Status ch1 register	PSI5SPDDS1	<PSI5S_base> + 1C8 _H	8, 16, 32	—
	PSI5-S DDSR Stop ch1 register	PSI5SPDDSP1	<PSI5S_base> + 1CC _H	8, 16, 32	—
	PSI5-S CPU Interrupt Status ch1 register	PSI5SPCIS1	<PSI5S_base> + 1D0 _H	16, 32	—
	PSI5-S CPU Interrupt Status Clear ch1 register	PSI5SPCISC1	<PSI5S_base> + 1D4 _H	16, 32	—
	PSI5-S Receive Config1 ch2 register	PSI5SPRCF12	<PSI5S_base> + 200 _H	32	—
	PSI5-S Receive Config2 ch2 register	PSI5SPRCF22	<PSI5S_base> + 204 _H	32	—
	PSI5-S WDT Enable ch2 register	PSI5SPWDE2	<PSI5S_base> + 208 _H	8, 16, 32	—
	PSI5-S WDT Prescaler ch2 register	PSI5SPWDP2	<PSI5S_base> + 20C _H	16, 32	—
	PSI5-S WDT Expiration Value ch2 register	PSI5SPWDEV2	<PSI5S_base> + 210 _H	32	—
	PSI5-S Tx Command Data ch2 register	PSI5SPTCD2	<PSI5S_base> + 214 _H	16, 32	—
	PSI5-S CPU Interrupt Enable ch2 register	PSI5SPCIE2	<PSI5S_base> + 218 _H	16, 32	—
	PSI5-S DMA Transfer Request Enable ch2 register	PSI5SPDRE2	<PSI5S_base> + 21C _H	32	—
	PSI5-S Sync Trigger Prescaler ch2 register	PSI5SPSTP2	<PSI5S_base> + 224 _H	16, 32	—
	PSI5-S Sync Trigger Expiration Value ch2 register	PSI5SPSTEV2	<PSI5S_base> + 228 _H	32	—

Table 25.11 List of Registers (4/13)

Module Name	Register Name	Symbol	Address	Access	Access Protection
PSI5S	PSI5-S Sync Trigger Select ch2 register	PSI5SPSTS2	<PSI5S_base> + 22C _H	8, 16, 32	—
	PSI5-S Receive Error Status ch2 register	PSI5SPRES2	<PSI5S_base> + 230 _H	8, 16, 32	—
	PSI5-S Receive Error Status Clear ch2 register	PSI5SPRESC2	<PSI5S_base> + 234 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data ch2 register	PSI5SPTCDT2	<PSI5S_base> + 238 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data Clear ch2 register	PSI5SPTCDC2	<PSI5S_base> + 23C _H	8, 16, 32	—
	PSI5-S DDSR Type ch2 register	PSI5SPDDTP2	<PSI5S_base> + 240 _H	8, 16, 32	—
	PSI5-S DDSR Data ch2 register	PSI5SPDDD2	<PSI5S_base> + 244 _H	32	—
	PSI5-S DDSR Status ch2 register	PSI5SPDDS2	<PSI5S_base> + 248 _H	8, 16, 32	—
	PSI5-S DDSR Stop ch2 register	PSI5SPDDSP2	<PSI5S_base> + 24C _H	8, 16, 32	-
	PSI5-S CPU Interrupt Status ch2 register	PSI5SPCIS2	<PSI5S_base> + 250 _H	16, 32	—
	PSI5-S CPU Interrupt Status Clear ch2 register	PSI5SPCISC2	<PSI5S_base> + 254 _H	16, 32	—
	PSI5-S Receive Config1 ch3 register	PSI5SPRCF13	<PSI5S_base> + 280 _H	32	—
	PSI5-S Receive Config2 ch3 register	PSI5SPRCF23	<PSI5S_base> + 284 _H	32	—
	PSI5-S WDT Enable ch3 register	PSI5SPWDE3	<PSI5S_base> + 288 _H	8, 16, 32	—
	PSI5-S WDT Prescaler ch3 register	PSI5SPWDP3	<PSI5S_base> + 28C _H	16, 32	—
	PSI5-S WDT Expiration Value ch3 register	PSI5SPWDEV3	<PSI5S_base> + 290 _H	32	—
	PSI5-S Tx Command Data ch3 register	PSI5SPTCD3	<PSI5S_base> + 294 _H	16, 32	—
	PSI5-S CPU Interrupt Enable ch3 register	PSI5SPCIE3	<PSI5S_base> + 298 _H	16, 32	—
	PSI5-S DMA Transfer Request Enable ch3 register	PSI5SPDRE3	<PSI5S_base> + 29C _H	32	—
	PSI5-S Sync Trigger Prescaler ch3 register	PSI5SPSTP3	<PSI5S_base> + 2A4 _H	16, 32	—
	PSI5-S Sync Trigger Expiration Value ch3 register	PSI5SPSTEV3	<PSI5S_base> + 2A8 _H	32	—
	PSI5-S Sync Trigger Select ch3 register	PSI5SPSTS3	<PSI5S_base> + 2AC _H	8, 16, 32	—
	PSI5-S Receive Error Status ch3 register	PSI5SPRES3	<PSI5S_base> + 2B0 _H	8, 16, 32	—
	PSI5-S Receive Error Status Clear ch3 register	PSI5SPRESC3	<PSI5S_base> + 2B4 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data ch3 register	PSI5SPTCDT3	<PSI5S_base> + 2B8 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data Clear ch3 register	PSI5SPTCDC3	<PSI5S_base> + 2BC _H	8, 16, 32	—
	PSI5-S DDSR Type ch3 register	PSI5SPDDTP3	<PSI5S_base> + 2C0 _H	8, 16, 32	—
	PSI5-S DDSR Data ch3 register	PSI5SPDDD3	<PSI5S_base> + 2C4 _H	32	—
	PSI5-S DDSR Status ch3 register	PSI5SPDDS3	<PSI5S_base> + 2C8 _H	8, 16, 32	—

Table 25.11 List of Registers (5/13)

Module Name	Register Name	Symbol	Address	Access	Access Protection
PSI5S	PSI5-S DDSR Stop ch3 register	PSI5SPDDSP3	<PSI5S_base> + 2CC _H	8, 16, 32	—
	PSI5-S CPU Interrupt Status ch3 register	PSI5SPCIS3	<PSI5S_base> + 2D0 _H	16, 32	—
	PSI5-S CPU Interrupt Status Clear ch3 register	PSI5SPCISC3	<PSI5S_base> + 2D4 _H	16, 32	—
	PSI5-S Receive Config1 ch4 register	PSI5SPRCF14	<PSI5S_base> + 300 _H	32	—
	PSI5-S Receive Config2 ch4 register	PSI5SPRCF24	<PSI5S_base> + 304 _H	32	—
	PSI5-S WDT Enable ch4 register	PSI5SPWDE4	<PSI5S_base> + 308 _H	8, 16, 32	—
	PSI5-S WDT Prescaler ch4 register	PSI5SPWDP4	<PSI5S_base> + 30C _H	16, 32	—
	PSI5-S WDT Expiration Value ch4 register	PSI5SPWDEV4	<PSI5S_base> + 310 _H	32	—
	PSI5-S Tx Command Data ch4 register	PSI5SPTCD4	<PSI5S_base> + 314 _H	16, 32	—
	PSI5-S CPU Interrupt Enable ch4 register	PSI5SPCIE4	<PSI5S_base> + 318 _H	16, 32	—
	PSI5-S DMA Transfer Request Enable ch4 register	PSI5SPDRE4	<PSI5S_base> + 31C _H	32	—
	PSI5-S Sync Trigger Prescaler ch4 register	PSI5SPSTP4	<PSI5S_base> + 324 _H	16, 32	—
	PSI5-S Sync Trigger Expiration Value ch4 register	PSI5SPSTEV4	<PSI5S_base> + 328 _H	32	—
	PSI5-S Sync Trigger Select ch4 register	PSI5SPSTS4	<PSI5S_base> + 32C _H	8, 16, 32	—
	PSI5-S Receive Error Status ch4 register	PSI5SPRES4	<PSI5S_base> + 330 _H	8, 16, 32	—
	PSI5-S Receive Error Status Clear ch4 register	PSI5SPRESC4	<PSI5S_base> + 334 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data ch4 register	PSI5SPTCDT4	<PSI5S_base> + 338 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data Clear ch4 register	PSI5SPTCDC4	<PSI5S_base> + 33C _H	8, 16, 32	—
	PSI5-S DDSR Type ch4 register	PSI5SPDDTP4	<PSI5S_base> + 340 _H	8, 16, 32	—
	PSI5-S DDSR Data ch4 register	PSI5SPDDD4	<PSI5S_base> + 344 _H	32	—
	PSI5-S DDSR Status ch4 register	PSI5SPDDS4	<PSI5S_base> + 348 _H	8, 16, 32	—
	PSI5-S DDSR Stop ch4 register	PSI5SPDDSP4	<PSI5S_base> + 34C _H	8, 16, 32	—
	PSI5-S CPU Interrupt Status ch4 register	PSI5SPCIS4	<PSI5S_base> + 350 _H	16, 32	—
	PSI5-S CPU Interrupt Status Clear ch4 register	PSI5SPCISC4	<PSI5S_base> + 354 _H	16, 32	—
	PSI5-S Receive Config1 ch5 register	PSI5SPRCF15	<PSI5S_base> + 380 _H	32	—
	PSI5-S Receive Config2 ch5 register	PSI5SPRCF25	<PSI5S_base> + 384 _H	32	—
	PSI5-S WDT Enable ch5 register	PSI5SPWDE5	<PSI5S_base> + 388 _H	8, 16, 32	—
	PSI5-S WDT Prescaler ch5 register	PSI5SPWDP5	<PSI5S_base> + 38C _H	16, 32	—
	PSI5-S WDT Expiration Value ch5 register	PSI5SPWDEV5	<PSI5S_base> + 390 _H	32	—

Table 25.11 List of Registers (6/13)

Module Name	Register Name	Symbol	Address	Access	Access Protection
PSI5S	PSI5-S Tx Command Data ch5 register	PSI5SPTCD5	<PSI5S_base> + 394 _H	16, 32	—
	PSI5-S CPU Interrupt Enable ch5 register	PSI5SPCIE5	<PSI5S_base> + 398 _H	16, 32	—
	PSI5-S DMA Transfer Request Enable ch5 register	PSI5SPDRE5	<PSI5S_base> + 39C _H	32	—
	PSI5-S Sync Trigger Prescaler ch5 register	PSI5SPSTP5	<PSI5S_base> + 3A4 _H	16, 32	—
	PSI5-S Sync Trigger Expiration Value ch5 register	PSI5SPSTEV5	<PSI5S_base> + 3A8 _H	32	—
	PSI5-S Sync Trigger Select ch5 register	PSI5SPSTS5	<PSI5S_base> + 3AC _H	8, 16, 32	—
	PSI5-S Receive Error Status ch5 register	PSI5SPRES5	<PSI5S_base> + 3B0 _H	8, 16, 32	—
	PSI5-S Receive Error Status Clear ch5 register	PSI5SPRESC5	<PSI5S_base> + 3B4 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data ch5 register	PSI5SPTCDT5	<PSI5S_base> + 3B8 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data Clear ch5 register	PSI5SPTCDC5	<PSI5S_base> + 3BC _H	8, 16, 32	—
	PSI5-S DDSR Type ch5 register	PSI5SPDDTP5	<PSI5S_base> + 3C0 _H	8, 16, 32	—
	PSI5-S DDSR Data ch5 register	PSI5SPDDD5	<PSI5S_base> + 3C4 _H	32	—
	PSI5-S DDSR Status ch5 register	PSI5SPDDS5	<PSI5S_base> + 3C8 _H	8, 16, 32	—
	PSI5-S DDSR Stop ch5 register	PSI5SPDDSP5	<PSI5S_base> + 3CC _H	8, 16, 32	—
	PSI5-S CPU Interrupt Status ch5 register	PSI5SPCIS5	<PSI5S_base> + 3D0 _H	16, 32	—
	PSI5-S CPU Interrupt Status Clear ch5 register	PSI5SPCISC5	<PSI5S_base> + 3D4 _H	16, 32	—
	PSI5-S Receive Config1 ch6 register	PSI5SPRCF16	<PSI5S_base> + 400 _H	32	—
	PSI5-S Receive Config2 ch6 register	PSI5SPRCF26	<PSI5S_base> + 404 _H	32	—
	PSI5-S WDT Enable ch6 register	PSI5SPWDE6	<PSI5S_base> + 408 _H	8, 16, 32	—
	PSI5-S WDT Prescaler ch6 register	PSI5SPWDP6	<PSI5S_base> + 40C _H	16, 32	—
	PSI5-S WDT Expiration Value ch6 register	PSI5SPWDEV6	<PSI5S_base> + 410 _H	32	—
	PSI5-S Tx Command Data ch6 register	PSI5SPTCD6	<PSI5S_base> + 414 _H	16, 32	—
	PSI5-S CPU Interrupt Enable ch6 register	PSI5SPCIE6	<PSI5S_base> + 418 _H	16, 32	—
	PSI5-S DMA Transfer Request Enable ch6 register	PSI5SPDRE6	<PSI5S_base> + 41C _H	32	—
	PSI5-S Sync Trigger Prescaler ch6 register	PSI5SPSTP6	<PSI5S_base> + 424 _H	16, 32	—
	PSI5-S Sync Trigger Expiration Value ch6 register	PSI5SPSTEV6	<PSI5S_base> + 428 _H	32	—
	PSI5-S Sync Trigger Select ch6 register	PSI5SPSTS6	<PSI5S_base> + 42C _H	8, 16, 32	—

Table 25.11 List of Registers (7/13)

Module Name	Register Name	Symbol	Address	Access	Access Protection
PSI5S	PSI5-S Receive Error Status ch6 register	PSI5SPRES6	<PSI5S_base> + 430 _H	8, 16, 32	—
	PSI5-S Receive Error Status Clear ch6 register	PSI5SPRESC6	<PSI5S_base> + 434 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data ch6 register	PSI5SPTCDT6	<PSI5S_base> + 438 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data Clear ch6 register	PSI5SPTCDC6	<PSI5S_base> + 43C _H	8, 16, 32	—
	PSI5-S DDSR Type ch6 register	PSI5SPDDTP6	<PSI5S_base> + 440 _H	8, 16, 32	—
	PSI5-S DDSR Data ch6 register	PSI5SPDDD6	<PSI5S_base> + 444 _H	32	—
	PSI5-S DDSR Status ch6 register	PSI5SPDDS6	<PSI5S_base> + 448 _H	8, 16, 32	—
	PSI5-S DDSR Stop ch6 register	PSI5SPDDSP6	<PSI5S_base> + 44C _H	8, 16, 32	—
	PSI5-S CPU Interrupt Status ch6 register	PSI5SPCIS6	<PSI5S_base> + 450 _H	16, 32	—
	PSI5-S CPU Interrupt Status Clear ch6 register	PSI5SPCISC6	<PSI5S_base> + 454 _H	16, 32	—
	PSI5-S Receive Config1 ch7 register	PSI5SPRCF17	<PSI5S_base> + 480 _H	32	—
	PSI5-S Receive Config2 ch7 register	PSI5SPRCF27	<PSI5S_base> + 484 _H	32	—
	PSI5-S WDT Enable ch7 register	PSI5SPWDE7	<PSI5S_base> + 488 _H	8, 16, 32	—
	PSI5-S WDT Prescaler ch7 register	PSI5SPWDP7	<PSI5S_base> + 48C _H	16, 32	—
	PSI5-S WDT Expiration Value ch7 register	PSI5SPWDEV7	<PSI5S_base> + 490 _H	32	—
	PSI5-S Tx Command Data ch7 register	PSI5SPTCD7	<PSI5S_base> + 494 _H	16, 32	—
	PSI5-S CPU Interrupt Enable ch7 register	PSI5SPCIE7	<PSI5S_base> + 498 _H	16, 32	—
	PSI5-S DMA Transfer Request Enable ch7 register	PSI5SPDRE7	<PSI5S_base> + 49C _H	32	—
	PSI5-S Sync Trigger Prescaler ch7 register	PSI5SPSTP7	<PSI5S_base> + 4A4 _H	16, 32	—
	PSI5-S Sync Trigger Expiration Value ch7 register	PSI5SPSTEV7	<PSI5S_base> + 4A8 _H	32	—
	PSI5-S Sync Trigger Select ch7 register	PSI5SPSTS7	<PSI5S_base> + 4AC _H	8, 16, 32	—
	PSI5-S Receive Error Status ch7 register	PSI5SPRES7	<PSI5S_base> + 4B0 _H	8, 16, 32	—
	PSI5-S Receive Error Status Clear ch7 register	PSI5SPRESC7	<PSI5S_base> + 4B4 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data ch7 register	PSI5SPTCDT7	<PSI5S_base> + 4B8 _H	8, 16, 32	—
	PSI5-S Timestamp Capture Data Clear ch7 register	PSI5SPTCDC7	<PSI5S_base> + 4BC _H	8, 16, 32	—
	PSI5-S DDSR Type ch7 register	PSI5SPDDTP7	<PSI5S_base> + 4C0 _H	8, 16, 32	—
	PSI5-S DDSR Data ch7 register	PSI5SPDDD7	<PSI5S_base> + 4C4 _H	32	—
	PSI5-S DDSR Status ch7 register	PSI5SPDDS7	<PSI5S_base> + 4C8 _H	8, 16, 32	—
	PSI5-S DDSR Stop ch7 register	PSI5SPDDSP7	<PSI5S_base> + 4CC _H	8, 16, 32	—

Table 25.11 List of Registers (8/13)

Module Name	Register Name	Symbol	Address	Access	Access Protection
PSI5S	PSI5-S CPU Interrupt Status ch7 register	PSI5SPCIS7	<PSI5S_base> + 4D0 _H	16, 32	—
	PSI5-S CPU Interrupt Status Clear ch7 register	PSI5SPCISC7	<PSI5S_base> + 4D4 _H	16, 32	—
	PSI5-S Receive MailBox ch0 Frm1 Status register	PSI5SPMB01S	<PSI5S_base> + 500 _H	32	—
	PSI5-S Receive MailBox ch0 Frm1 Data register	PSI5SPMB01D	<PSI5S_base> + 504 _H	32	—
	PSI5-S Receive MailBox ch0 Frm1 Timestamp register	PSI5SPMB01T	<PSI5S_base> + 508 _H	32	—
	PSI5-S Receive MailBox ch0 Frm2 Status register	PSI5SPMB02S	<PSI5S_base> + 50C _H	32	—
	PSI5-S Receive MailBox ch0 Frm2 Data register	PSI5SPMB02D	<PSI5S_base> + 510 _H	32	—
	PSI5-S Receive MailBox ch0 Frm2 Timestamp register	PSI5SPMB02T	<PSI5S_base> + 514 _H	32	—
	PSI5-S Receive MailBox ch1 frm1 Status register	PSI5SPMB11S	<PSI5S_base> + 548 _H	32	—
	PSI5-S Receive MailBox ch1 frm1 Data register	PSI5SPMB11D	<PSI5S_base> + 54C _H	32	—
	PSI5-S Receive MailBox ch1 frm1 Timestamp register	PSI5SPMB11T	<PSI5S_base> + 550 _H	32	—
	PSI5-S Receive MailBox ch1 frm2 Status register	PSI5SPMB12S	<PSI5S_base> + 554 _H	32	—
	PSI5-S receive MailBox ch1 frm2 Data register	PSI5SPMB12D	<PSI5S_base> + 558 _H	32	—
	PSI5-S Receive MailBox ch1 frm2 Timestamp register	PSI5SPMB12T	<PSI5S_base> + 55C _H	32	—
	PSI5-S Receive MailBox ch1 frm3 Status register	PSI5SPMB13S	<PSI5S_base> + 560 _H	32	—
	PSI5-S Receive MailBox ch1 frm3 Data register	PSI5SPMB13D	<PSI5S_base> + 564 _H	32	—
	PSI5-S Receive MailBox ch1 frm3 Timestamp register	PSI5SPMB13T	<PSI5S_base> + 568 _H	32	—
	PSI5-S Receive MailBox ch1 frm4 Status register	PSI5SPMB14S	<PSI5S_base> + 56C _H	32	—
	PSI5-S Receive MailBox ch1 frm4 Data register	PSI5SPMB14D	<PSI5S_base> + 570 _H	32	—
	PSI5-S Receive MailBox ch1 frm4 Timestamp register	PSI5SPMB14T	<PSI5S_base> + 574 _H	32	—
	PSI5-S Receive MailBox ch1 frm5 Status register	PSI5SPMB15S	<PSI5S_base> + 578 _H	32	—
	PSI5-S Receive MailBox ch1 frm5 Data register	PSI5SPMB15D	<PSI5S_base> + 57C _H	32	—
	PSI5-S Receive MailBox ch1 frm5 Timestamp register	PSI5SPMB15T	<PSI5S_base> + 580 _H	32	—
	PSI5-S Receive MailBox ch1 frm6 Status register	PSI5SPMB16S	<PSI5S_base> + 584 _H	32	—

Table 25.11 List of Registers (9/13)

Module Name	Register Name	Symbol	Address	Access	Access Protection
PSI5S	PSI5-S Receive MailBox ch1 frm6 Data register	PSI5SPMB16D	<PSI5S_base> + 588 _H	32	—
	PSI5-S Receive MailBox ch1 frm6 Timestamp register	PSI5SPMB16T	<PSI5S_base> + 58C _H	32	—
	PSI5-S Receive MailBox ch2 frm1 Status register	PSI5SPMB21S	<PSI5S_base> + 590 _H	32	—
	PSI5-S Receive MailBox ch2 frm1 Data register	PSI5SPMB21D	<PSI5S_base> + 594 _H	32	—
	PSI5-S Receive MailBox ch2 frm1 Timestamp register	PSI5SPMB21T	<PSI5S_base> + 598 _H	32	—
	PSI5-S Receive MailBox ch2 frm2 Status register	PSI5SPMB22S	<PSI5S_base> + 59C _H	32	—
	PSI5-S Receive MailBox ch2 frm2 Data register	PSI5SPMB22D	<PSI5S_base> + 5A0 _H	32	—
	PSI5-S Receive MailBox ch2 frm2 Timestamp register	PSI5SPMB22T	<PSI5S_base> + 5A4 _H	32	—
	PSI5-S Receive MailBox ch2 frm3 Status register	PSI5SPMB23S	<PSI5S_base> + 5A8 _H	32	—
	PSI5-S Receive MailBox ch2 frm3 Data register	PSI5SPMB23D	<PSI5S_base> + 5AC _H	32	—
	PSI5-S receive MailBox ch2 frm3 Timestamp register	PSI5SPMB23T	<PSI5S_base> + 5B0 _H	32	—
	PSI5-S receive MailBox ch2 frm4 Status register	PSI5SPMB24S	<PSI5S_base> + 5B4 _H	32	—
	PSI5-S Receive MailBox ch2 frm4 Data register	PSI5SPMB24D	<PSI5S_base> + 5B8 _H	32	—
	PSI5-S Receive MailBox ch2 frm4 Timestamp register	PSI5SPMB24T	<PSI5S_base> + 5BC _H	32	—
	PSI5-S Receive MailBox ch2 frm5 Status register	PSI5SPMB25S	<PSI5S_base> + 5C0 _H	32	—
	PSI5-S Receive MailBox ch2 frm5 Data register	PSI5SPMB25D	<PSI5S_base> + 5C4 _H	32	—
	PSI5-S Receive MailBox ch2 frm5 Timestamp register	PSI5SPMB25T	<PSI5S_base> + 5C8 _H	32	—
	PSI5-S Receive MailBox ch2 frm6 Status register	PSI5SPMB26S	<PSI5S_base> + 5CC _H	32	—
	PSI5-S Receive MailBox ch2 frm6 Data register	PSI5SPMB26D	<PSI5S_base> + 5D0 _H	32	—
	PSI5-S Receive MailBox ch2 frm6 Timestamp register	PSI5SPMB26T	<PSI5S_base> + 5D4 _H	32	—
	PSI5-S Receive MailBox ch3 frm1 Status register	PSI5SPMB31S	<PSI5S_base> + 5D8 _H	32	—
	PSI5-S Receive MailBox ch3 frm1 Data register	PSI5SPMB31D	<PSI5S_base> + 5DC _H	32	—
	PSI5-S Receive MailBox ch3 frm1 Timestamp register	PSI5SPMB31T	<PSI5S_base> + 5E0 _H	32	—
	PSI5-S Receive MailBox ch3 frm2 Status register	PSI5SPMB32S	<PSI5S_base> + 5E4 _H	32	—

Table 25.11 List of Registers (10/13)

Module Name	Register Name	Symbol	Address	Access	Access Protection
PSI5S	PSI5-S Receive MailBox ch3 frm2 Data register	PSI5SPMB32D	<PSI5S_base> + 5E8 _H	32	—
	PSI5-S Receive MailBox ch3 frm2 Timestamp register	PSI5SPMB32T	<PSI5S_base> + 5EC _H	32	—
	PSI5-S Receive MailBox ch3 frm3 Status register	PSI5SPMB33S	<PSI5S_base> + 5F0 _H	32	—
	PSI5-S Receive MailBox ch3 frm3 Data register	PSI5SPMB33D	<PSI5S_base> + 5F4 _H	32	—
	PSI5-S Receive MailBox ch3 frm3 Timestamp register	PSI5SPMB33T	<PSI5S_base> + 5F8 _H	32	—
	PSI5-S Receive MailBox ch3 frm4 Status register	PSI5SPMB34S	<PSI5S_base> + 5FC _H	32	—
	PSI5-S Receive MailBox ch3 frm4 Data register	PSI5SPMB34D	<PSI5S_base> + 600 _H	32	—
	PSI5-S Receive MailBox ch3 frm4 Timestamp register	PSI5SPMB34T	<PSI5S_base> + 604 _H	32	—
	PSI5-S Receive MailBox ch3 frm5 Status register	PSI5SPMB35S	<PSI5S_base> + 608 _H	32	—
	PSI5-S Receive MailBox ch3 frm5 Data register	PSI5SPMB35D	<PSI5S_base> + 60C _H	32	—
	PSI5-S Receive MailBox ch3 frm5 Timestamp register	PSI5SPMB35T	<PSI5S_base> + 610 _H	32	—
	PSI5-S Receive MailBox ch3 frm6 Status register	PSI5SPMB36S	<PSI5S_base> + 614 _H	32	—
	PSI5-S Receive MailBox ch3 frm6 Data register	PSI5SPMB36D	<PSI5S_base> + 618 _H	32	—
	PSI5-S Receive MailBox ch3 frm6 Timestamp register	PSI5SPMB36T	<PSI5S_base> + 61C _H	32	—
	PSI5-S Receive MailBox ch4 frm1 Status register	PSI5SPMB41S	<PSI5S_base> + 620 _H	32	—
	PSI5-S Receive MailBox ch4 frm1 Data register	PSI5SPMB41D	<PSI5S_base> + 624 _H	32	—
	PSI5-S Receive MailBox ch4 frm1 Timestamp register	PSI5SPMB41T	<PSI5S_base> + 628 _H	32	—
	PSI5-S Receive MailBox ch4 frm2 Status register	PSI5SPMB42S	<PSI5S_base> + 62C _H	32	—
	PSI5-S Receive MailBox ch4 frm2 Data register	PSI5SPMB42D	<PSI5S_base> + 630 _H	32	—
	PSI5-S Receive MailBox ch4 frm2 Timestamp register	PSI5SPMB42T	<PSI5S_base> + 634 _H	32	—
	PSI5-S Receive MailBox ch4 frm3 Status register	PSI5SPMB43S	<PSI5S_base> + 638 _H	32	—
	PSI5-S Receive MailBox ch4 frm3 Data register	PSI5SPMB43D	<PSI5S_base> + 63C _H	32	—
	PSI5-S Receive MailBox ch4 frm3 Timestamp register	PSI5SPMB43T	<PSI5S_base> + 640 _H	32	—
	PSI5-S Receive MailBox ch4 frm4 Status register	PSI5SPMB44S	<PSI5S_base> + 644 _H	32	—

Table 25.11 List of Registers (11/13)

Module Name	Register Name	Symbol	Address	Access	Access Protection
PSI5S	PSI5-S Receive MailBox ch4 frm4 Data register	PSI5SPMB44D	<PSI5S_base> + 648 _H	32	—
	PSI5-S Receive MailBox ch4 frm4 Timestamp register	PSI5SPMB44T	<PSI5S_base> + 64C _H	32	—
	PSI5-S Receive MailBox ch4 frm5 Status register	PSI5SPMB45S	<PSI5S_base> + 650 _H	32	—
	PSI5-S Receive MailBox ch4 frm5 Data register	PSI5SPMB45D	<PSI5S_base> + 654 _H	32	—
	PSI5-S Receive MailBox ch4 frm5 Timestamp register	PSI5SPMB45T	<PSI5S_base> + 658 _H	32	—
	PSI5-S Receive MailBox ch4 frm6 Status register	PSI5SPMB46S	<PSI5S_base> + 65C _H	32	—
	PSI5-S Receive MailBox ch4 frm6 Data register	PSI5SPMB46D	<PSI5S_base> + 660 _H	32	—
	PSI5-S Receive MailBox ch4 frm6 Timestamp register	PSI5SPMB46T	<PSI5S_base> + 664 _H	32	—
	PSI5-S Receive MailBox ch5 frm1 Status register	PSI5SPMB51S	<PSI5S_base> + 668 _H	32	—
	PSI5-S Receive MailBox ch5 frm1 Data register	PSI5SPMB51D	<PSI5S_base> + 66C _H	32	—
	PSI5-S Receive MailBox ch5 frm1 Timestamp register	PSI5SPMB51T	<PSI5S_base> + 670 _H	32	—
	PSI5-S Receive MailBox ch5 frm2 Status register	PSI5SPMB52S	<PSI5S_base> + 674 _H	32	—
	PSI5-S Receive MailBox ch5 frm2 Data register	PSI5SPMB52D	<PSI5S_base> + 678 _H	32	—
	PSI5-S Receive MailBox ch5 frm2 Timestamp register	PSI5SPMB52T	<PSI5S_base> + 67C _H	32	—
	PSI5-S Receive MailBox ch5 frm3 Status register	PSI5SPMB53S	<PSI5S_base> + 680 _H	32	—
	PSI5-S Receive MailBox ch5 frm3 Data register	PSI5SPMB53D	<PSI5S_base> + 684 _H	32	—
	PSI5-S Receive MailBox ch5 frm3 Timestamp register	PSI5SPMB53T	<PSI5S_base> + 688 _H	32	—
	PSI5-S Receive MailBox ch5 frm4 Status register	PSI5SPMB54S	<PSI5S_base> + 68C _H	32	—
	PSI5-S Receive MailBox ch5 frm4 Data register	PSI5SPMB54D	<PSI5S_base> + 690 _H	32	—
	PSI5-S Receive MailBox ch5 frm4 Timestamp register	PSI5SPMB54T	<PSI5S_base> + 694 _H	32	—
	PSI5-S Receive MailBox ch5 frm5 Status register	PSI5SPMB55S	<PSI5S_base> + 698 _H	32	—
	PSI5-S Receive MailBox ch5 frm5 Data register	PSI5SPMB55D	<PSI5S_base> + 69C _H	32	—
	PSI5-S Receive MailBox ch5 frm5 Timestamp register	PSI5SPMB55T	<PSI5S_base> + 6A0 _H	32	—
	PSI5-S Receive MailBox ch5 frm6 Status register	PSI5SPMB56S	<PSI5S_base> + 6A4 _H	32	—

Table 25.11 List of Registers (12/13)

Module Name	Register Name	Symbol	Address	Access	Access Protection
PSI5S	PSI5-S Receive MailBox ch5 frm6 Data register	PSI5SPMB56D	<PSI5S_base> + 6A8 _H	32	—
	PSI5-S Receive MailBox ch5 frm6 Timestamp register	PSI5SPMB56T	<PSI5S_base> + 6AC _H	32	—
	PSI5-S Receive MailBox ch6 frm1 Status register	PSI5SPMB61S	<PSI5S_base> + 6B0 _H	32	—
	PSI5-S Receive MailBox ch6 frm1 Data register	PSI5SPMB61D	<PSI5S_base> + 6B4 _H	32	—
	PSI5-S Receive MailBox ch6 frm1 Timestamp register	PSI5SPMB61T	<PSI5S_base> + 6B8 _H	32	—
	PSI5-S Receive MailBox ch6 frm2 Status register	PSI5SPMB62S	<PSI5S_base> + 6BC _H	32	—
	PSI5-S Receive MailBox ch6 frm2 Data register	PSI5SPMB62D	<PSI5S_base> + 6C0 _H	32	—
	PSI5-S Receive MailBox ch6 frm2 Timestamp register	PSI5SPMB62T	<PSI5S_base> + 6C4 _H	32	—
	PSI5-S Receive MailBox ch6 frm3 Status register	PSI5SPMB63S	<PSI5S_base> + 6C8 _H	32	—
	PSI5-S Receive MailBox ch6 frm3 Data register	PSI5SPMB63D	<PSI5S_base> + 6CC _H	32	—
	PSI5-S Receive MailBox ch6 frm3 Timestamp register	PSI5SPMB63T	<PSI5S_base> + 6D0 _H	32	—
	PSI5-S Receive MailBox ch6 frm4 Status register	PSI5SPMB64S	<PSI5S_base> + 6D4 _H	32	—
	PSI5-S Receive MailBox ch6 frm4 Data register	PSI5SPMB64D	<PSI5S_base> + 6D8 _H	32	—
	PSI5-S Receive MailBox ch6 frm4 Timestamp register	PSI5SPMB64T	<PSI5S_base> + 6DC _H	32	—
	PSI5-S Receive MailBox ch6 frm5 Status register	PSI5SPMB65S	<PSI5S_base> + 6E0 _H	32	—
	PSI5-S Receive MailBox ch6 frm5 Data register	PSI5SPMB65D	<PSI5S_base> + 6E4 _H	32	—
	PSI5-S Receive MailBox ch6 frm5 Timestamp register	PSI5SPMB65T	<PSI5S_base> + 6E8 _H	32	—
	PSI5-S Receive MailBox ch6 frm6 Status register	PSI5SPMB66S	<PSI5S_base> + 6EC _H	32	—
	PSI5-S Receive MailBox ch6 frm6 Data register	PSI5SPMB66D	<PSI5S_base> + 6F0 _H	32	—
	PSI5-S Receive MailBox ch6 frm6 Timestamp register	PSI5SPMB66T	<PSI5S_base> + 6F4 _H	32	—
	PSI5-S Receive MailBox ch7 frm1 Status register	PSI5SPMB71S	<PSI5S_base> + 6F8 _H	32	—
	PSI5-S Receive MailBox ch7 frm1 Data register	PSI5SPMB71D	<PSI5S_base> + 6FC _H	32	—
	PSI5-S Receive MailBox ch7 frm1 Timestamp register	PSI5SPMB71T	<PSI5S_base> + 700 _H	32	—
	PSI5-S Receive MailBox ch7 frm2 Status register	PSI5SPMB72S	<PSI5S_base> + 704 _H	32	—

Table 25.11 List of Registers (13/13)

Module Name	Register Name	Symbol	Address	Access	Access Protection
PSI5S	PSI5-S Receive MailBox ch7 frm2 Data register	PSI5SPMB72D	<PSI5S_base> + 708 _H	32	—
	PSI5-S Receive MailBox ch7 frm2 Timestamp register	PSI5SPMB72T	<PSI5S_base> + 70C _H	32	—
	PSI5-S Receive MailBox ch7 frm3 Status register	PSI5SPMB73S	<PSI5S_base> + 710 _H	32	—
	PSI5-S Receive MailBox ch7 frm3 Data register	PSI5SPMB73D	<PSI5S_base> + 714 _H	32	—
	PSI5-S Receive MailBox ch7 frm3 Timestamp register	PSI5SPMB73T	<PSI5S_base> + 718 _H	32	—
	PSI5-S Receive MailBox ch7 frm4 Status register	PSI5SPMB74S	<PSI5S_base> + 71C _H	32	—
	PSI5-S Receive MailBox ch7 frm4 Data register	PSI5SPMB74D	<PSI5S_base> + 720 _H	32	—
	PSI5-S Receive MailBox ch7 frm4 Timestamp register	PSI5SPMB74T	<PSI5S_base> + 724 _H	32	—
	PSI5-S Receive MailBox ch7 frm5 Status register	PSI5SPMB75S	<PSI5S_base> + 728 _H	32	—
	PSI5-S Receive MailBox ch7 frm5 Data register	PSI5SPMB75D	<PSI5S_base> + 72C _H	32	—
	PSI5-S Receive MailBox ch7 frm5 Timestamp register	PSI5SPMB75T	<PSI5S_base> + 730 _H	32	—
	PSI5-S Receive MailBox ch7 frm6 Status register	PSI5SPMB76S	<PSI5S_base> + 734 _H	32	—
	PSI5-S Receive MailBox ch7 frm6 Data register	PSI5SPMB76D	<PSI5S_base> + 738 _H	32	—
	PSI5-S Receive MailBox ch7 frm6 Timestamp register	PSI5SPMB76T	<PSI5S_base> + 73C _H	32	—

25.3.2 Common Register/Config

25.3.2.1 PSI5SPUOEB — PSI5-S/UART Operation Enable Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OPEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.12 PSI5SPUOEB Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	OPEN	Operation enable 0: Disable 1: Enable The value 1 can be written to this bit when PUOS.ACSTS is 0 (= configuration mode). The value 0 can be written to this bit at any time. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.2.2 PSI5SPUOMD — PSI5-S/UART Operation Mode Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OPMD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.13 PSI5SPUOMD Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	OPMD	Operation mode select 0: UART mode 1: PSI5S mode This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). This bit is cleared when by writing 1 to PSI5SPUSWR.SWRST.

25.3.2.3 PSI5SPUOS — PSI5-S/UART Operation Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTS	ACSTS	SWSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.14 PSI5SPUOS Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved These bits are always read as 0. The write value is ignored.
2	MSTS	Mode status 0: UART mode (PSI5SPUOEB.OPEN = 1, PSI5SPUOMD.OPMD = 0) 1: PSI5S mode (PSI5SPUOEB.OPEN = 1, PSI5SPUOMD.OPMD = 1) This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
1	ACSTS	Active status 0: Not Active (Configuration mode: PSI5SPUOEB.OPEN = 0) 1: Active (UART mode or PSI5S mode: PSI5SPUOEB.OPEN = 1) This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
0	SWSTS	Software (SW) reset status 0: SW reset not asserted 1: SW reset asserted Writing 1 to PSI5SPUSWR.SWRST sets this bit to 1. This bit is read as 1 during SW reset execution. After SW reset execution, the bit is cleared to 0.

25.3.2.4 PSI5SPUNFST — PSI5-S/UART Noise Filter Set Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NFSET
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.15 PSI5SPUNFST Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	NFSET	Noise filter setting 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.2.5 PSI5SPUSWR — PSI5-S/UART Software Reset Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.16 PSI5SPUSWR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	SWRST	Software (SW) reset 0: Ignored 1: Start software reset for PSI5-S This bit is always read as 0.

25.3.2.6 PSI5SPRMBC — PSI5-S Receive MailBox Data Clear Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MBCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.17 PSI5SPRMBC Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	MBCLR	MailBox clear All MailBox data clear 0: Ignored 1: All MailBox data cleared to 0 This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode).

25.3.2.7 PSI5SPUCLB — PSI5-S/UART Communication Loop Back Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TMKV[6:0]							LBEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.18 PSI5SUCLB Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are always read as 0. The write value is ignored.
7 to 1	TMKV	Test Mode Key Values Test mode key values for loopback test function These bits are always read as 0.
0	LBEN	3rd write value of loopback test sequence enable 0: Disable 1: Enable Test function set sequence is described in Section 25.5.11.1, Loopback Test Setting Procedure . This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.2.8 PSI5SPUPTS — PSI5-S/UART Rx/Tx Parity Set Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	UTPRTY[1:0]	—	—	—	—	—	—	—	URPRTY[1:0]	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 25.19 PSI5SUPTS Register Contents

Bit Position	Bit Name	Function
31 to 10	—	Reserved These bits are always read as 0. The write value is ignored.
9 to 8	UTPRTY[1:0]	UART Tx Parity 00 _B : Parity disable 01 _B : Even parity 10 _B : 0 parity (Parity is always 0) 11 _B : Odd parity When the CPU sets parity to disable (= 00 _B), a packet (UART frame) is composed of 1 start bit, 8 data bits and 1 stop bit (10 bits in total). Otherwise, a packet (UART frame) is composed of 1 start bit, 8 data bits, 1 parity bit and 1 stop bit (11 bits in total). These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
7 to 2	—	Reserved These bits are always read as 0. The write value is ignored.
1 to 0	URPRTY[1:0]	UART Rx parity 00 _B : Parity disable 01 _B : Even parity 10 _B : Parity don't care 11 _B : Odd parity When the CPU sets parity to disable (= 00 _B), a packet (UART frame) is composed of 1 start bit, 8 data bits and 1 stop bit (10 bits in total). Otherwise, a packet (UART frame) is composed of 1 start bit, 8 data bits, 1 parity bit and 1 stop bit (11 bits in total). These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.2.9 PSI5SPUBCE — PSI5-S/UART Baud Rate Clock Enable Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCKEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.20 PSI5SUBCE Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	SCKEN	UART clock (Psis_tx_sclk) output enable 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.2.10 PSI5SPUBPR — PSI5-S/UART Baud Rate Parameter Register

Value after reset: 0004 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RXOSMP[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKDIV[7:0]							—	SCKPRS[6:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.21 PSI5SUBPR Register Contents

Bit Position	Bit Name	Function
31 to 20	—	Reserved These bits are always read as 0. The write value is ignored.
19 to 16	RXOSMP[3:0]	RX Over sample number 0 to 3: Setting prohibited 4: 5 samples ... x: x+1 samples ... 15: 16 samples These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
15 to 8	SCKDIV[7:0]	Clock divide value of UART SCLK(phis_tx_sclk) 0 = 1/1 1 = 1/2 ... x = 1/(x+1) ... 255 = 1/256 These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
7	—	Reserved This bit is always read as 0. The write value is ignored.
6 to 0	SCKPRS[6:0]	Prescaler of UART SCLK(phis_tx_sclk) 0 = 1/1 1 = 1/2 ... x = 1/(x+1) ... 127 = 1/128 These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.2.11 PSI5SPTPS — PSI5-S Timestamp Prescaler Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	—	—	—	TSPRSU[9:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	—	—	—	—	—	—	—	TSPRSL[6:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 25.22 PSI5SPTPS Register Contents

Bit Position	Bit Name	Function
31 to 26	—	Reserved These bits are always read as 0. The write value is ignored.
25 to 16	TSPRSU[9:0]	Timestamp prescaler (upper) These bits define the load data of the timestamp prescaler counter (upper 10 bits). These bits configure a max 1 ms enable pulse from a 1 μs enable pulse The timestamp prescaler counter (upper) loads this setting and starts counting down until it reaches 1. Loading and counting down count are continuously repeated. When the timestamp prescaler counter (upper) becomes 1, the timestamp tick is output. If the CPU sets 0 to these bits, this pulse is not output. These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
15 to 7	—	Reserved These bits are always read as 0. The write value is ignored.
6 to 0	TSPRSL[6:0]	Timestamp prescaler (lower) These bits define the load data of the timestamp prescaler counter (lower 7 bits). The timestamp prescaler counter (lower) loads this setting and starts counting down until it reaches 1. Loading and counting down count are continuously repeated. These bits configure a 1 μs clock from PCLK. When the timestamp prescaler counter (lower) becomes 1, a 1 μs enable pulse is output. When the CPU sets 0 to these bits, the 1 μs enable pulse is not output. These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.2.12 PSI5SPTCAS — PSI5-S Timestamp Counter A Select Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCAC LS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TSCAE BS	—	—	—	—	—	—	—	TSCAC KS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 25.23 PSI5SPTCAS Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved These bits are always read as 0. The write value is ignored.
16	TSCACLS	Timestamp counter A clear select 0: Signal generated by PSI5-S is selected 1: GTM output is selected This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
15 to 9	—	Reserved These bits are always read as 0. The write value is ignored.
8	TSCAEBS	Timestamp counter A enable select 0: Signal generated by PSI5-S is selected (= PTCAE.TSCAEB PATCE.ATSCEB). 1: GTM output is selected This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
7 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	TSCACKS	Timestamp counter A clock select 0: Signal generated by PSI5-S is selected 1: GTM output is selected This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.2.13 PSI5SPTCBS — PSI5-S Timestamp Counter B Select Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCBCLS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TSCBEBS	—	—	—	—	—	—	—	TSCBCKS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 25.24 PSI5SPTCBS Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved These bits are always read as 0. The write value is ignored.
16	TSCBCLS	Timestamp counter B clear select 0: Signal generated by PSI5-S is selected 1: GTM output is selected This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
15 to 9	—	Reserved These bits are always read as 0. The write value is ignored.
8	TSCBEBS	Timestamp counter B enable select 0: Signal generated by PSI5-S is selected 1: GTM output is selected This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
7 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	TSCBCKS	Timestamp counter B clock select 0: Signal generated by PSI5-S is selected 1: GTM output is selected This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.2.14 PSI5SPTCAE — PSI5-S Timestamp Counter A Enable Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCAEB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.25 PSI5SPTCAE Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	TSCAEB	Timestamp counter A enable 0: Disable 1: Enable When PSI5SPTCAS.TSCAEB is 0, this bit is selected and is used for timestamp counterA. When PSI5SPTCAS.TSCAEB is 1, this bit is disabled. This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.2.15 PSI5SPTCAC — PSI5-S Timestamp Counter A Clear Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCAC LR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.26 PSI5SPTCAC Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	TSCACLR	Timestamp counter A clear 0: Ignored 1: Clear timestamp counterA When PSI5SPTCAS.TSCACLS is 0, this bit is enabled and uses timestamp counter A clear. When PSI5SPTCAS.TSCACLS is 1, this bit is disabled. This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).

25.3.2.16 PSI5SPTCBE — PSI5-S Timestamp Counter B Enable Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCBE B
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.27 PSI5SPTCBE Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	TSCBEB	Timestamp counter B enable 0: Disable 1: Enable When PSI5SPTCBS.TSCBEB is 0, this bit is enabled and is used for timestamp counter B. When PSI5SPTCBS.TSCBEB is 1, this bit is disabled. This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.2.17 PSI5SPTCBC — PSI5-S Timestamp Counter B Clear Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCBCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.28 PSI5SPTCBC Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	TSCBCLR	Timestamp counter B clear 0: Ignored 1: Clears timestamp counter B When PSI5SPTCBS.TSCBCLS is 0, this bit is enabled and uses timestamp counter B clear. When PSI5SPTCBS.TSCBCLS is 1, this bit is not used. This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).

25.3.2.18 PSI5SPATCE — PSI5-S All Timestamp Counter Enable Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATSCE B
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.29 PSI5SPATCE Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	ATSCEB	All timestamp counter enable 0: Ignored Timestamp counter A is enabled according to the setting of PSI5SPTCAE.TSCAEB Timestamp counter B is enabled according to the setting of PSI5SPTCBE.TSCBEB 1: All timestamp counters are enabled regardless of the settings of PSI5SPTCAE.TSCAEB / PSI5SPTCBE.TSCBEB. This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.2.19 PSI5SPATCC — PSI5-S All Timestamp Counter Clear Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATSCCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.30 PSI5SPATCC Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	ATSCCLR	All Timestamp Counters Clear 0: Ignored 1: Clears all Timestamp counter When PSI5SPTCAS.TSCACLS is 0, this bit is enabled clearing timestamp counter A. When PSI5SPTCAS.TSCACLS is 1, this bit is disabled. When PSI5SPTCBS.TSCBCLS is 0, this bit is enabled clearing timestamp counter B. When PSI5SPTCBS.TSCBCLS is 1, this bit is disabled. This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).

25.3.2.20 PSI5SUCRIE — UART Communication Rx Interrupt Enable

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	IERFIN	IEROE	IERFE	IERPE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 25.31 PSI5SUCRIE Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved These bits are always read as 0. The write value is ignored.
3	IERFIN	Interrupt enable of UART Rx finish flag 0: Disable 1: Enable This bit controls interrupt by PSI5SUCRS.UTRFIN. This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 0 (= UART mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
2	IEROE	Interrupt enable of UART Rx overrun error flag 0: Disable 1: Enable This bit controls interrupt by PSI5SUCRS.UTRFOE. This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 0 (= UART mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
1	IERFE	Interrupt enable of UART Rx framing error flag 0: Disable 1: Enable This bit controls interrupt by PSI5SUCRS.UTRFE. This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 0 (= UART mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
0	IERPE	Interrupt enable of UART Rx parity error flag 0: Disable 1: Enable This bit controls interrupt by PSI5SUCRS.UTRPE. This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 0 (= UART mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.2.21 PSI5SUCTIE — UART Communication Tx Interrupt Enable

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IETFIN	IETOWE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 25.32 PSI5SUCTIE Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved These bits are always read as 0. The write value is ignored.
1	IETFIN	Interrupt enable of UART Tx finish flag 0: Disable 1: Enable This bit controls interrupt by PSI5SUCRS.UTTFIN. This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 0 (= UART mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
0	IETOWE	Interrupt enable of UART Tx overwrite error flag 0: Disable 1: Enable This bit controls interrupt by PSI5SUCRS.UTTFOWE. This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 0 (= UART mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.2.22 PSI5UCDRE — UART Communication DMA Request Enable Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRQEU TFN	DRQEU RFN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 25.33 PSI5UCDRE Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved These bits are always read as 0. The write value is ignored.
1	DRQEUTFN	DMA request enable at UART Tx finish 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 0 (= UART mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
0	DRQEURFN	DMA request enable at UART Rx finish 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 0 (= UART mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.3 Common Register/Rx

25.3.3.1 PSI5SUCRD — UART Communication Rx Data Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	UTRDT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.34 PSI5SUCRD Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are always read as 0. The write value is ignored.
7 to 0	UTRDT[7:0]	UART read data When the stop bit of the Rx UART frame is detected in UART mode, PSI5SUCRD.UTRDT is stored. If the next UART frame finishes before the CPU has read this address, PSI5SUCRS.UTROE is set to 1 and this register is overwritten with new data. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.3.2 PSI5UCRS — UART Communication Rx Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	UTRFIN	UTROE	UTRFE	UTRPE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.35 PSI5UCRS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 4	—	Reserved These bits are always read as 0. The write value is ignored.
3	UTRFIN	<p>UART Rx finish flag</p> <p>0: A frame is not received successfully 1: A frame is received successfully</p> <p>When the stop bit of the UART frame is detected in UART mode without any error occurring, this bit is set to 1 and an interrupt (int_psis_ch0) and DMA request (dma_psis_ch7_rx) occur. This bit is cleared by writing 1 to PSI5UCRSC.UTRFINCL.</p> <p>When PSI5UCRS.UTRFIN is set and 1 is written to PSI5UCRSC.UTRFINCL in the same clock cycle, PSI5UCRS.UTRFIN remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
2	UTROE	<p>UART Rx overrun error flag</p> <p>0: No error 1: An overrun error detected</p> <p>If the stop bit of the next UART frame is detected in UART mode before the CPU has read PSI5UCRD.UTRDT, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. This bit is cleared by writing 1 to PSI5UCRSC.UTROECL.</p> <p>When PSI5UCRS.UTROE is set and 1 is written to PSI5UCRSC.UTROECL in the same clock cycle, PSI5UCRSC.UTROE remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
1	UTRFE	<p>UART Rx framing error flag</p> <p>0: No error 1: A framing error detected</p> <p>If the stop bit of the UART frame is detected in UART mode when the bit detection value is 0, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. If an Rx overrun error occurs during stop bit detection, this bit remains 0. This bit is cleared by writing 1 to PSI5UCRSC.UTRFECL.</p> <p>When PSI5UCRS.UTRFE is set and 1 is written to PSI5UCRSC.UTRFECL in the same clock cycle, PSI5UCRSC.UTRFE remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>

Table 25.35 PSI5SUCRS Register Contents (2/2)

Bit Position	Bit Name	Function
0	UTRPE	<p>UART Rx parity error</p> <p>0: No error</p> <p>1: A parity error detected</p> <p>If the stop bit of the UART frame is detected in UART mode when a parity error has been detected, this bit is set to 1 and an interrupt (int_psis_ch0) occurs.</p> <p>If a framing error or Rx overrun error occurs during stop bit detection, this bit remains 0.</p> <p>This bit is cleared by writing 1 to PSI5SUCRSC.UTRPECL.</p> <p>When PSI5SUCRS.UTRPE is set and 1 is written to PSI5SUCRSC.UTRPECL in the same clock cycle, PSI5SUCRSC.UTRPE remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>

25.3.3.3 PSI5UCRSC — UART Communication Rx Status Clear Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	UTRFIN CL	UTROE CL	UTRFE CL	UTRPE CL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 25.36 PSI5UCRSC Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved These bits are always read as 0. The write value is ignored.
3	UTRFINCL	UART Rx finish flag clear 0: Ignored 1: UART Rx finish flag (PSI5UCRS.UTRFIN) clear This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 0 (= UART mode).
2	UTROECL	UART Rx overrun error clear 0: Ignored 1: Overrun error (PSI5UCRS.UTROE) clear This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 0 (= UART mode).
1	UTRFECL	UART Rx framing error clear 0: Ignored 1: Framing error (PSI5UCRS.UTRFE) clear This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 0 (= UART mode).
0	UTRPECL	UART Rx parity error clear 0: Ignored 1: Parity error (PSI5UCRS.UTRPE) clear This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 0 (= UART mode).

25.3.4 Common Register/Tx

25.3.4.1 PSI5SPTFST — PSI5-S Tx Frame Start Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.37 PSI5SPTFST Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	TXST	Tx start PSI5-S command data transmission start 0: Ignored 1: Transmission start When the CPU writes 1 to this bit, 1-8 command data (PSI5SPTFD1.TDT1 to PTFD2.TDT8) transmission starts. This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).

25.3.4.2 PSI5SPTFNM — PSI5-S Tx Frame Number Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TXNUM[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 25.38 PSI5SPTFNM Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved These bits are always read as 0. The write value is ignored.
2 to 0	TXNUM[2:0]	Tx command data number 000 _B : 1 packet (UART frame) send (PSI5SPTFD1.TDT1) 001 _B : 2 packet (UART frame) send (PSI5SPTFD1.TDT1 to PSI5SPTFD1.TDT2) 010 _B : 3 packet (UART frame) send (PSI5SPTFD1.TDT1 to PSI5SPTFD1.TDT3) 011 _B : 4 packet (UART frame) send (PSI5SPTFD1.TDT1 to PSI5SPTFD1.TDT4) 100 _B : 5 packet (UART frame) send (PSI5SPTFD1.TDT1 to PSI5SPTFD2.TDT5) 101 _B : 6 packet (UART frame) send (PSI5SPTFD1.TDT1 to PSI5SPTFD2.TDT6) 110 _B : 7 packet (UART frame) send (PSI5SPTFD1.TDT1 to PSI5SPTFD2.TDT7) 111 _B : 8 packet (UART frame) send (PSI5SPTFD1.TDT1 to PSI5SPTFD2.TDT8) These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.4.3 PSI5SPTFD1 — PSI5-S Tx Frame Data1 Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TDT4[7:0]								TDT3[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDT2[7:0]								TDT1[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.39 PSI5SPTFD1 Register Contents

Bit Position	Bit Name	Function
31 to 24	TDT4[7:0]	<p>These bits define the transmission data of the 4th packet (UART frame). When PSI5SPTFNM.TXNUM is less than 3, this setting is ignored. These bits cannot be written when PSI5SPTFS.TXSTS is 1. These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
23 to 16	TDT3[7:0]	<p>These bits define the transmission data of the 3rd packet (UART frame). When PSI5SPTFNM.TXNUM is less than 2, this setting is ignored. These bits cannot be written when PSI5SPTFS.TXSTS is 1. These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
15 to 8	TDT2[7:0]	<p>These bits define the transmission data of the 2nd packet (UART frame). When PSI5SPTFNM.TXNUM is less than 1, this setting is ignored. These bits cannot be written when PSI5SPTFS.TXSTS is 1. These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
7 to 0	TDT1[7:0]	<p>These bits define the transmission data of the 1st packet (UART frame). These bits cannot be written when PSI5SPTFS.TXSTS is 1. These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>

25.3.4.4 PSI5SPTFD2 — PSI5-S Tx Frame Data2 Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TDT8[7:0]								TDT7[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDT6[7:0]								TDT5[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.40 PSI5SPTFD2 Register Contents

Bit Position	Bit Name	Function
31 to 24	TDT8[7:0]	<p>These bits define the transmission data of the 8th packet.</p> <p>When PSI5SPTFNM.TXNUM is less than 7, these setting are ignored.</p> <p>These bits cannot be written when PSI5SPTFS.TXSTS is 1.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p> <p>These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
23 to 16	TDT7[7:0]	<p>These bits define the transmission data of the 7th packet.</p> <p>When PSI5SPTFNM.TXNUM is less than 6, these setting are ignored.</p> <p>These bits cannot be written when PSI5SPTFS.TXSTS is 1.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p> <p>These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
15 to 8	TDT6[7:0]	<p>These bits define the transmission data of the 6th packet.</p> <p>When PSI5SPTFNM.TXNUM is less than 5, these setting are ignored.</p> <p>These bits cannot be written when PSI5SPTFS.TXSTS is 1.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p> <p>These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
7 to 0	TDT5[7:0]	<p>These bits define the transmission data of the 5th packet.</p> <p>When PSI5SPTFNM.TXNUM is less than 4, these setting are ignored.</p> <p>These bits cannot be written when PSI5SPTFS.TXSTS is 1.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p> <p>These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>

25.3.4.5 PSI5SPTFS — PSI5-S Tx Frame Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.41 PSI5SPTFS Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	TXSTS	<p>Tx status</p> <p>0: Transmission not busy</p> <p>1: Transmission busy</p> <p>When the CPU sets 1 to PSI5SPTFST.TXST, this bit is set to 1.</p> <p>When all transmission data (PSI5SPTFD1, PSI5SPTFD2) has been sent to the Tx shifter, this bit is reset to 0.</p> <p>When this bit is 1, PSI5SPTFD1 and PSI5SPTFD2 cannot be written.</p> <p>This bit is read only. The write value is ignored.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>

25.3.4.6 PSI5SPTFIS — PSI5-S Tx FIFO Status Register

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXFFFL	TXFFEP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.42 PSI5SPTFIS Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved These bits are always read as 0. The write value is ignored
1	TXFFFL	Tx FIFO full 0: Not full 1: Full This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).
0	TXFFEP	Tx FIFO empty 0: Not empty 1: Empty This bit is set to 1 by writing 1 to PSI5SPUSWR.SWRST. This bit is set to 1 when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).

25.3.4.7 PSI5SUCTD — UART Communication Tx Data Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	UTTDT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.43 PSI5SUCTD Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are always read as 0. The write value is ignored.
7 to 0	UTTDT[7:0]	UART transmission data These bits cannot be written when PSI5SUCTM.UTTBBF is 1. When the CPU writes 1 to PSI5SUCTD.UTTDT in PSI5SUCTM.UTTBBF, PSI5SUCTS.UTTOWE is set to 1. These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 0 (= UART mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.4.8 PSI5SUCTM — UART Communication Tx Monitoring Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UTTF	UTTBB F
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.44 PSI5SUCTM Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved These bits are always read as 0. The write value is ignored.
1	UTTF	UART transmission flag 0: Not transmitting 1: Transmitting When stop bit output ends without any subsequent write data, this bit is reset to 0. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).
0	UTTBBF	UART Tx shifter busy flag 0: Permitted to write to PSI5SUCTD.UTTDT 1: Prohibited to write to PSI5SUCTD.UTTDT This bit shows status of Tx shifter busy in UART mode. When CPU writes PSI5SUCTD.UTTDT, this bit is set to 1. When PSI5SUCTD.UTTDT data is stored in the Tx shifter, this bit is reset to 0. When this bit is 1, PSI5SUCTD.UTTDT cannot be written. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).

25.3.4.9 PSI5SUCTS — UART Communication Tx Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UTTFIN	UTTOWE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.45 PSI5SUCTS Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved These bits are always read as 0. The write value is ignored.
1	UTTFIN	UART transmission finish 0: Transmission not finished 1: A UART frame transmission has finished When PSI5SUCTD.UTTDT data is sent to the Tx shifter, this bit is set to 1 and an interrupt (int_psis_ch1) and DMA request (dma_psis_ch7_tx) occur. This bit is cleared by writing 1 to PSI5SUCTSC.UTTFINCL. When PSI5SUCTS.UTTFIN is set and 1 is written to PSI5SUCTSC.UTTFINCL in the same clock cycle, PSI5SUCTS.UTTFIN remains 1. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).
0	UTTOWE	UART transmission overwrite error 0: No error 1: Overwrite error When the CPU writes 1 to PSI5SUCTD.UTTDT in PSI5SUCTM.UTTBBF, this bit sets to 1 and an interrupt (int_psis_ch1) occurs. This bit is cleared by writing 1 to PSI5SUCTSC.UTTOWECL. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).

25.3.4.10 PSI5SUCTSC — UART Communication Tx Status Clear Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UTTFIN CL	UTTOW ECL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 25.46 PSI5SUCTSC Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved These bits are always read as 0. The write value is ignored.
1	UTTFINCL	UART Tx finish flag clear 0: Ignored 1: UART Tx finish flag (PSI5SUCTS.UTTFIN) clear This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 0 (= UART mode).
0	UTTOWECL	UART Tx overwrite error clear 0: Ignored 1: Overwrite error (PSI5SUCTS.UTTOWE) clear This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 0 (= UART mode).

25.3.5 Ch0 Register/Config

25.3.5.1 PSI5SPRCF10 — PSI5-S Receive Config1 Ch0 Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PFRMIDLE[3:0]				F6PKT[2:0]			F5PKT[2:0]			F4PKT[2:1]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	F4PKT[0]	F3PKT[2:0]			F2PKT[2:0]			F1PKT[2:0]			—*1	—*2	TSCS	TSEN	RFCPS	CHEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Table 25.47 PSI5SPRCF10 Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 28	—	Reserved These bits are always read as 0. The write value is ignored.
27 to 24	PFRMIDLE[3:0]	Packet frame idle Minimum packet frame gap (all channel) 0: The next packet frame is detected after 1 gap. 1: The next packet frame is detected after 2 gaps. ... n: The next packet frame is detected after n+1 gaps. ... 15: The next packet frame is detected after 16 gaps. These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
23 to 21	F6PKT[2:0]	Frame 6 packet number (Ch0) 000 _B : Frame 6 data is ignored 011 _B : Packet number is set to 3 100 _B : Packet number is set to 4 101 _B : Packet number is set to 5 110 _B : Packet number is set to 6 Other than above: Setting prohibited “Frame 6” is a packet frame whose FID is 101 _B . These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
20 to 18	F5PKT[2:0]	Frame 5 packet number (Ch0) 000 _B : Frame 5 data is ignored 011 _B : Packet number is to set 3 100 _B : Packet number is to set 4 101 _B : Packet number is to set 5 110 _B : Packet number is to set 6 Other than above: Setting prohibited “Frame 5” is a packet frame whose FID is 100 _B . These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

Table 25.47 PSI5SPRCF10 Register Contents (2/3)

Bit Position	Bit Name	Function
17 to 15	F4PKT[2:0]	<p>Frame 4 packet number (Ch0)</p> <p>000_B: Frame 4 data is ignored</p> <p>011_B: Packet number is to set 3</p> <p>100_B: Packet number is to set 4</p> <p>101_B: Packet number is to set 5</p> <p>110_B: Packet number is to set 6</p> <p>Other than above: Setting prohibited "Frame 4" is a packet frame whose FID is 011_B.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode).</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
14 to 12	F3PKT[2:0]	<p>Frame 3 packet number (Ch0)</p> <p>000_B: Frame 3 data is ignored</p> <p>011_B: Packet number is to set 3</p> <p>100_B: Packet number is to set 4</p> <p>101_B: Packet number is to set 5</p> <p>110_B: Packet number is to set 6</p> <p>Other than above: Setting prohibited</p> <p>"Frame 3" is a packet frame whose FID is 010_B.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode).</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
11 to 9	F2PKT[2:0]	<p>Frame 2 packet number (Ch0)</p> <p>000_B: Frame 2 data is ignored</p> <p>011_B: Packet number is to set 3</p> <p>100_B: Packet number is to set 4</p> <p>101_B: Packet number is to set 5</p> <p>110_B: Packet number is to set 6</p> <p>Other than above: Setting prohibited</p> <p>"Frame 2" is a packet frame whose FID is 001_B.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode).</p> <p>These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
8 to 6	F1PKT[2:0]	<p>Frame 1 packet number (Ch0)</p> <p>000_B: Frame 2 data is ignored</p> <p>011_B: Packet number is to set 3</p> <p>100_B: Packet number is to set 4</p> <p>101_B: Packet number is to set 5</p> <p>110_B: Packet number is to set 6</p> <p>Other than above: Setting prohibited</p> <p>"Frame 1" is a packet frame whose FID is 000_B.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode).</p> <p>These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
5 to 4	Reserved	<p>When read, the value after reset is returned. When writing, write the value after reset.</p> <p>*1</p> <p>*2</p>
3	TSCS	<p>Timestamp counter select (Ch0)</p> <p>0: Select timestamp counter B</p> <p>1: Select timestamp counter A</p> <p>This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode).</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
2	TSEN	<p>Timestamp capture enable (Ch0)</p> <p>0: Disable</p> <p>1: Enable</p> <p>This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode).</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p>

Table 25.47 PSI5SPRCF10 Register Contents (3/3)

Bit Position	Bit Name	Function
1	RFCPS	Rx frame checksum CRC/parity select (Ch0) 0: Parity select 1: CRC select This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
0	CHEN	Channel enable (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

Note 1. bit[5]: Channel 0 does not have SYSEL, because channel 0 does not transmit.

Note 2. bit[4]: Channel 0 does not have TSCTS (Timestamp capture trigger select), because channel 0 has only 1 kind of capture trigger.

25.3.5.2 PSI5SPRCF20 — PSI5-S Receive Config2 Ch0 Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	F6PAYLD[4:0]				F5PAYLD[4:0]				F4PAYLD[4:1]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	F4PAYLD[0]	F3PAYLD[4:0]				F2PAYLD[4:0]				F1PAYLD[4:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.48 PSI5SPRCF20 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 30	—	Reserved These bits are always read as 0. The write value is ignored.
29 to 25	F6PAYLD[4:0]	The payload bit length of packet frame 6 (Ch0) 8: Sets 8 ... x: Sets x ... 28: Sets 28 Other than above: Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope. These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
24 to 20	F5PAYLD[4:0]	The payload bit length of packet frame 5 (Ch0) 8: Sets 8 ... x: Sets x ... 28: Sets 28 Other than above: Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope. These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

Table 25.48 PSI5SPRCF20 Register Contents (2/2)

Bit Position	Bit Name	Function
19 to 15	F4PAYLD[4:0]	<p>The payload bit length of packet frame 4 (Ch0)</p> <p>8: Sets 8</p> <p>...</p> <p>x: Sets x</p> <p>...</p> <p>28: Sets 28</p> <p>Other than above: Setting prohibited</p> <p>If the initial value (0) is set to these bits, it is handled as 8.</p> <p>These bits are used to determine the CRC/parity calculation scope.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode).</p> <p>These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
14 to 10	F3PAYLD[4:0]	<p>The payload bit length of packet frame 3 (Ch0)</p> <p>8: Sets 8</p> <p>...</p> <p>x: Sets x</p> <p>...</p> <p>28: Sets 28</p> <p>Other than above: Setting prohibited</p> <p>If the initial value (0) is set to these bits, it is handled as 8.</p> <p>These bits are used to determine the CRC/parity calculation scope.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode).</p> <p>These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
9 to 5	F2PAYLD[4:0]	<p>The payload bit length of packet frame 2 (Ch0)</p> <p>8: Sets 8</p> <p>...</p> <p>x: Sets x</p> <p>...</p> <p>28: Sets 28</p> <p>Other than above: Setting prohibited</p> <p>If the initial value (0) is set to these bits, it is handled as 8.</p> <p>These bits are used to determine the CRC/parity calculation scope.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode).</p> <p>These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
4 to 0	F1PAYLD[4:0]	<p>The payload bit length of packet frame 1 (Ch0)</p> <p>8: Sets 8</p> <p>...</p> <p>x: Sets x</p> <p>...</p> <p>28: Sets 28</p> <p>Other than above: Setting prohibited</p> <p>If the initial value (0) is set to these bits, it is handled as 8.</p> <p>These bits are used to determine the CRC/parity calculation scope.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode).</p> <p>These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>

25.3.5.3 PSI5SPWDE0 — PSI5-S WDT Enable Ch0 Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTEB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.49 PSI5SPWDE0 Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	WDTEB	Watchdog timer of Rx frame enable (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing to PSI5SPUSWR.SWRST.

25.3.5.4 PSI5SPWDP0 — PSI5-S WDT Prescaler Ch0 Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	WDTPRS[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.50 PSI5SPWDP0 Register Contents

Bit Position	Bit Name	Function
31 to 12	—	Reserved These bits are always read as 0. The write value is ignored.
11 to 0	WDTPRS[11:0]	Watchdog timer prescaler (Ch0) 0: Stop watchdog timer 1 to 4095: Enabled at 1 clock/x clock (x:1 to 4095) These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.5.5 PSI5SPWDEV0 — PSI5-S WDT Expiration Value Ch0 Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	WDTEX[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTEX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.51 PSI5SPWDEV0 Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Reserved These bits are always read as 0. The write value is ignored.
23 to 0	WDTEX[23:0]	Watchdog timer expiration value (Ch0) These bits define the expiration value of the watchdog timer for Rx packet frame monitor in Ch0. When the watchdog counter counts down to 0 from this setting value, the watchdog timer is judged to have expired. These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.5.6 PSI5SPCIE0 — PSI5-S CPU Interrupt Enable Ch0 Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	IEBCTFN	—	—	IEBRFN	IEBRFEX	IEBRFLK	IEBROV	IEBRWDT	—	IEBUTFR	IEBUTPT	IEBTRST	IEBPT	IEBCRC	IEBXCR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.52 PSI5SPCIE0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 15	—	Reserved These bits are always read as 0. The write value is ignored.
14	IEBCTFN	Interrupt enable of command Tx finish (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
13 to 12	—	Reserved These bits are always read as 0. The write value is ignored.
11	IEBRFN	Interrupt enable of Rx packet frame finish flag (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
10	IEBRFEX	Interrupt enable of Rx frame excess error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
9	IEBRFLK	Interrupt enable of Rx frame lack error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
8	IEBROV	Interrupt enable of Rx overrun error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

Table 25.52 PSI5SPCIE0 Register Contents (2/2)

Bit Position	Bit Name	Function
7	IEBRWDT	<p>Interrupt enable of Rx WDT error (Ch0)</p> <p>0: Disable 1: Enable</p> <p>This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
6	—	<p>Reserved</p> <p>This bit is always read as 0. The write value is ignored.</p>
5	IEBUTFR	<p>Interrupt enable of Rx UART framing error (Ch0)</p> <p>0: Disable 1: Enable</p> <p>This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
4	IEBUTPT	<p>Interrupt enable of Rx UART parity error (Ch0)</p> <p>0: Disable 1: Enable</p> <p>This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
3	IEBTRST	<p>Interrupt enable of Rx transceiver status error (Ch0)</p> <p>0: Disable 1: Enable</p> <p>This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
2	IEBPT	<p>Interrupt enable of Rx payload data parity error (Ch0)</p> <p>0: Disable 1: Enable</p> <p>This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
1	IEBCRC	<p>Interrupt enable of Rx payload data CRC error (Ch0)</p> <p>0: Disable 1: Enable</p> <p>This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
0	IEBXCRC	<p>Interrupt enable of packet frame XCRC error (Ch0)</p> <p>0: Disable 1: Enable</p> <p>This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p>

25.3.5.7 PSI5SPDRE0 — PSI5-S DMA Transfer Request Enable Ch0 Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRQE WDT	DRQER FN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 25.53 PSI5SPDRE0 Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved These bits are always read as 0. The write value is ignored.
1	DRQEWDT	Enable of DMA request by the WDT error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
0	DRQERFN	Enable of DMA request by the channel data Rx finish (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.6 Ch0 Register/Rx

25.3.6.1 PSI5SPRES0 — PSI5-S Receive Error Status Ch0 Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—*1	—*1	—*1	—*1	RERRF2	RERRF1
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.54 PSI5SPRES0 Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved These bits are always read as 0. The write value is ignored.
1	RERRF2	Rx error at packet frame 2 (Ch0) 0: No error 1: An error has occurred This bit is set to 1 when any of “Rx overrun error”, “Rx WDT error”, “UART framing error”, “UART parity error”, “transceiver status error”, “payload data parity error”, “payload data CRC error” or “packet frame XCRC error” in packet frame2 of Ch0 has occurred. This bit is cleared when writing 0b1 to PSI5SPRESC0.RERRCLF2. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).
0	RERRF1	Rx error at packet frame 1 (Ch0) 0: No error 1: An error has occurred This bit is set to 1 when any of “Rx overrun error”, “transceiver status error”, “payload data parity error”, or “payload data CRC error” in packet frame1 of Ch0 has occurred. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).

Note 1. Ch0 only has Frame 1 and frame2. (Other channels have frame 1 to frame 6.)

25.3.6.2 PSI5SPRESC0 — PSI5-S Receive Error Status Clear Ch0 Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—*1	—*1	—*1	—*1	RERRCLF2	RERRCLF1
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 25.55 PSI5SPRESC0 Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved These bits are always read as 0. The write value is ignored.
1	RERRCLF2	Rx error clear for packet frame 2 (Ch0) 0: Ignored 1: Clears PSI5SPRES0.RERRF2 This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).
0	RERRCLF1	Rx error clear for packet frame1(Ch0) 0: Ignored 1: Clears PSI5SPRES0.RERRF1 This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).

Note 1. Ch0 only has packet frame 1 and packet frame2. (Other channels have frame1 to frame 6.)

25.3.6.3 PSI5SPTCDT0 — PSI5-S Timestamp Capture Data Ch0 Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TSCD[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSCD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.56 PSI5SPTCDT0 Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Reserved These bits are always read as 0. The write value is ignored.
23 to 0	TSCD[23:0]	Timestamp capture data (Ch0) These bits are cleared by writing 1 to PSI5SPTCDC0.TSCCLR. These bits are cleared by writing 0 to PSI5SPRCF10.TSEN. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.6.4 PSI5SPTCDC0 — PSI5-S Timestamp Capture Data Clear Ch0 Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.57 PSI5SPTCDC0 Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	TSCCLR	Timestamp capture clear(Ch0) 0: Ignored 1: Clears timestamp capture This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).

25.3.7 Ch0 Register/Interrupt

25.3.7.1 PSI5SPCIS0 — PSI5-S CPU Interrupt Status Ch0 Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ISTCTFN	—	—	ISTRFN	ISTRFEX	ISTRFLK	ISTROV	ISTRWDT	—	ISTUTFR	ISTUTPT	ISTTRS	ISTPT	ISTCRC	ISTXCR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.58 PSI5SPCIS0 Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 15	—	Reserved These bits are always read as 0. The write value is ignored.
14	ISTCTFN	Interrupt status of command Tx finish 0: PSI5-S command transmission has not finished 1: PSI5-S command transmission has finished When the last command sent to the Tx shifter, a PSI5-S frame is stored in a MB and the frame has no errors, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. This bit is cleared by writing 1 to PSI5SPCISC0.ISTCTFN. When PSI5SPCISC0.ISTCTFN is set and 1 is written to PSI5SPCISC0.ISTCTFN in the same clock cycle, PSI5SPCISC0.ISTCTFN remains 1. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).
13 to 12	Reserved	When read, the value after reset is returned.
11	ISTRFN	Interrupt status of Rx finish (Ch0) 0: PSI5-S frame is not received successfully 1: PSI5-S frame is received successfully When a PSI5-S frame is stored in a MB and the PSI5-S frame has no errors, this bit is set to 1 and an interrupt (int_psis_ch0) and DMA request (dma_psis_ch0_rx) occur. This bit is cleared by writing 1 to PSI5SPCISC0.ISTRFN. When PSI5SPCISC0.ISTRFN is set and 1 is written to PSI5SPCISC0.ISTRFN in the same clock cycle, PSI5SPCISC0.ISTRFN remains 1. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).
10	ISTRFEX	Interrupt status of Rx frame excess error (Ch0) 0: No error 1: Error detected When a packet over PSI5SPRCF10.FmPKT(m = 1 to 6) is received, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. This bit is cleared by writing 1 to PSI5SPCISC0.ISTRFEX. When PSI5SPCISC0.ISTRFEX is set and 1 is written to PSI5SPCISC0.ISTRFEX in the same clock cycle, PSI5SPCISC0.ISTRFEX remains 1. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).

Table 25.58 PSI5SPCIS0 Register Contents (2/3)

Bit Position	Bit Name	Function
9	ISTRFLK	<p>Interrupt status of Rx frame lack error (Ch0)</p> <p>0: No error 1: Error detected</p> <p>When a packet frame gap is detected while the packet is under PSI5SPRCF10.FmPKT (m=1 to 6), this bit is set to 1 and an interrupt (int_psis_ch0) occurs.</p> <p>This bit is cleared by writing 1 to PSI5SPCISC0.ISTCRFLK.</p> <p>When PSI5SPCIS0.ISTRFLK is set and 1 is written to PSI5SPCISC0.ISTCRFLK in the same clock cycle, PSI5SPCIS0.ISTRFLK remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
8	ISTROV	<p>Interrupt status of Rx overrun error (Ch0)</p> <p>0: No error 1: Error detected</p> <p>When an attempt is made to store the next PSI5-S frame in a MB before the CPU has read that MB, this bit is set to 1 and an interrupt (int_psis_ch0) occurs.</p> <p>This bit is cleared by writing 1 to PSI5SPCISC0.ISTCROV.</p> <p>When PSI5SPCISC0.ISTROV is set and 1 is written to PSI5SPCISC0.ISTCROV in the same clock cycle, PSI5SPCISC0.ISTROV remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
7	ISTRWDT	<p>Interrupt status of Rx WDT error (Ch0)</p> <p>0: No error 1: Error detected</p> <p>When a WDT error occurs in PSI5S mode, this bit is set to 1 and an interrupt (int_psis_ch0) and DMA request (dma_psis_ch0_rx) occur.</p> <p>This bit is cleared by writing 1 to PSI5SPCISC0.ISTCRWDT.</p> <p>When PSI5SPCIS0.ISTRWDT is set and 1 is written to PSI5SPCISC0.ISTCRWDT in the same clock cycle, PSI5SPCIS0.ISTRWDT remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
6	—	<p>Reserved</p> <p>This bit is always read as 0. The write value is ignored.</p>
5	ISTUTFR	<p>Interrupt status of Rx UART framing error (Ch0)</p> <p>0: No error 1: Error detected</p> <p>If the stop bit of the UART frame is detected in PSI5S mode when the bit detection value is 0, this bit is set to 1 and an interrupt (int_psis_ch0) occurs.</p> <p>This bit is cleared by writing 1 to PSI5SPCISC0.ISTCUTFR.</p> <p>When PSI5SPCIS0.ISTUTFR is set and 1 is written to PSI5SPCISC0.ISTCUTFR in the same clock cycle, PSI5SPCIS0.ISTUTFR remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
4	ISTUTPT	<p>Interrupt status of Rx UART parity error (Ch0)</p> <p>0: No error 1: Error detected</p> <p>If the stop bit of the UART frame is detected in PSI5S mode when a UART parity error has occurred, this bit is set to 1 and an interrupt (int_psis_ch0) occurs.</p> <p>This bit is cleared by writing 1 to PSI5SPCISC0.ISTCUTPT.</p> <p>When PSI5SPCIS0.ISTUTPT is set and 1 is written to PSI5SPCISC0.ISTCUTPT in the same clock cycle, PSI5SPCIS0.ISTUTPT remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>

Table 25.58 PSI5SPCIS0 Register Contents (3/3)

Bit Position	Bit Name	Function
3	ISTTRST	<p>Interrupt status of Rx transceiver status error (Ch0)</p> <p>0: No error 1: Error detected</p> <p>When a PSI5-S frame is stored in a MB and a transceiver status error occurs, this bit is set to 1 and an interrupt (int_psis_ch0) occurs.</p> <p>This bit is cleared by writing 1 to PSI5SPCISC0.ISTCTRST.</p> <p>When PSI5SPCIS0.ISTTRST is set and 1 is written to PSI5SPCISC0.ISTCTRST in the same clock cycle, PSI5SPCIS0.ISTTRST remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
2	ISTPT	<p>Interrupt status of payload data parity error (Ch0)</p> <p>0: No error 1: Error detected</p> <p>When a PSI5-S frame is stored in a MB and a payload data parity error occurs, this bit is set to 1 and an interrupt (int_psis_ch0) occurs.</p> <p>This bit is cleared by writing 1 to PSI5SPCISC0.ISTCPT.</p> <p>When PSI5SPCIS0.ISTPT is set and 1 is written to PSI5SPCISC0.ISTCPT in the same clock cycle, PSI5SPCIS0.ISTPT remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
1	ISTCRC	<p>Interrupt status of payload data CRC error (Ch0)</p> <p>0: No error 1: Error detected</p> <p>When a PSI5-S frame is stored in a MB and a payload data CRC error occurs, this bit is set to 1 and an interrupt (int_psis_ch0) occurs.</p> <p>This bit is cleared by writing 1 to PSI5SPCISC0.ISTCCRC.</p> <p>When PSI5SPCIS0.ISTCRC is set and 1 is written to PSI5SPCISC0.ISTCCRC in the same clock cycle, PSI5SPCIS0.ISTCRC remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
0	ISTXCRC	<p>Interrupt status of packet frame XCRC error (Ch0)</p> <p>0: No error 1: Error detected</p> <p>When a PSI5-S frame is stored in a MB and an XCRC error occurs, this bit is set to 1 and an interrupt (int_psis_ch0) occurs.</p> <p>This bit is cleared by writing 1 to PSI5SPCISC0.ISTXCRC.</p> <p>When PSI5SPCIS0.ISTXCRC is set and 1 is written to PSI5SPCISC0.ISTXCRC in the same clock cycle, PSI5SPCIS0.ISTXCRC remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>

25.3.7.2 PSI5SPCISC0 — PSI5-S CPU Interrupt Status Clear Ch0 Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ISTCCTFN	—	—	ISTCRFN	ISTCRFEX	ISTCRFLK	ISTCROV	ISTCRWDT	—	ISTCUTFR	ISTCUTPT	ISTCTRST	ISTCPT	ISTCCR	ISTCXCRC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.59 PSI5SPCISC0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 15	—	Reserved These bits are always read as 0. The write value is ignored.
14	ISTCCTFN	Interrupt status of command Tx finish (Ch0) 0: Ignored 1: Clears PSI5SPCISC0.ISTCCTFN This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).
13 to 12	—	Reserved These bits are always read as 0. The write value is ignored.
11	ISTCRFN	Interrupt status of Rx finish (Ch0) 0: Ignored 1: Clears PSI5SPCISC0.ISTRFN This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).
10	ISTCRFEX	Interrupt status of Rx frame excess error (Ch0) 0: Ignored 1: Clears PSI5SPCISC0.ISTRFEX This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).
9	ISTCRFLK	Interrupt status of Rx frame lack error (Ch0) 0: Ignored 1: Clears PSI5SPCISC0.ISTRFLK This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).
8	ISTCROV	Interrupt status of Rx overrun error (Ch0) 0: Ignored 1: Clears PSI5SPCISC0.ISTROV This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).

Table 25.59 PSI5SPCISC0 Register Contents (2/2)

Bit Position	Bit Name	Function
7	ISTCRWDT	<p>Interrupt status of Rx WDT error (Ch0)</p> <p>0: Ignored 1: Clears PSI5SPCIS0.ISTRWDT</p> <p>This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>
6	—	<p>Reserved</p> <p>This bit is always read as 0. The write value is ignored.</p>
5	ISTCUTFR	<p>Interrupt status of Rx UART framing error (Ch0)</p> <p>0: Ignored 1: Clears PSI5SPCIS0.ISTUTFR</p> <p>This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>
4	ISTCUTPT	<p>Interrupt status of Rx UART parity error (Ch0)</p> <p>0: Ignored 1: Clears PSI5SPCIS0.ISTUTPT</p> <p>This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>
3	ISTCTRST	<p>Interrupt status of Rx transceiver status error (Ch0)</p> <p>0: Ignored 1: Clears PSI5SPCIS0.ISTTRST</p> <p>This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>
2	ISTCPT	<p>Interrupt status of payload data parity error (Ch0)</p> <p>0: Ignored 1: Clears PSI5SPCIS0.ISTPT</p> <p>This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>
1	ISTCCRC	<p>Interrupt status of payload data CRC error (Ch0)</p> <p>0: Ignored 1: Clears PSI5SPCIS0.ISTCRC</p> <p>This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>
0	ISTXCRC	<p>Interrupt status of packet frame XCRC error (Ch0)</p> <p>0: Ignored 1: Clears PSI5SPCIS0.ISTXCRC</p> <p>This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>

25.3.8 Ch *n* Register/Config (*n*: 1 to 7)

25.3.8.1 PSI5SPRCF1*n* — PSI5-S Receive Config1 Chn Register (*n*: 1 to 7)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	F6PKT[2:0]			F5PKT[2:0]			F4PKT[2:1]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	F4PKT[0]	F3PKT[2:0]			F2PKT[2:0]			F1PKT[2:0]			SYSEL	TSCSTS	TSCS	TSEN	RFCPS	CHEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.60 PSI5SPRCF1*n* Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 24	—	Reserved These bits are always read as 0. The write value is ignored.
23 to 21	F6PKT[2:0]	<p>Frame 6 packet number (Ch <i>n</i>)</p> <p>000_B: Frame 6 data is ignored</p> <p>011_B: Packet number is to set 3</p> <p>100_B: Packet number is to set 4</p> <p>101_B: Packet number is to set 5</p> <p>110_B: Packet number is to set 6</p> <p>Other than above: Setting prohibited</p> <p>The packet number means the number of UART frames number per packet frame.</p> <p>“Frame 6” is the packet frame whose FID is 101_B.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode).</p> <p>These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
20 to 18	F5PKT[2:0]	<p>Frame 5 packet number (Ch <i>n</i>)</p> <p>000_B: Frame 5 data is ignored</p> <p>011_B: Packet number is to set 3</p> <p>100_B: Packet number is to set 4</p> <p>101_B: Packet number is to set 5</p> <p>110_B: Packet number is to set 6</p> <p>Other than above: Setting prohibited</p> <p>The packet number means the number of UART frames number per packet frame.</p> <p>“Frame 5” is the packet frame whose FID is 100_B.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode).</p> <p>These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>

Table 25.60 PSI5SPRCF1n Register Contents (2/3)

Bit Position	Bit Name	Function
17 to 15	F4PKT[2:0]	<p>Frame 4 packet number (Ch <i>n</i>)</p> <p>000_B: Frame 4 data is ignored 011_B: Packet number is to set 3 100_B: Packet number is to set 4 101_B: Packet number is to set 5 110_B: Packet number is to set 6 Other than above: Setting prohibited</p> <p>The packet number means the number of UART frames number per packet frame. “Frame 4” is the packet frame whose FID is 011_B.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
14 to 12	F3PKT[2:0]	<p>Frame 3 packet number (Ch <i>n</i>)</p> <p>000_B: Frame 3 data is ignored 011_B: Packet number is to set 3 100_B: Packet number is to set 4 101_B: Packet number is to set 5 110_B: Packet number is to set 6 Other than above: Setting prohibited</p> <p>The packet number means the number of UART frames number per packet frame. “Frame 3” is the packet frame whose FID is 010_B.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
11 to 9	F2PKT[2:0]	<p>Frame 2 packet number (Ch <i>n</i>)</p> <p>000_B: Frame 2 data is ignored 011_B: Packet number is to set 3 100_B: Packet number is to set 4 101_B: Packet number is to set 5 110_B: Packet number is to set 6 Other than above: Setting prohibited</p> <p>The packet number means the number of UART frames number per packet frame. “Frame 2” is the packet frame whose FID is 001_B.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
8 to 6	F1PKT[2:0]	<p>Frame 1 packet number (Ch <i>n</i>)</p> <p>000_B: Frame 1 data is ignored 011_B: Packet number is to set 3 100_B: Packet number is to set 4 101_B: Packet number is to set 5 110_B: Packet number is to set 6 Other than above: Setting prohibited</p> <p>The packet number means the number of UART frames number per packet frame. “Frame 1” is the packet frame whose FID is 000_B.</p> <p>These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
5	SYSEL	<p>Asynchronous mode/synchronous mode select</p> <p>0: Synchronous mode 1: Asynchronous mode</p> <p>This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p>

Table 25.60 PSI5SPRCF1n Register Contents (3/3)

Bit Position	Bit Name	Function
4	TSCTS	Timestamp capture trigger select (Ch <i>n</i>) 0: Transmission synchronous pulse timing selected 1: Header receive timing selected (CH <i>n</i>) This bit can be written when PSI5S'PUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
3	TSCS	Timestamp counter select (Ch <i>n</i>) 0: Timestamp counter B selected 1: Timestamp counter A selected This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
2	TSEN	Timestamp capture enable (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
1	RFPCS	Rx frame checksum CRC/parity select (Ch <i>n</i>) 0: Parity selected 1: CRC selected This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
0	CHEN	Channel enable (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: *n*: 1 to 7

25.3.8.2 PSI5SPRCF2n — PSI5-S Receive Config2 Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	F6PAYLD[4:0]				F5PAYLD[4:0]				F4PAYLD[4:1]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	F4PAYLD[0]	F3PAYLD[4:0]				F2PAYLD[4:0]				F1PAYLD[4:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.61 PSI5SPRCF2n Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 30	—	Reserved These bits are always read as 0. The write value is ignored.
29 to 25	F6PAYLD[4:0]	The payload bit length of packet frame 6 (Ch <i>n</i>) (Refer to Figure 25.3 about payload) 0 to 7: Setting prohibited 8: Sets 8 ... x: Sets x ... 28: Sets 28 29 to 31: Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope (<i>n</i> :1 to 7). These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
24 to 20	F5PAYLD[4:0]	The payload bit length of packet frame 5 (Ch <i>n</i>) (Refer to Figure 25.3 about payload) 0 to 7: Setting prohibited 8: Sets 8 ... x: Sets x ... 28: Sets 28 29 to 31: Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope (<i>n</i> :1 to 7). These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

Table 25.61 PSI5SPRCF2n Register Contents (2/3)

Bit Position	Bit Name	Function
19 to 15	F4PAYLD[4:0]	<p>The payload bit length of packet frame 4 (Ch <i>n</i>) (Refer to Figure 25.3 about payload)</p> <p>0 to 7: Setting prohibited 8: Sets 8 ... x: Sets x ... 28: Sets 28 29 to 31: Setting prohibited</p> <p>If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope (<i>n</i>:1 to 7). These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
14 to 10	F3PAYLD[4:0]	<p>The payload bit length of packet frame 3 (Ch <i>n</i>) (Refer to Figure 25.3 about payload)</p> <p>0 to 7: Setting prohibited 8: Sets 8 ... x: Sets x ... 28: Sets 28 29 to 31: Setting prohibited</p> <p>If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope (<i>n</i>:1 to 7). These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
9 to 5	F2PAYLD[4:0]	<p>The payload bit length of packet frame 2 (Ch <i>n</i>) (Refer to Figure 25.3 about payload)</p> <p>0 to 7: Setting prohibited 8: Sets 8 ... x: Sets x ... 28: Sets 28 29 to 31: Setting prohibited</p> <p>If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope (<i>n</i>:1 to 7). These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>

Table 25.61 PSI5SPRCF2n Register Contents (3/3)

Bit Position	Bit Name	Function
4 to 0	F1PAYLD[4:0]	<p>The payload bit length of packet frame 1 (Ch <i>n</i>) (Refer to Figure 25.3 about payload)</p> <p>0 to 7: Setting prohibited 8: Sets 8 ... x: Sets x ... 28: Sets 28 29 to 31: Setting prohibited</p> <p>If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope. (<i>n</i>:1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>

Note: n: 1 to 7

25.3.8.3 PSI5SPWDEn — PSI5-S WDT Enable Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTEB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.62 PSI5SPWDEn Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	WDTEB	Watchdog Timer of Rx frame Enable (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) and when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: n: 1 to 7

25.3.8.4 PSI5SPWDPn — PSI5-S WDT Prescaler Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	WDTPRS[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.63 PSI5SPWDPn Register Contents

Bit Position	Bit Name	Function
31 to 12	—	Reserved These bits are always read as 0. The write value is ignored.
11 to 0	WDTPRS[11:0]	Watchdog timer prescaler (Ch <i>n</i>) 0: Stop WDT timer 1 to 4095: Enabled at 1 clock/ <i>x</i> clock (<i>x</i> :1 to 4095) These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: n: 1 to 7

25.3.8.5 PSI5SPWDEVn — PSI5-S WDT Expiration Value Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	WDTEX[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTEX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.64 PSI5SPWDEVn Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Reserved These bits are always read as 0. The write value is ignored.
23 to 0	WDTEX[23:0]	Watchdog timer expiration value (Ch <i>n</i>) When the watchdog counter counts down to 0 from this setting value, the watchdog timer is judged to have expired. (<i>n</i> : 1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: *n*: 1 to 7

25.3.8.6 PSI5SPTCDn — PSI5-S Tx Command Data Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ATRSCMD[4:0]				ACHID[2:0]			TRSCMD[4:0]				CHID[2:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.65 PSI5SPTCDn Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved These bits are always read as 0. The write value is ignored.
15 to 11	ATRSCMD[4:0]	Alternate transport command (Ch <i>n</i>) When ECU-to-sensor data is 1, select this command. (<i>n</i> : 1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
10 to 8	ACHID[2:0]	Alternate transport ChID (Ch <i>n</i>) When ECU-to-sensor data is 1, select this ChID. (<i>n</i> : 1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
7 to 3	TRSCMD[4:0]	Transport command (Ch <i>n</i>) When ECU-to-sensor data is 0, select this command. (<i>n</i> : 1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
2 to 0	CHID[2:0]	Transport ChID (Ch <i>n</i>) When ECU-to-sensor data is 0, select this ChID. (<i>n</i> : 1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: *n*: 1 to 7

25.3.8.7 PSI5SPCIEn — PSI5-S CPU Interrupt Enable Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	IEBDDSFN	IEBDDSOV	IEBRFN	IEBRFEX	IEBRFLK	IEBROV	IEBRWDT	—	—	—	IEBTRST	IEBPT	IEBCRC	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R

Table 25.66 PSI5SPCIEn Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 14	—	Reserved These bits are always read as 0. The write value is ignored.
13	IEBDDSFN	Interrupt enable of DDSR finish flag (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
12	IEBDDSOV	Interrupt enable of DDSR overwrite (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
11	IEBRFN	Interrupt enable of Rx frame finish flag (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
10	IEBRFEX	Interrupt enable of Rx frame excess error (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
9	IEBRFLK	Interrupt enable of Rx frame lack error (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

Table 25.66 PSI5SPCIEn Register Contents (2/2)

Bit Position	Bit Name	Function
8	IEBROV	Interrupt enable of Rx overrun error (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
7	IEBRWDT	Interrupt enable of Rx frame WDT error (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
6 to 4	—	Reserved These bits are always read as 0. The write value is ignored.
3	IEBTRST	Interrupt enable of Rx transceiver status error (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
2	IEBPT	Interrupt enable of parity error (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
1	IEBCRC	Interrupt enable of CRC error (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
0	—	Reserved This bit is always read as 0. The write value is ignored.

Note: *n*: 1 to 7

25.3.8.8 PSI5SPDREn — PSI5-S DMA Transfer Request Enable Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DRQET FN	DRQE WDT	DRQER FN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 25.67 PSI5SPDREn Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved These bits are always read as 0. The write value is ignored.
2	DRQETFN	DMA request enable at ddsr Tx finish (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
1	DRQEWDT	DMA request enable at WDT (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
0	DRQERFN	DMA request enable at Rx finish (Ch <i>n</i>) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: *n*: 1 to 7

25.3.8.9 PSI5SPSTPn — PSI5-S Sync Trigger Prescaler Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	STPRS[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.68 PSI5SPSTPn Register Contents

Bit Position	Bit Name	Function
31 to 12	—	Reserved These bits are always read as 0. The write value is ignored.
11 to 0	STPRS[11:0]	Synchronous trigger generation counter's prescaler (Ch <i>n</i>) 0: Enabled at 1 clock/1 clock 1: Enabled at 1 clock/2 clock ... x: Enabled at 1 clock/(x+1) clock ... 4095: Enabled at 1 clock/4096 clock When PSI5SPSTSn.STSEL is 0, these bits are enabled. When PSI5SPSTSn.STSEL is 1, these bits are disabled. These bits can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: *n*: 1 to 7

25.3.8.10 PSI5SPSTEVn — PSI5-S Sync Trigger Expiration Value Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	STEX[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STEX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.69 PSI5SPSTEVn Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Reserved These bits are always read as 0. The write value is ignored.
23 to 0	STEX[23:0]	Synchronous trigger generation counter expiration value (Ch <i>n</i>) When PSI5SPSTSn.STSEL is 0, these bits are enabled. When PSI5SPSTSn.STSEL is 1, these bits are disabled. These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: *n*: 1 to 7

25.3.8.11 PSI5SPSTSn — PSI5-S Sync Trigger Select Ch n Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STSEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.70 PSI5SPSTSn Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	STSEL	Sync trigger select Synchronous trigger select (Ch <i>n</i>) 0: Signal generated by PSI5-S is selected 1: GTM output is selected, This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: *n*: 1 to 7

25.3.9 Ch *n* Register/Rx (*n*: 1 to 7)

25.3.9.1 PSI5SPRESn — PSI5-S Receive Error Status Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	RERRF6	RERRF5	RERRF4	RERRF3	RERRF2	RERRF1
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.71 PSI5SPRESn Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 6	—	Reserved These bits are always read as 0. The write value is ignored.
5	RERRF6	Rx error at packet frame 6 (Ch <i>n</i>) 0: No error 1: An error has occurred This bit is set to 1 when any of “Rx overrun error”, “transceiver status error”, “payload data parity error”, or “payload data CRC error” in packet frame6 of Ch <i>n</i> has occurred. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).
4	RERRF5	Rx error at packet frame 5 (Ch <i>n</i>) 0: No error 1: An error has occurred This bit is set to 1 when any of “Rx overrun error”, “transceiver status error”, “payload data parity error”, or “payload data CRC error” in packet frame5 of Ch <i>n</i> has occurred. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).
3	RERRF4	Rx error at packet frame 4 (Ch <i>n</i>) 0: No error 1: An error has occurred This bit is set to 1 when any of “Rx overrun error”, “transceiver status error”, “payload data parity error”, or “payload data CRC error” in packet frame4 of Ch <i>n</i> has occurred. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).
2	RERRF3	Rx error at packet frame 3 (Ch <i>n</i>) 0: No error 1: An error has occurred This bit is set to 1 when any of “Rx overrun error”, “transceiver status error”, “payload data parity error”, or “payload data CRC error” in packet frame3 of Ch <i>n</i> has occurred. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).

Table 25.71 PSI5SPRESn Register Contents (2/2)

Bit Position	Bit Name	Function
1	RERRF2	<p>Rx error at packet frame 2 (Ch <i>n</i>)</p> <p>0: No error 1: An error has occurred</p> <p>This bit is set to 1 when any of “Rx overrun error”, “transceiver status error”, “payload data parity error”, or “payload data CRC error” in packet frame2 of Ch <i>n</i> has occurred. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
0	RERRF1	<p>Rx error at packet frame 1 (Ch <i>n</i>)</p> <p>0: No error 1: An error has occurred</p> <p>This bit is set to 1 when any of “Rx overrun error”, “transceiver status error”, “payload data parity error”, or “payload data CRC error” in packet frame1 of Ch <i>n</i> has occurred. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>

Note: *n*: 1 to 7

25.3.9.2 PSI5SPRESCn — PSI5-S Receive Error Status Clear Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	RERRC LF6	RERRC LF5	RERRC LF4	RERRC LF3	RERRC LF2	RERRC LF1
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.72 PSI5SPRESCn Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 6	—	Reserved These bits are always read as 0. The write value is ignored.
5	RERRCLF6	Rx error clear packet Frame6 (Ch <i>n</i>) 0: Ignored 1: Clear PSI5SPRES _{<i>n</i>} .RERRF6 This bit is always read as 0. (<i>n</i> : 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).
4	RERRCLF5	Rx error clear packet Frame5 (Ch <i>n</i>) 0: Ignored 1: Clear PSI5SPRES _{<i>n</i>} .RERRF5 This bit is always read as 0. (<i>n</i> : 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).
3	RERRCLF4	Rx error clear packet Frame4 (Ch <i>n</i>) 0: Ignored 1: Clear PSI5SPRES _{<i>n</i>} .RERRF4 This bit is always read as 0. (<i>n</i> : 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).
2	RERRCLF3	Rx error clear packet Frame3 (Ch <i>n</i>) 0: Ignored 1: Clear PSI5SPRES _{<i>n</i>} .RERRF3 This bit is always read as 0. (<i>n</i> : 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).

Table 25.72 PSI5SPRESCn Register Contents (2/2)

Bit Position	Bit Name	Function
1	RERRCLF2	<p>Rx error clear packet Frame2 (Ch <i>n</i>)</p> <p>0: Ignored</p> <p>1: Clear PSI5SPRES<i>n</i>.RERRF2</p> <p>This bit is always read as 0.</p> <p>(<i>n</i>: 1 to 7)</p> <p>This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>
0	RERRCLF1	<p>Rx error clear packet Frame1 (Ch <i>n</i>)</p> <p>0: Ignored</p> <p>1: Clear PSI5SPRES<i>n</i>.RERRF1</p> <p>This bit is always read as 0.</p> <p>(<i>n</i>: 1 to 7)</p> <p>This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>

Note: *n*: 1 to 7

25.3.9.3 PSI5SPTCDTn — PSI5-S Timestamp Capture Data Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TSCD[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSCD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.73 PSI5SPTCDTn Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Reserved These bits are always read as 0. The write value is ignored.
23 to 0	TSCD[23:0]	Timestamp capture data (Ch <i>n</i>) These bits are cleared by writing 1 to PSI5SPTCDn.TSCCLR. These bits are cleared by writing 0 to PSI5SPRCF1n.TSEN. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: *n*: 1 to 7

25.3.9.4 PSI5SPTCDCn — PSI5-S Timestamp Capture Data Clear Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.74 PSI5SPTCDCn Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	TSCCLR	Timestamp capture clear (Ch <i>n</i>) 0: Ignored 1: Clear timestamp capture data (PSI5SPDCD <i>n</i> .TSCD) This bit is always read as 0. (<i>n</i> : 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).

Note: *n*: 1 to 7

25.3.10 Ch *n* Register/Tx (*n*: 1 to 7)

25.3.10.1 PSI5SPDDTP_{*n*} — PSI5-S DDSR Type Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDSRTYPE[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Note: *n*: 1 to 7

Table 25.75 PSI5SPDDTP_{*n*} Register Contents

Bit Position	Bit Name	Function
31 to 2	—	.Reserved These bits are always read as 0. The write value is ignored.
1 to 0	DDSRTYPE[1:0]	DDSR transmission type (Ch <i>n</i>) (<i>n</i> : 1 to 7) 00 _B : Frame 1 (Short) 01 _B : Frame 2 (Long) 10 _B : Frame 3 (XLong) 11 _B : Frame 4 (XXLong) Writing to these bits is prohibited when PSI5SPDDSn.DDSRSTS is 1. These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.

25.3.10.2 PSI5SPDDn — PSI5-S DDSR Data Chn Register

Value after reset: 00FF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DDSRDT[19:12]							
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DDSRDT[11:0]											DDSRADR[3:0]				
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.76 PSI5SPDDn Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Reserved These bits are always read as 0. The write value is ignored.
23 to 4	DDSRDT[19:0]	DDSR transmission data (Ch <i>n</i>) These bits cannot be written when PSI5SPDDSn.DDSRSTS is 1. When PSI5SPDDTPn.DDSRTYPE is 0, DDSR transmission data uses the 3 LSBs. PSI5SPDDn.DDSRDT [19:3] should be set to all 1's. When PSI5SPDDTPn.DDSRTYPE is 1, DDSR transmission data uses the 13 LSBs. PSI5SPDDn.DDSRDT [19:13] should be set to all 1's. When PSI5SPDDTPn.DDSRTYPE is 2, DDSR transmission data uses the 19 LSBs. PSI5SPDDn.DDSRDT [19] should be set to all 1's. When PSI5SPDDTPn.DDSRTYPE is 3, DDSR transmission data uses 20bits. These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
3 to 0	DDSRADR[3:0]	DDSR transmission address (Ch <i>n</i>) These bits cannot be written when PSI5SPDDSn.DDSRSTS is 1. (<i>n</i> : 1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0 (= configuration mode) or when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: *n*: 1 to 7

25.3.10.3 PSI5SPDDSn — PSI5-S DDSR Status Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDSRSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.77 PSI5SPDDSn Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	DDSRSTS	DDSR status 0: DDSR transmission is not in progress 1: DDSR transmission is in progress In PSI5S mode, writing to PSI5SPDDDn sets this bit to 1. When the last data of PSI5SPDDDn is written to the Tx shifter, this bit becomes 0. PSI5SPDDDn is cannot written when PSI5SPDDSn.DDSRSTS is 1. This bit is cleared by writing 1 to PSI5SPDDSPn.DDSRSTP. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).

Note: *n*: 1 to 7

25.3.10.4 PSI5SPDDSPn — PSI5-S DDSR Stop Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDSRSTP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.78 PSI5SPDDSPn Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value is ignored.
0	DDSRSTP	DDSR Tx stop (Ch <i>n</i>) 0: Ignored 1: Stop transmission When this bit is written, PSI5SPDDSPn.DDSRSTS is reset to 0. This bit is always read as 0. (<i>n</i> : 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).

Note: *n*: 1 to 7

25.3.11 Ch *n* Register/Interrupt (*n*: 1 to 7)

25.3.11.1 PSI5SPCIS_{*n*} — PSI5-S CPU Interrupt Status Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	ISTDDSFN	ISTDDSOW	ISTRFN	ISTRFX	ISTRFLK	ISTROV	ISTRWDT	—	—	—	ISTTRST	ISTPT	ISTCRC	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.79 PSI5SPCIS_{*n*} Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 14	—	Reserved These bits are always read as 0. The write value is ignored.
13	ISTDDSFN	CPU interrupt status of DDSR finish (Ch <i>n</i>) 0: DDSR transmission has not finished 1: DDSR transmission has finished When PSI5SPDDD _{<i>n</i>} data is written to the Tx shifter, this bit is set to 1 and an interrupt (int_psis_chn) and DMA request (dma_psis_chn_tx) occur. (<i>n</i> = 1 to 7) This bit is cleared by writing 1 to PSI5SPCISC _{<i>n</i>} .ISTCDDSFN. When PSI5SPCIS _{<i>n</i>} .ISTDDSFN is set and 1 is written to PSI5SPCISC _{<i>n</i>} .ISTCDDSFN in the same clock cycle, PSI5SPCIS _{<i>n</i>} .ISTDDSFN remains 1. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).
12	ISTDDSOW	CPU interrupt status of DDSR overwrite error (Ch <i>n</i>) 0: No error 1: Error detected When the CPU writes 1 to PSI5SPDDD _{<i>n</i>} in PSI5SPDDS _{<i>n</i>} .DDSRSTS, this bit is set to 1 and an interrupt (int_psis_chn) occurs. (<i>n</i> : 1 to 7) This bit is cleared by writing 1 to PSI5SPCISC _{<i>n</i>} .ISTCDDSOW. When PSI5SPCIS _{<i>n</i>} .ISTDDSOW is set and 1 is written to PSI5SPCISC _{<i>n</i>} .ISTCDDSOW in the same clock cycle, PSI5SPCIS _{<i>n</i>} .ISTDDSOW remains 1. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).

Table 25.79 PSI5SPCISn Register Contents (2/3)

Bit Position	Bit Name	Function
11	ISTRFN	<p>Interrupt status of Rx finish (Ch <i>n</i>)</p> <p>0: PSI5-S frame is not received successfully 1: PSI5-S frame is received successfully</p> <p>When a PSI5-S frame is stored in a MB and the PSI5-S frame has no errors (no mailbox overrun error), this bit is set to 1 and an interrupt (int_psis_chn) and DMA request (dma_psis_chn_rx) occur.</p> <p>(<i>n</i>: 1 to 7)</p> <p>This bit is cleared by writing 1 to PSI5SPCISCn.ISTCRFN.</p> <p>When PSI5SPCISn.ISTRFN is set and 1 is written to PSI5SPCISCn.ISTCRFN in the same clock cycle, PSI5SPCISn.ISTRFN remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
10	ISTRFEX	<p>Interrupt status of Rx frame excess error (Ch0)</p> <p>0: No error 1: Error detected</p> <p>When a packet over PSI5SPRCF1n.FmPKT (<i>m</i> = 1 to 6) is received, this bit is set to 1 and an interrupt (int_psis_chn) occurs.</p> <p>This bit is cleared by writing 1 to PSI5SPCISCn.ISTCRFEX.</p> <p>When PSI5SPCISn.ISTRFEX is set and 1 is written to PSI5SPCISCn.ISTCRFEX in the same clock cycle, PSI5SPCISn.ISTRFEX remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
9	ISTRFLK	<p>Interrupt status of Rx frame lack error (Ch0)</p> <p>0: No error 1: Error detected</p> <p>When a packet frame gap is detected while the packet is under PSI5SPRCF1n.FmPKT (<i>m</i> = 1 to 6), this bit is set to 1 and an interrupt (int_psis_chn) occurs.</p> <p>(<i>n</i>: 1 to 7)</p> <p>This bit is cleared by writing 1 to PSI5SPCISCn.ISTCRFLK.</p> <p>When PSI5SPCISn.ISTRFLK is set and 1 is written to PSI5SPCISCn.ISTCRFLK in the same clock cycle, PSI5SPCISn.ISTRFLK remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
8	ISTROV	<p>Interrupt status of Rx overrun error (Ch <i>n</i>)</p> <p>0: No error 1: Error detected</p> <p>When an attempt is made to store the next PSI5-S frame in a MB before the CPU has read that MB, this bit is set to 1 and an interrupt (int_psis_chn) occurs.</p> <p>(<i>n</i>: 1 to 7)</p> <p>This bit is cleared by writing 1 to PSI5SPCISCn.ISTCROV.</p> <p>When PSI5SPCISn.ISTROV is set and 1 is written to PSI5SPCISCn.ISTCROV in the same clock cycle, PSI5SPCISn.ISTROV remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>

Table 25.79 PSI5SPCISn Register Contents (3/3)

Bit Position	Bit Name	Function
7	ISTRWDT	<p>Interrupt status of Rx WDT error (Ch <i>n</i>)</p> <p>0: No error 1: Error detected</p> <p>When an WDT error occurs in PSI5S mode, this bit is set to 1 and an interrupt (int_psis_chn) and DMA request (dma_psis_chn_rx) occur.</p> <p>(<i>n</i>: 1 to 7)</p> <p>This bit is cleared by writing 1 to PSI5SPCISn.ISTRWDT.</p> <p>When PSI5SPCISn.ISTRWDT is set and 1 is written to PSI5SPCISn.ISTRWDT in the same clock cycle, PSI5SPCISn.ISTRWDT remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
6 to 4	—	<p>Reserved</p> <p>These bits are always read as 0. The write value is ignored. *1</p>
3	ISTTRST	<p>Interrupt status of Rx transceiver status error (Ch <i>n</i>)</p> <p>0: No error 1: Error detected</p> <p>When a PSI5-S frame is stored in a MB and a transceiver status error occurs, this bit is set to 1 and an interrupt (int_psis_chn) occurs.</p> <p>(<i>n</i>: 1 to 7)</p> <p>This bit is cleared by writing 1 to PSI5SPCISn.ISTTRST.</p> <p>When PSI5SPCISn.ISTTRST is set and 1 is written to PSI5SPCISn.ISTTRST in the same clock cycle, PSI5SPCISn.ISTTRST remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
2	ISTPT	<p>Interrupt status of payload data parity error (Ch <i>n</i>)</p> <p>0: No error 1: Error detected</p> <p>When a PSI5-S frame is stored in a MB and a payload data parity error occurs, this bit is set to 1 and an interrupt (int_psis_chn) occurs.</p> <p>(<i>n</i>: 1 to 7)</p> <p>This bit is cleared by writing 1 to PSI5SPCISn.ISTPT.</p> <p>When PSI5SPCISn.ISTPT is set and 1 is written to PSI5SPCISn.ISTPT in the same clock cycle, PSI5SPCISn.ISTPT remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
1	ISTCRC	<p>Interrupt status of payload data CRC error (Ch <i>n</i>)</p> <p>0: No error 1: Error detected</p> <p>When a PSI5-S frame is stored in a MB and a payload data CRC error occurs, this bit is set to 1 and an interrupt (int_psis_chn) occurs.</p> <p>(<i>n</i>: 1 to 7)</p> <p>This bit is cleared by writing 1 to PSI5SPCISn.ISTCRC.</p> <p>When PSI5SPCISn.ISTCRC is set and 1 is written to PSI5SPCISn.ISTCRC in the same clock cycle, PSI5SPCISn.ISTCRC remains 1.</p> <p>This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>This bit is cleared when PSI5SPUOS.ACSTS is changed to 0 (= configuration mode).</p>
0	—	<p>Reserved</p> <p>This bit is always read as 0. The write value is ignored. *1</p>

Note: *n*: 1 to 7

Note 1. b[5][4][0]: These bits are not reserved in not Ch0. If these errors (UART framing error, UART parity error, XCRC Error) occur, the error is stored in channel 0, frame 2.

25.3.11.2 PSI5SPCISn — PSI5-S CPU Interrupt Status Clear Chn Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	ISTCDD SFN	ISTCDD SOW	ISTCRF N	ISTCRF EX	ISTCRF LK	ISTCR OV	ISTCR WDT	—	—	—	ISTCTR ST	ISTCPT	ISTCCR C	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R

Table 25.80 PSI5SPCISn Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 14	—	Reserved These bits are always read as 0. The write value is ignored.
13	ISTCDDSFN	Interrupt status clear DDSR Tx finish (Chn) 0: Ignored 1: Clear PSI5SPCISn.ISTDDSFN This bit defines about clearance of CPU interrupt status of DDSR finish (Ch n) in PSI5S mode (PSI5SPCISn.ISTDDSFN). When this bit is written to 1, PSI5SPCIS0.ISTDDSFN is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).
12	ISTCDDSOW	Interrupt status clear DDSr overwrite error Clear at CPU interrupt status of DDSR overwrite error(Chn) 0: Ignored 1: Clear PSI5SPCISn.ISTDDSOW This bit defines about clearance of CPU interrupt status of DDSR overwrite (Ch n) in PSI5S mode (PSI5SPCISn.ISTDDSOW). When this bit is written to 1, PSI5SPCIS0.ISTDDSOW is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).
11	ISTCRFN	Interrupt status clear Rx finish Clear at CPU interrupt status of Rx finish (Chn) 0: Ignored 1: Clear PSI5SPCISn.ISTRFN This bit defines about clearance of CPU interrupt status of Rx finish (Ch n) in PSI5S mode (PSI5SPCISn.ISTRFN). When this bit is written to 1, PSI5SPCISn.ISTRFN is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).

Table 25.80 PSI5SPCISn Register Contents (2/3)

Bit Position	Bit Name	Function
10	ISTCRFEX	<p>Interrupt status clear Rx frame excess error Clear at CPU interrupt status of Rx frame excess error (Chn) 0: Ignored 1: Clear PSI5SPCISn.ISTRFEX</p> <p>This bit defines about clearance of CPU interrupt status of Rx frame (packet) (Ch n) excess error in PSI5S mode (PSI5SPCISn.ISTRFEX). When this bit is written to 1, PSI5SPCISn.ISTRFEX is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>
9	ISTCRFLK	<p>Interrupt status clear Rx frame lack error Clear at CPU interrupt status of Rx frame lack error (Chn) 0: Ignored 1: Clear PSI5SPCISn.ISTRFLK</p> <p>This bit defines about clearance of CPU interrupt status of Rx frame (packet) (Ch n) lack error (Chn) in PSI5S mode (PSI5SPCISn.ISTRFLK). When this bit is written to 1, PSI5SPCISn.ISTRFLK is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>
8	ISTCROV	<p>Interrupt status clear Rx overrun error Clears at CPU interrupt status of Rx overrun error (Chn) 0: Ignored 1: Clear PSI5SPCISn.ISTROV</p> <p>This bit defines about clearance of CPU interrupt status of overrun error (Ch n) in PSI5S mode (PSI5SPCISn.ISTROV). When this bit is written to 1, PSI5SPCISn.ISTROV is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>
7	ISTCRWDT	<p>Interrupt status clear Rx WDT error Clear at CPU interrupt status of Rx WDT error (Chn) 0: Ignored 1: Clear PSI5SPCISn.ISTRWDT</p> <p>This bit defines about clearance of CPU interrupt status of WDT error (Ch n) in PSI5S mode (PSI5SPCISn.ISTRWDT). When this bit is written to 1, PSI5SPCISn.ISTRWDT is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>
6 to 4	—	<p>Reserved These bits are always read as 0. The write value is ignored.</p>

Table 25.80 PSI5SPCISn Register Contents (3/3)

Bit Position	Bit Name	Function
3	ISTCTRST	<p>Interrupt status clear RX transceiver status error Clear at CPU interrupt status of Rx transceiver status error (Chn) 0: Ignored 1: Clear PSI5SPCISn.ISTTRST</p> <p>This bit defines about clearance of CPU interrupt status of transceiver status error (Ch n) in PSI5S mode (PSI5SPCISn.ISTTRST). When this bit is written to 1, PSI5SPCISn.ISTTRST is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>
2	ISTCPT	<p>Interrupt status clear parity error Clear at CPU interrupt status of payload data parity error (Chn) 0: Ignored 1: Clear PSI5SPCISn.ISTPT</p> <p>This bit defines about clearance of CPU interrupt status of payload data parity error (Ch n) in PSI5S mode (PSI5SPCISn.ISTPT). When this bit is written to 1, PSI5SPCISn.ISTPT is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>
1	ISTCCRC	<p>Interrupt status clear CRC error Clear at CPU interrupt status of payload data CRC error (Chn) 0: Ignored 1: Clear PSI5SPCISn.ISTCRC</p> <p>This bit defines about clearance of CPU interrupt status of payload data CRC error (Ch n) in PSI5S mode (PSI5SPCISn.ISTCRC). When this bit is written to 1, PSI5SPCISn.ISTCRC is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 1 and PSI5SPUOS.MSTS is 1 (= PSI5S mode).</p>
0	—	<p>Reserved This bit is always read as 0. The write value is ignored.</p>

Note: n: 1 to 7

25.3.12 Ch 0 Frm m MB Data (m: 1, 2)

25.3.12.1 PSI5SPMB0mS — PSI5-S Receive MailBox Ch0 Frmm Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCI[3:0]			—	—	—	CHID[2:0]			FID[2:0]			MBORE RR	WDTER R	—	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UTFRE RR	UTPTE RR	HEADE RR	HEADST[1:0]		CRCER R	CRC[2:0]		XCRCE RR	XCRC[5:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.81 PSI5SPMB0mS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	DCI[3:0]	DCI value (Ch0, Frm <i>m</i>) The DCI value is generated by a 4-bit counter, and every time the PSI5 frame data is restored, it is incremented by 1. These bits are read only. The write value is ignored. (m: 1, 2) These bits are cleared by writing 1 to PSI5SPRMB0.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
27 to 25	—	Reserved These bits are always read as 0. The write value is ignored.
24 to 22	CHID[2:0]	Rx channel ID (Ch0, Frm <i>m</i>) These bits are cleared by writing 1 to PSI5SPRMB0.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
21 to 19	FID[2:0]	Rx frame ID (Ch0, Frm <i>m</i>) These bits are cleared by writing 1 to PSI5SPRMB0.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
18	MBOREERR	Mailbox overrun error (Ch0, Frm <i>m</i>) 0: No error 1: Error detected This bit is cleared by writing 1 to PSI5SPRMB0.MBCLR. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
17	WDTERR	Rx frame WDT error (Ch0, Frm <i>m</i>) 0: No error 1: Error detected This bit is cleared by writing to PSI5SPRMB0.MBCLR. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
16	—	Reserved This bit is always read as 0. The write value is ignored.

Table 25.81 PSI5SPMB0mS Register Contents (2/2)

Bit Position	Bit Name	Function
15	UTFRERR	<p>UART framing error (Ch0, Frm2) *1</p> <p>0: No error 1: Error detected</p> <p>This bit is cleared by writing 1 to PSI5SPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>When this error occurs, data is stored in channel 0, frame 2. So, this error exists only in channel 0, frame 2.</p> <p>*1 This bit is only in 50C_H (Frm2)</p>
14	UTPTERR	<p>UART parity error (Ch0, Frm2) *1</p> <p>0: No error 1: Error detected</p> <p>This bit is cleared by writing 1 to PSI5SPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>When this error occurs, data is stored in channel 0, frame 2. So, this error exists only in channel 0, frame 2.</p> <p>*1 This bit is only in 50C_H (Frm2)</p>
13	HEADERR	<p>Header error (Ch0, Frm <i>m</i>)</p> <p>0: No error 1: Error detected</p> <p>This bit is cleared by writing 1 to PSI5SPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
12 to 11	HEADST[1:0]	<p>Header status</p> <p>These bits are cleared by writing 1 to PSI5SPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
10	CRCERR	<p>Rx CRC/Parity error (Ch0, Frm <i>m</i>)</p> <p>0: No error 1: Error detected</p> <p>When PSI5SPRCF10.RFCPS is 1, this bit shows a CRC error. When PSI5SPRCF10.RFCPS is 0, this bit shows a parity error.</p> <p>This bit is cleared by writing 1 to PSI5SPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
9 to 7	CRC[2:0]	<p>Rx frame CRC/Parity (Ch0, Frm <i>m</i>)*2</p> <p>When PSI5SPRCF10.RFCPS is 1, these bits show the CRC (3bits). When PSI5SPRCF10.RFCPS is 0, bit [9] shows the parity, and bits [8:7] are reserved.</p> <p>These bits are cleared by writing 1 to PSI5SPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>
6	XCRCERR	<p>Rx XCRC error (Ch0, Frm2) *1</p> <p>0: No error 1: Error detected</p> <p>This bit is cleared by writing 1 to PSI5SPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.</p> <p>When this error occurs, data is stored in channel 0, frame 2. So, this error exists only in channel 0, frame 2.</p> <p>*1 This bit is only in 50C_H (Frm2)</p>
5 to 0	XCRC[5:0]	<p>Rx frame XCRC (Ch0, Frm <i>m</i>)</p> <p>These bits are cleared by writing 1 to PSI5SPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.</p>

Note: m: 1, 2

Frm, frame: packet frame

Note 1. Bits [15], [14] and [6] are used only in 50C_H (Frm2)

Note 2. When PSI5SPRCF10.RFCPS is 1, bits [9:7] are the CRC, and when PSI5SPRCF10.RFCPS is 0, bit [9] is the parity, bits [8:7] are reserved.

25.3.12.2 PSI5SPMB0mD — PSI5-S Receive MailBox Ch0 Frmm Data Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCI[3:0]				DATA[27:16]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.82 PSI5SPMB0mD Register Contents

Bit Position	Bit Name	Function
31 to 28	DCI[3:0]	DCI value (Ch0, Frm <i>m</i>) These bits are cleared by writing 1 to PSI5SPRMB0.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
27 to 0	DATA[27:0]	Message data (Ch0, Frm <i>m</i>) When the number of payloads (PSI5SPRCF20.FmPAYLD) is less than 28, the PSI5-S stores the payload from the LSB and the MSB is filled with 0. These bits are cleared by writing 1 to PSI5SPRMB0.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: m: 1, 2

Frm, frame: packet frame

25.3.12.3 PSI5SPMB0mT — PSI5-S Receive MailBox Ch0 Frmm Timestamp Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCI[3:0]				—	—	—	—	TMST[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMST[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.83 PSI5SPMB0mT Register Contents

Bit Position	Bit Name	Function
31 to 28	DCI[3:0]	DCI value (CH0, Frm <i>m</i>) These bits are cleared by writing 1 to PSI5SPRMB0.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
27 to 24	—	Reserved These bits are always read as 0. The write value is ignored.
23 to 0	TMST[23:0]	Timestamp data (CH0, Frm <i>m</i>) These bits are cleared by writing 1 to PSI5SPRMB0.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: m: 1, 2

Frm, frame: packet frame

25.3.13 Ch n Frm m MB Data (n: 1 to 7) (m: 1 to 6)

25.3.13.1 PSI5SPMBnmS — PSI5-S Receive MailBox ch n Frm m Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCI[3:0]			—	—	—	CHID[2:0]			FID[2:0]			MBORERR	—	—	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	HEADERR	HEADST[1:0]	CRCERR	CRC[2:0]			—	XCRC[5:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.84 PSI5SPMBnmS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	DCI[3:0]	DCI value (Chn, Frm m) The DCI value is generated by a 4-bit counter, and every time the PSI5 frame data is restored, the count is incremented by 1. These bits are cleared by writing 1 to PSI5SPRMB.CMBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
27 to 25	—	Reserved These bits are always read as 0. The write value is ignored.
24 to 22	CHID[2:0]	Rx channel ID (Chn, Frm m) These bits are cleared by writing 1 to PSI5SPRMB.CMBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
21 to 19	FID[2:0]	Rx frame ID (Chn, Frm m) These bits are cleared by writing 1 to PSI5SPRMB.CMBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
18	MBORERR	Mailbox overrun error (Chn, Frm m) 0: No error 1: Error detected This bit is cleared by writing 1 to PSI5SPRMB.CMBCLR. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
17 to 14	—	Reserved These bits are always read as 0. The write value is ignored.
13	HEADERR	Header error (Chn, Frm m) 0: No error 1: Error detected This bit is cleared by writing 1 to PSI5SPRMB.CMBCLR. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
12 to 11	HEADST[1:0]	Header status (Chn, Frm m) These bits are cleared by writing 1 to PSI5SPRMB.CMBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

Table 25.84 PSI5SPMBnmS Register Contents (2/2)

Bit Position	Bit Name	Function
10	CRCERR	Rx CRC/Parity error (Chn, Frm <i>m</i>) 0: No error 1: Error detected When PSI5SPRCF1 <i>m</i> .RFCPS is 1, this bit shows a CRC error. And when PSI5SPRCF1 <i>m</i> .RFCPS is 0, this bit shows a parity error. This bit is cleared by writing 1 to PSI5SPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SPUSWR.SWRST.
9 to 7	CRC[2:0]	Rx frame CRC/Parity (Chn, Frm <i>m</i>)* ¹ These bits show the Rx frame CRC/Parity. (Ch <i>n</i> , Frm <i>m</i>) When PSI5SPRCF1 <i>m</i> .RFCPS is 1, these bits show the CRC (3bits). When PSI5SPRCF1 <i>m</i> .RFCPS is 0, bit [9] shows the parity, and bits [8:7] are reserved. These bits are cleared by writing 1 to PSI5SPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
6	—	Reserved This bit is always read as 0. The write value is ignored.
5 to 0	XCRC[5:0]	Rx frame XCRC (Chn, Frm <i>m</i>) These bits are cleared by writing 1 to PSI5SPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: *n*: 1 to 7

m: 1 to 6

Frm, frame: packet frame

Bits [15], [14] and [6]: These bits are not reserved in channel 0. When these errors (UART framing error, UART parity error, XCRC Error) occur, the error is stored in channel 0, frame 2.

Note 1. When PSI5SPRCF1*n*.RFCPS is 1 bits [9:7] are the CRC, and when PSI5SPRCF1*n*.RFCPS is 0 bit [9] is the parity, and bits [8:7] are reserved.

25.3.13.2 PSI5SPMBnmD — PSI5-S Receive MailBox Ch0 Frmm Data Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCI[3:0]				DATA[27:16]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.85 PSI5SPMBnmD Register Contents

Bit Position	Bit Name	Function
31 to 28	DCI[3:0]	DCI value (Ch <i>n</i> , Frm <i>m</i>) These bits are cleared by writing 1 to PSI5SPRMBBC.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
27 to 0	DATA[27:0]	Rx message data (Ch <i>n</i> , Frm <i>m</i>) When the number of payloads (PRCF1n.FmPKT) is less than 28, the PSI5-S stores the payload from the LSB and the MSB is filled with 0. These bits are cleared by writing 1 to PSI5SPRMBBC.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: *n*: 1 to 7
m: 1 to 6
 Frm, frame: packet frame

25.3.13.3 PSI5SPMBnmT — PSI5-S Receive MailBox Chn Frmm Timestamp Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCI[3:0]				—	—	—	—	TMST[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMST[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.86 PSI5SPMBnmT Register Contents

Bit Position	Bit Name	Function
31 to 28	DCI[3:0]	DCI value (Ch <i>n</i> , Frm <i>m</i>) These bits are cleared by writing 1 to PSI5SPRMBBC.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.
27 to 24	—	Reserved These bits are always read as 0. The write value is ignored.
23 to 0	TMST[23:0]	Timestamp data (Ch <i>n</i> , Frm <i>m</i>) These bits are cleared by writing 1 to PSI5SPRMBBC.MBCLR. These bits are cleared by writing 1 to PSI5SPUSWR.SWRST.

Note: *n*: 1 to 7

m: 1 to 6

Frm, frame: packet frame

25.4 Operation Modes

PSI5-S has three operation modes: Configuration, PSI5S, and UART modes.

The operation mode state machine controls the transition of PSI5-S between these operation modes.

Figure 25.5 shows the state transitions controlled by the operation mode state machine.

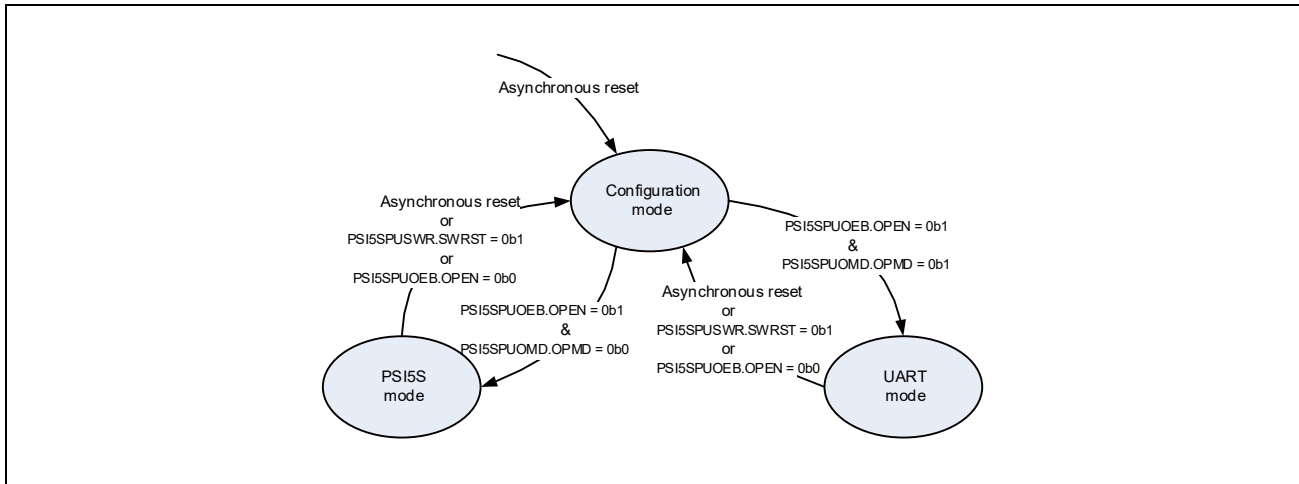


Figure 25.5 State Transitions by Operation Mode State Machine

Table 25.87 below describes the individual states:

Table 25.87 Operation Modes (States)

No	Operation Mode	Explanation
1	Configuration mode	Configuration operation mode • Setting SFR of PSI5-S.
2	PSI5S mode	PSI5-S operation mode • Communication of a PSI5 frame is performed.
3	UART mode	UART operation mode • Communication of UART packet frame.

When the PSI5-S is not transmitting after 16 PCLK cycles of setting of returning to the configuration mode from PSI5S mode or UART mode, the PSI5-S returns to the configuration mode.

When the PSI5-S is transmitting after 16 PCLK cycles of setting of returning to the configuration mode, the PSI5-S returns to the configuration mode in the following timing:

In UART mode: The PSI5-S ends transmission operation of the 2 UART frames at the maximum.

In PSI5S mode: The PSI5-S ends transmission operation of the 9 UART frames at the maximum.

NOTE

Even during reception, the PSI5-S stops reception operation and returns to the reception standby state.

Interrupt and DMA request when the PSI5-S return to the configuration mode is following:

Interrupt: PSI5-S does not generate interrupts after 14 or more PCLK cycles of returning to the configuration mode.

Note that, because interrupt status flags are cleared when the PSI5-S returns to the configuration mode, interrupt status cannot be read if the PSI5-S returns to the configuration mode after it has issued an interrupt.

DMA request: PSI5-S does not issue DMA requests after 14 or more PCLK cycles of returning to the configuration mode.

25.5 Setting Procedures

Figure 25.6 shows the setting procedure to use PSI5-S in PSI5S mode.

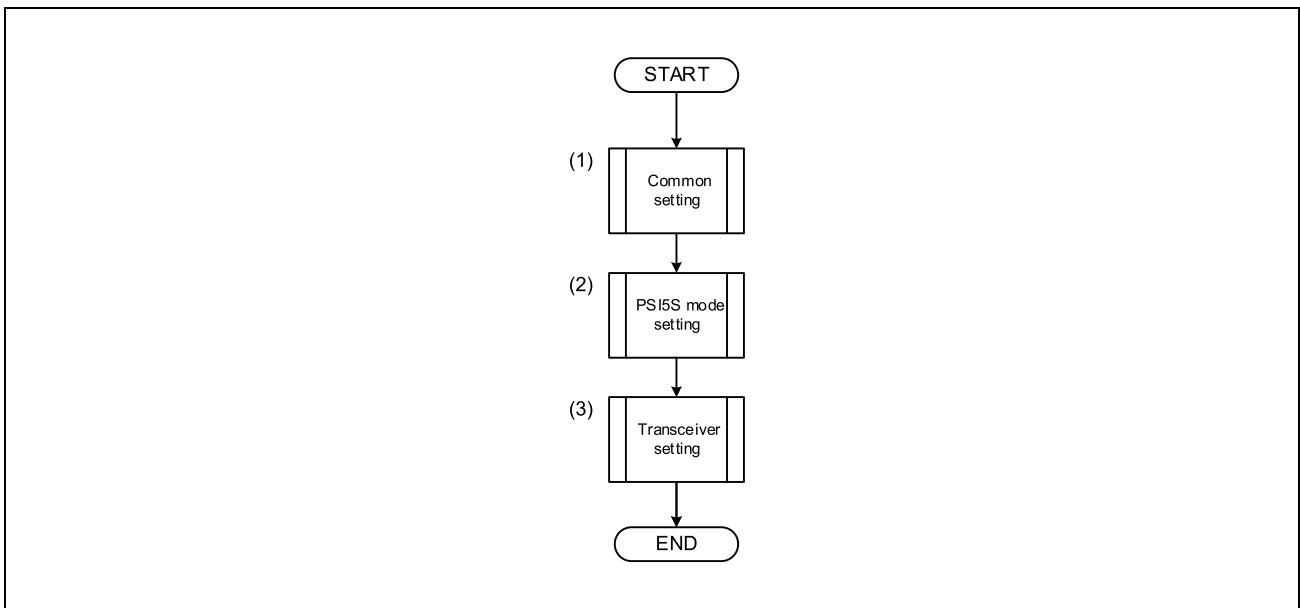


Figure 25.6 PSI5S Mode Setting Procedure

- (1) Specify the common settings. For details, see **Section 25.5.1, Common Setting Procedure**.
- (2) Set up PSI5S mode. For details, see **Section 25.5.2, PSI5S Mode Setting Procedure**.
- (3) Set up the transceiver. For details, see **Section 25.5.4, Procedure for Transmitting Transceiver Commands**.

Figure 25.7 shows the setting procedure to use the PSI5-S in UART mode.

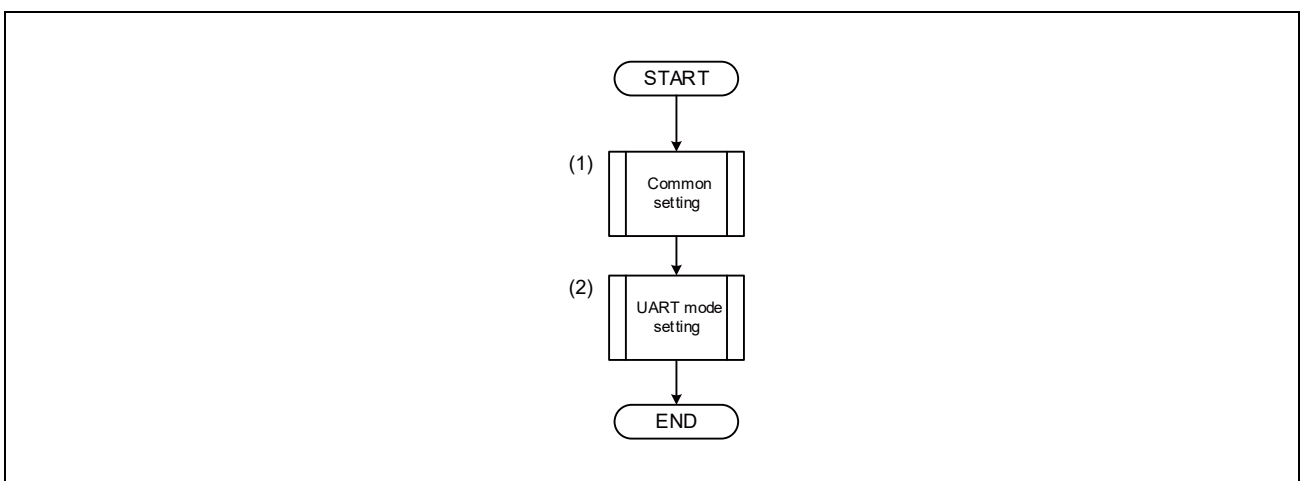


Figure 25.7 UART Mode Setting Procedure

- (1) Specify the common settings. For details, see **Section 25.5.1, Common Setting Procedure**.
- (2) Set up the UART mode. For details, see **Section 25.5.3, UART Mode Setting Procedure**.

25.5.1 Common Setting Procedure

This section describes the common setting procedure to use the PSI5-S in PSI5-S or UART mode.

Perform the procedure shown in **Figure 25.8** below when the PSI5-S is in configuration mode.

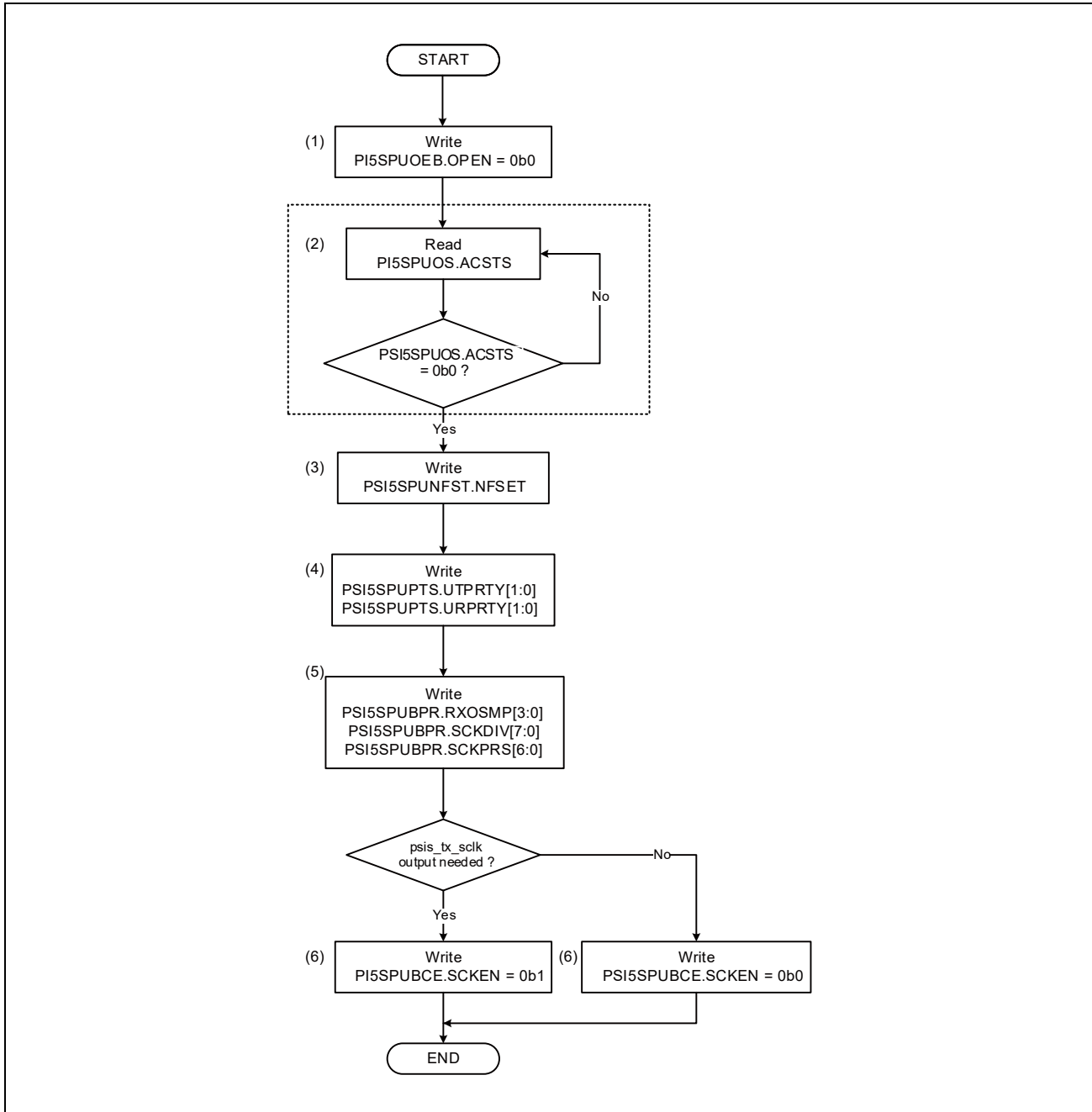


Figure 25.8 Common Setting Procedure

- (1) PSI5SPUOEB: PSI5-S/UART Operation Enable register
Change the operation mode to configuration mode.
- (2) PSI5SPUOS: PSI5-S /UART Operation Status register
Wait until PSI5-S transitions to the configuration mode.

- (3) **PSI5SPUNFST**: PSI5-S/UART Noise Filter Set register
Specify whether to use a noise filter for the signal input to the `psis_rx_data` pin, which is a UART Rx pin.
- (4) **PSI5SPUPTS**: PSI5-S/UART rx/tx Parity Set register
Specify the parity bit settings for both UART reception and UART transmission.
- (5) **PSI5SPUBPR**: PSI5-S/UART Baud rate Parameter register
Specify the baud rate for UART communication. The baud rate is determined by the oversample number, clock division ratio, and division ratio of the prescaler.
The clock division ratio and the division ratio of the prescaler are also used to set the frequency of the sampling clock (`psis_tx_sclk`).

Calculate the baud rate of UART communication using the following formula:

$$\text{Baud rate} = [\text{psis_clk frequency}] * (1/\text{prescaler division ratio}) * (1/\text{clock division ratio}) * (1/\text{oversample number})$$

Example: Prescaler division ratio = 1 (PCKPRS setting = 0)

Clock division ratio = 3 (SCKDIV setting = 2)

Oversample number = 5 (RXOSMP setting = 4)

The calculation result is as follows:

$$\text{Baud rate} = 80\text{MHz} * (1/1) * (1/3) * (1/5) = 5.33 \text{ Mbps}$$

Calculate the frequency of the sampling clock using the following formula:

$$\text{psis_tx_sclk} = [\text{psis_mult_clk frequency}] * (1/2) * (1/\text{prescaler division ratio}) * (1/\text{clock division ratio})$$

Example: Prescaler division ratio = 1 (PCKPRS setting = 0)

Clock division ratio = 3 (SCKDIV setting = 2)

The calculation result is as follows:

$$\text{psis_tx_sclk} = 160\text{MHz} * (1/2) * (1/1) * (1/3) = 26.67 \text{ MHz}$$

If the output of the `psis_tx_sclk` signal is required, specify the following setting:

- (6) **PSI5SPUBCE**: PSI5-S/UART Baud rate Clock Enable register
Specify whether to enable the output of `psis_tx_sclk` clock.

25.5.2 PSI5S Mode Setting Procedure

This section describes the setting procedure to use PSI5-S in PSI5S mode.

Perform the procedure shown in **Figure 25.9** below after performing the procedure described in **Section 25.5.1, Common Setting**.

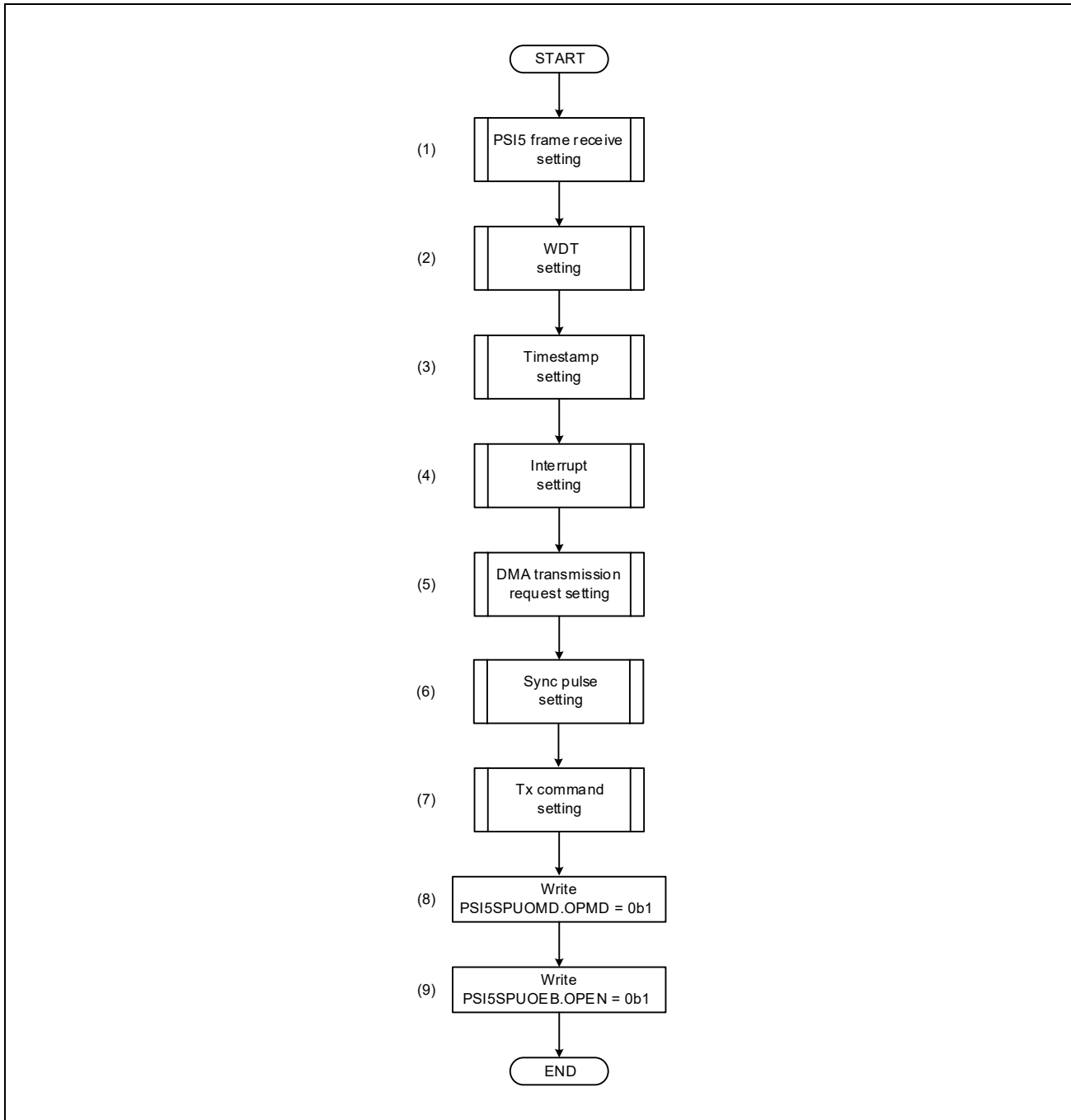


Figure 25.9 PSI5S Mode Setting Procedure

- (1) Set up PSI5 frame reception. For details, see **Section 25.5.2.1, PSI5 Frame Reception Setting**.
- (2) Set up the WDT. For details, see **Section 25.5.2.2, WDT Settin**.
- (3) Set up timestamps. For details, see **Section 25.5.2.3, Timestamp Setting**.

- (4) Set up interrupts. For details, see **Section 25.5.2.4, Interrupt Setting**.
- (5) Set up DMA transfer requests. For details, see **Section 25.5.2.5, DMA Transfer Request Setting**.
- (6) Setup synchronization pulse. For details, see **Section 25.5.2.6, Synchronization Pulse Setting Procedure**.
- (7) Set up Tx command. For details, see **Section 25.5.2.7, Tx Command Setting Procedure**.
- (8) PSI5SPUOMD: PSI5-S/UART Operation Mode register
Set the operation mode to PSI5S mode.
- (9) PSI5SPUOEB: PSI5-S/UART Operation Enable register
Start the operation of PSI5-S.

25.5.2.1 PSI5 Frame Reception Setting

This section describes how to set up PSI5 frame reception.

Perform the procedure shown in **Figure 25.10** below for channel 0 and the required channels among channels 1 to 7.

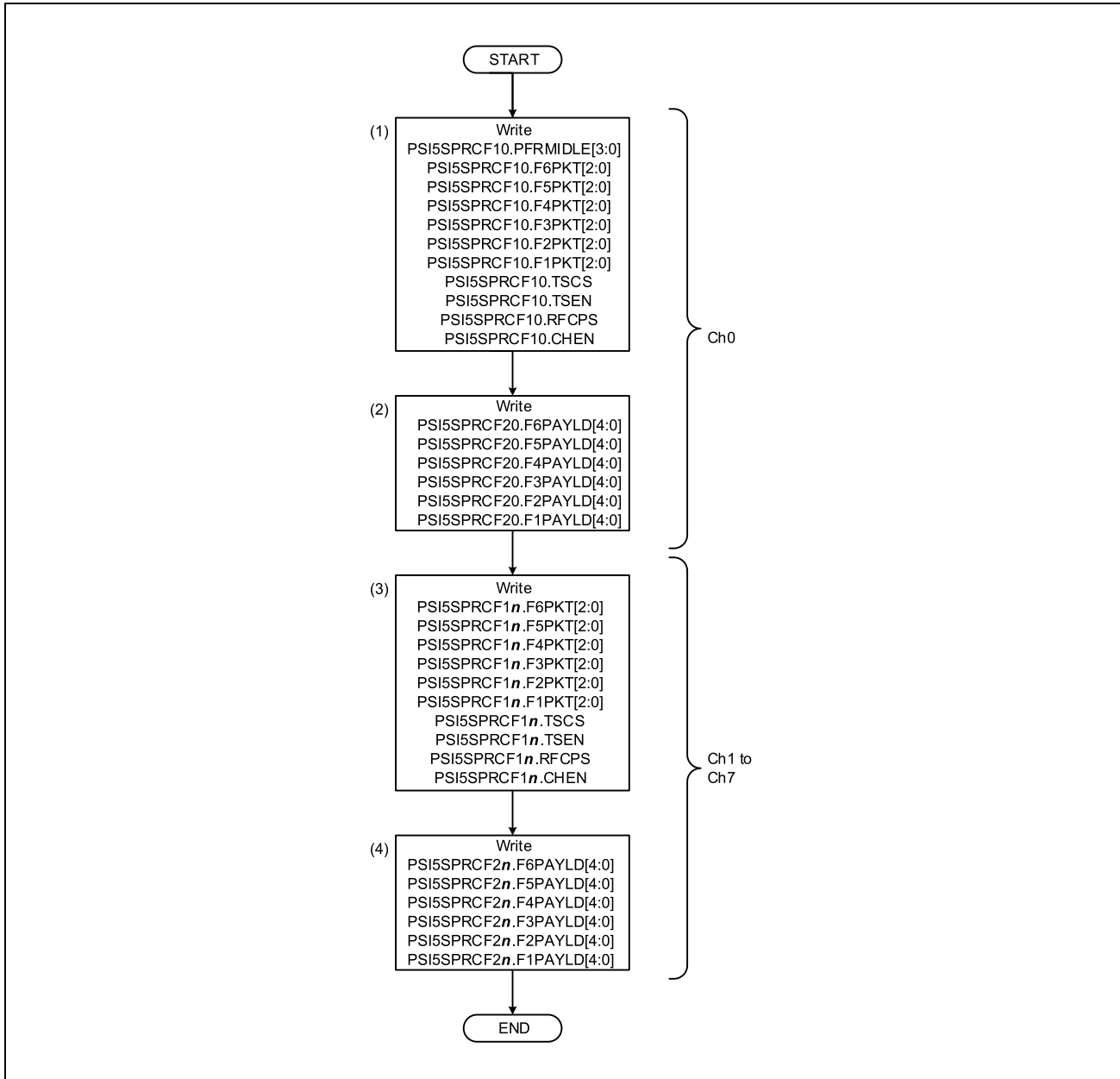


Figure 25.10 PSI5 Frame Reception Setting Procedure

- (1) PSI5SPRCF10: PSI5-S Receive Config1 ch0 register
Specify the desired settings in the PSI5 frame reception configuration 1 register for channel 0.
- (2) PSI5SPRCF20: PSI5-S Receive Config2 ch0 register
Specify the desired settings in the PSI5 frame reception configuration 2 register for channel 0.
- (3) PSI5SPRCF1n: PSI5-S Receive Config1 chn register ($n = 1$ to 7)
Specify the desired settings in the PSI5 frame reception configuration 1 register for channel n .

(4) PSI5SPRCF2n: PSI5-S Receive Config2 chn register ($n = 1$ to 7)

Specify the desired settings in the PSI5 frame reception configuration 2 register for channel n .

The value of the PSI5SPRCF1n.FmPKT[2:0] bits is determined by the relationship between the values of the PSI5SPRCF1n.RFCPS and PSI5SPRCF2n.FmPAYLD[4:0] bits as shown in **Table 25.88**. Note that if a value other than those shown in the table is set to the PSI5SPRCF1n.FmPKT[2:0] bits, the operation of PSI5-S will be unpredictable. ($n = 1$ to 7, $m = 1$ to 6)

Table 25.88 List of PRCF1n.FmPKT[2:0] Settings

PSI5SPRCF2n.FmPAYLD[4:0]	PSI5SPRCF1n.FmPKT[2:0]	
	PSI5SPRCF1n.RFCPS = 0 (Checksum of Rx frame is parity)	PSI5SPRCF1n.RFCPS = 1 (Checksum of Rx frame is CRC)
8	3 _H	4 _H
9	3 _H	4 _H
10	4 _H	4 _H
11	4 _H	4 _H
12	4 _H	4 _H
13	4 _H	4 _H
14	4 _H	4 _H
15	4 _H	4 _H
16	4 _H	5 _H
17	4 _H	5 _H
18	5 _H	5 _H
19	5 _H	5 _H
20	5 _H	5 _H
21	5 _H	5 _H
22	5 _H	5 _H
23	5 _H	5 _H
24	5 _H	6 _H
25	5 _H	6 _H
26	6 _H	6 _H
27	6 _H	6 _H
28	6 _H	6 _H

Note: $n = 1$ to 7, $m = 1$ to 6

(1) PSI5 Frame Reception Setting (No Payload Checksum in Reception Frame)

IF there is no Payload Checksum in a reply frame from a transceiver (excluding the Payload bit number of 10, 18 and 24 bits), the Rx Frame checksum CRC/Parity Select bit (PSI5SPRCF1n.RFCPS) is set as Parity (=0) and the payload parity error is ignored. PSI5-S recognizes the stuffing bit as a parity bit.

For the procedure of the frame reception from a transceiver, see **Section 25.5.6.3, Response Frame from Transceiver Reception**.

25.5.2.2 WDT Setting

This section describes how to set up the WDT.

Perform the procedure shown in **Figure 25.11** below for the channels that require the WDT.

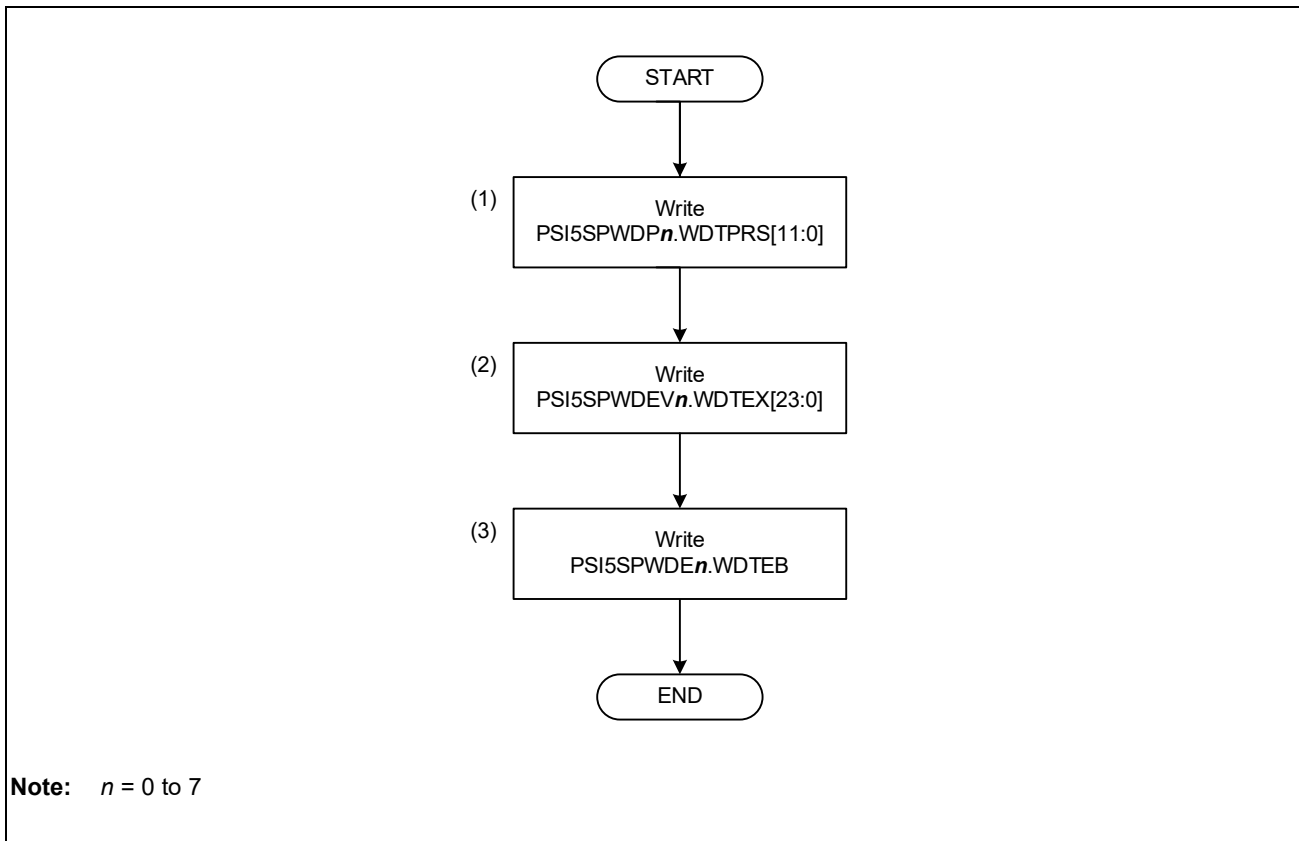


Figure 25.11 WDT Setting Procedure

- (1) PSI5SPWDP n : PSI5-S WDT Prescaler chn register
Specify the WDT prescaler settings for channel n . (n : 0 to 7)
- (2) PSI5SPWDEV n : PSI5-S WDT Expiration Value chn register
Specify the expiration value of the WDT for channel n . (n : 0 to 7)
- (3) PSI5PWDE n : PSI5-S WDT Enable chn register
Enable the WDT for channel n . (n : 0 to 7)
The WDT can be enabled when PSI5-S is in either the PSI5-S_active or configuration state.

25.5.2.3 Timestamp Setting

This section describes how to set up timestamp counters A and B.

Perform the procedure shown in **Figure 25.12** below for the timestamps to be used. **Figure 25.12** shows the setting procedure to control timestamp counters A and B separately.

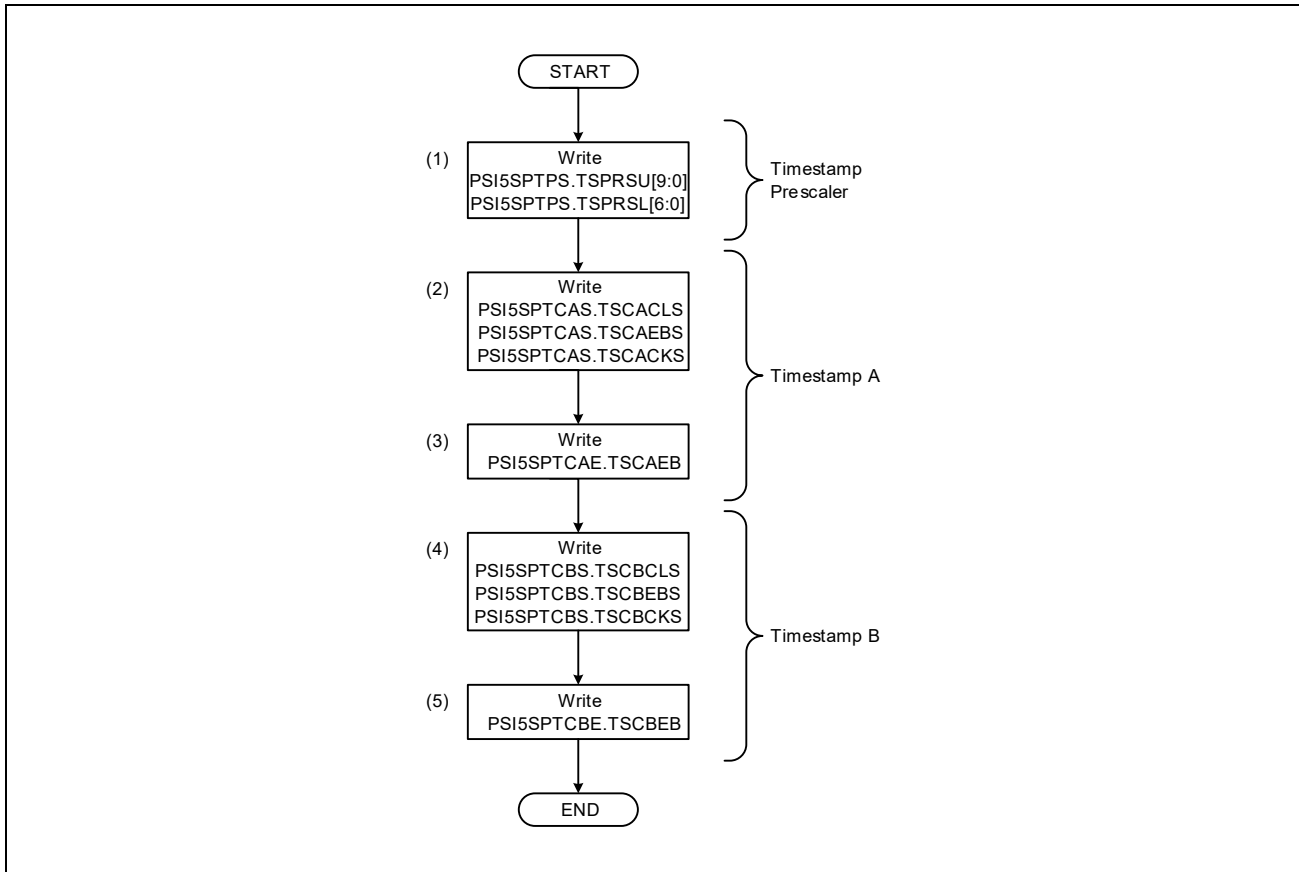


Figure 25.12 Timestamp Setting Procedure (for Separate Control)

- (1) PSI5SPTPS: PSI5-S Timestamp Prescaler register
Specify the settings of the timestamp prescaler.
- (2) PSI5SPTCAS: PSI5-S Timestamp Counter A Select register
Select the clear signal, enable signal, and clock signal for timestamp counter A.
- (3) PSI5SPTCAE: PSI5-S Timestamp Counter A Enable register
Enable the operation of timestamp counter A. By this setting, timestamp counter A starts.
- (4) PSI5SPTCBS: PSI5-S Timestamp Counter B Select register
Select the clear signal, enable signal, and clock signal for timestamp counter B.
- (5) PSI5SPTCBE: PSI5-S Timestamp Counter B Enable register
Enable the operation of timestamp counter B.

Figure 25.13 below shows a different setting procedure than the above to control timestamp counters A and B together.

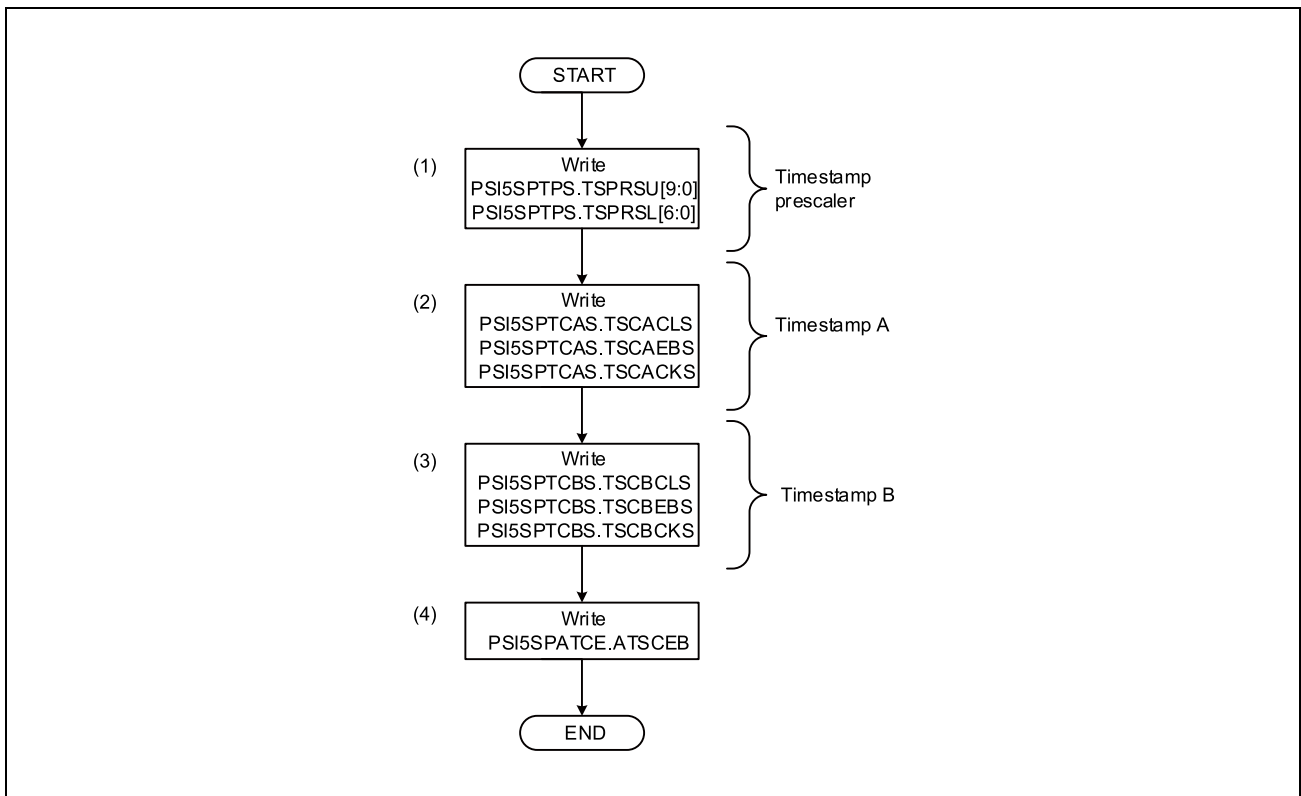


Figure 25.13 Timestamp Setting Procedure (for Batch Control)

- (1) PSI5SPTPS: PSI5-S Timestamp Prescaler register
Specify the settings of the timestamp prescaler.
- (2) PSI5SPTCAS: PSI5-S Timestamp Counter A Select register
Select the clear signal, enable signal, and clock signal for timestamp counter A.
- (3) PSI5SPTCBS: PSI5-S Timestamp Counter B Select register
Select the clear signal, enable signal, and clock signal for timestamp counter B.
- (4) PSI5SPATCE: PSI5-S All Timestamp Counter Enable register
Enable the operation of timestamp counters A and B together. By this setting, timestamp counters A and B start.

Note that clear instructions can also be issued for timestamp counters A and B separately or in a batched manner.

25.5.2.4 Interrupt Setting

This section describes how to set up interrupts.

Perform the procedure shown in **Figure 25.14** below for the required channels among channels 0 to 7.

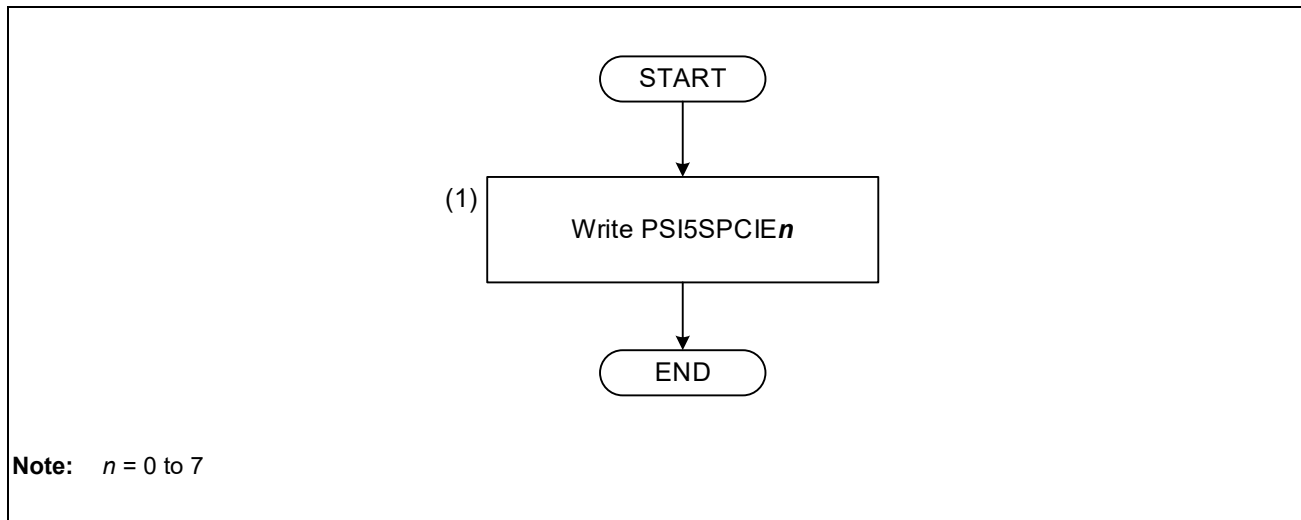


Figure 25.14 Interrupt Setting Procedure

- (1) PSI5SPCIE n : PSI5-S CPU Interrupt Enable chn register (n : 0 to 7)
Specify the settings of the interrupt enable bits according to the interrupt factors.

Table 25.89 below lists the enable bits in the interrupt enable register.

Table 25.89 Interrupt Enable Register Bits

Register Symbol	Bit Symbol	Factor
PSI5SPCIE0	IEBXCRC	XCRC error
PSI5SPCIE(0 to 7)	IEBCRC	CRC error
PSI5SPCIE(0 to 7)	IEBPT	Parity error
PSI5SPCIE(0 to 7)	IEBTRST	Transceiver status error
PSI5SPCIE0	IEBUTPT	UART Rx parity error
PSI5SPCIE0	IEBUTFR	UART Rx framing error
PSI5SPCIE(0 to 7)	IEBRWDT	WDT error
PSI5SPCIE(0 to 7)	IEBROV	PSI5 frame reception overrun error
PSI5SPCIE(0 to 7)	IEBRFLK	PSI5 frame lack error
PSI5SPCIE(0 to 7)	IEBRFEX	PSI5 frame excess error
PSI5SPCIE(0 to 7)	IEBRFN	PSI5 frame reception finish
PSI5SPCIE(1 to 7)	IEBDDSOW	DDSR overwrite error
PSI5SPCIE(1 to 7)	IEBDDSFN	DDSR transmit finish
PSI5SPCIE0	IEBCTFN	Command data transmit finish

25.5.2.5 DMA Transfer Request Setting

This section describes how to set up DMA transfer requests.

Perform the procedure shown in **Figure 25.15** below for the required channels among channels 0 to 7.

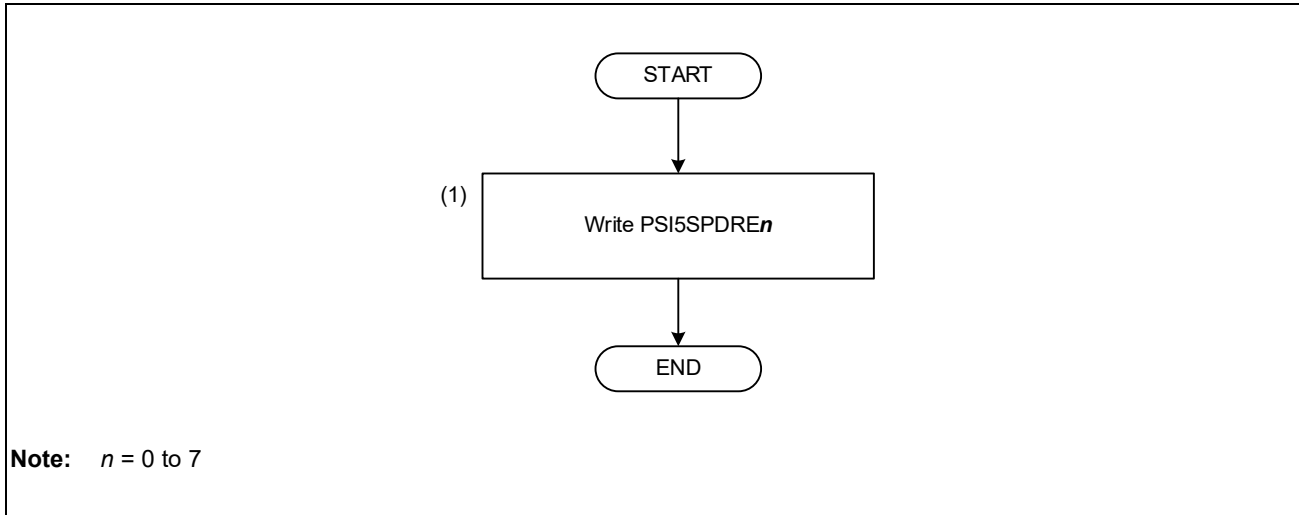


Figure 25.15 DMA Transfer Request Enable Setting Procedure

When enabling the output of DMA transfer requests, check the contents of the following status register:

- (1) PSI5SPDRE n : PSI5-S DMA transfer Request Enable ch n register (n : 0 to 7)
Enable the output of DMA transfer requests.

Table 25.90 below lists the enable bits in the transfer request enable register.

Table 25.90 DMA Transfer Request Enable Register Bits

Register Symbol	Bit Symbol	Factor
PSI5SPDRE(0 to 7)	DRQERFN	PSI5 last frame reception finish
PSI5SPDRE(0 to 7)	DRQEWDT	WDT error
PSI5SPDRE(1 to 7)	DRQETFN	DDSR transmit finish

25.5.2.6 Synchronization Pulse Setting Procedure

This section describes how to set up a synchronization pulse.

Perform the procedure shown in **Figure 25.16** below for the channels that require a synchronization pulse.

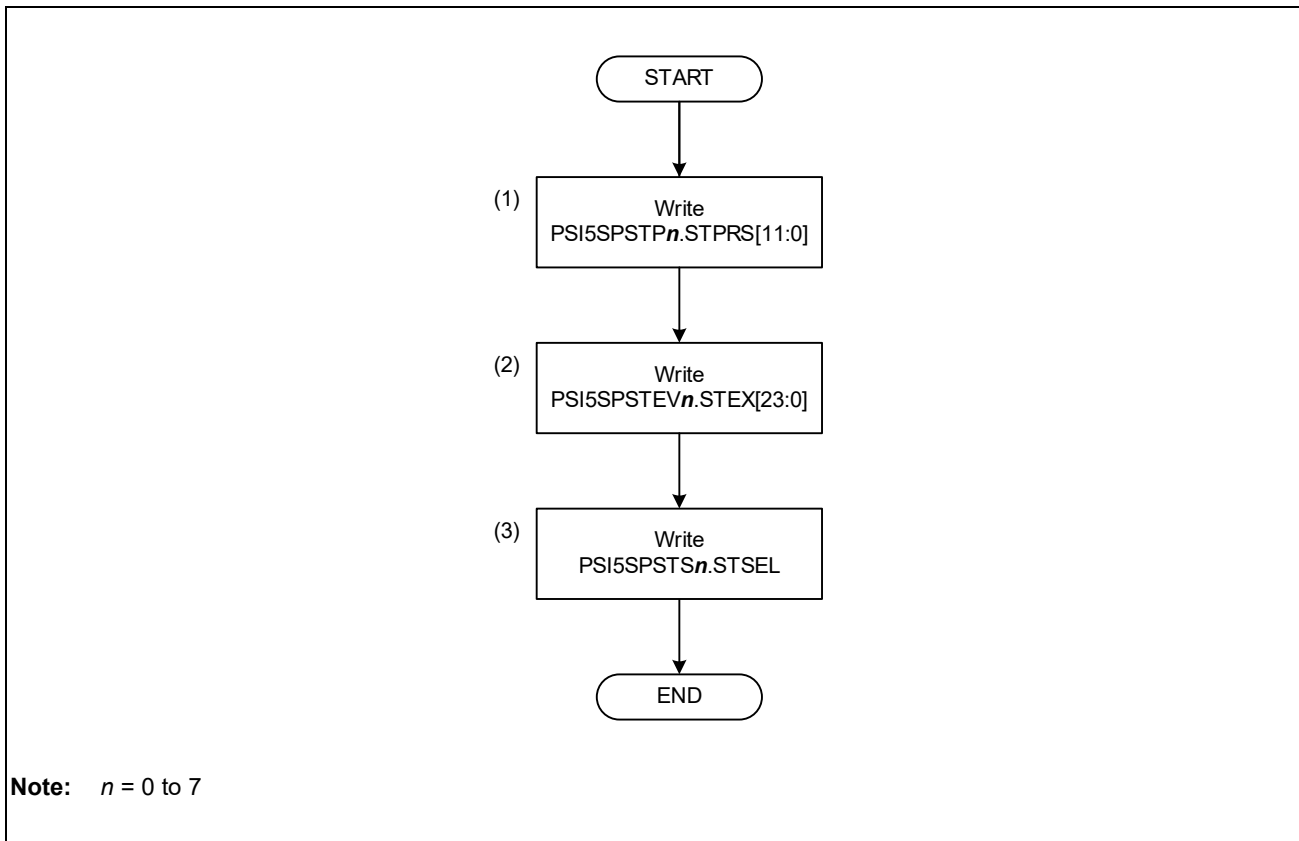


Figure 25.16 Synchronization Pulse Setting Procedure

- (1) PSI5SPSTP n : PSI5-S Sync Trigger Prescaler chn register.
Specify settings of the sync trigger prescaler for channel n . (n : 1 to 7)
- (2) PSI5SPSTEV n : PSI5-S Sync Trigger Expiration Value chn register.
Specify settings of the sync trigger expiration value for channel n . (n : 1 to 7)
- (3) PSI5SPSTS n : PSI5-S Sync Trigger Select chn register.
Specify settings of the selection of synchronous trigger for channel n . (n : 1 to 7)

25.5.2.7 Tx Command Setting Procedure

This section describes how to set up the Tx command data for ECU-to-sensor communication.

Perform the procedure shown in **Figure 25.17** below for the Tx command of each channels that requires a synchronization pulse.

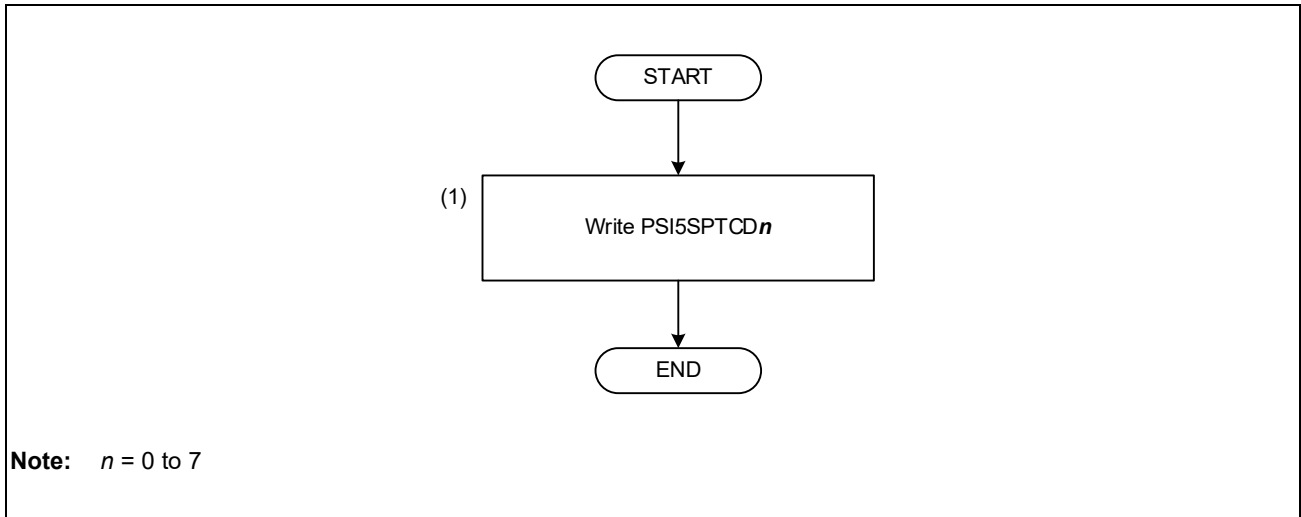


Figure 25.17 Tx Command Setting Procedure

- (1) PSI5SPTCD n : PSI5-S Tx command Data ch n register.
Specify the settings of the Tx command for channel n . (n : 1 to 7)

25.5.3 UART Mode Setting Procedure

PSI5-S has a UART mode in which PSI5-S can operate as a UART with limited functions.

For details on the UART specifications, see **Table 25.8, Features**.

Perform the procedure shown in **Figure 25.18** below after performing the procedure described in **Section 25.5.1, Common Setting Procedure**.

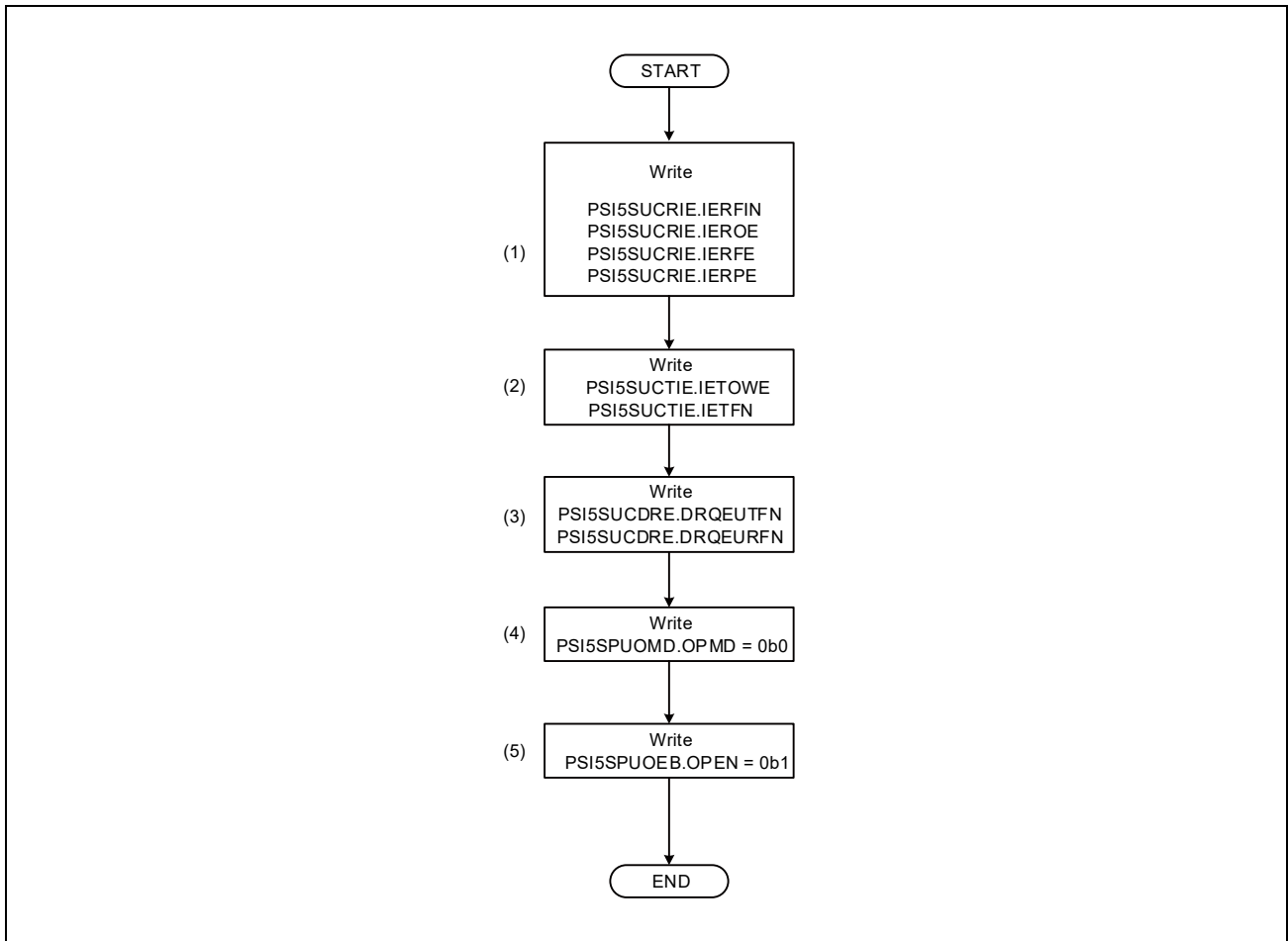


Figure 25.18 UART Mode Setting Procedure

- (1) PSI5SUCRIE: UART Communication Rx Interrupt Enable register
Specify settings of the interrupts at the end of UART frame reception, at the occurrence of a UART overrun error, at the occurrence of a UART framing error and at the occurrence of a UART parity error.
- (2) PSI5SUCTIE: UART Communication Tx Interrupt Enable register
Specify settings of the interrupts at the end of UART frame transmission and at the occurrence of a UART overwrite error.
- (3) PSI5SUCDRE: UART Communication DMA Request Enable register
Specify settings of the DMA transfer requests at the end of UART reception and at the end of UART transmission.
- (4) PSI5SPUOMD: PSI5-S/UART Operation Mode register
Set the operation mode to UART mode.

- (5) PSI5SPUOEB: PSI5-S/UART Operation Enable register
Start the operation of PSI5-S.

25.5.4 Procedure for Transmitting Transceiver Commands

This section describes how to transmit commands for an externally connected PSI5 transceiver.

Perform this setting procedure after performing the procedure described in **Section 25.5.2, PSI5S Mode Setting Procedure**.

PSI5 frame reception setting of Ch0 is required, because the response frame from the transceiver is stored in Ch0 MB.

(See **Section 25.5.2.1, PSI5 Frame Reception Setting**)

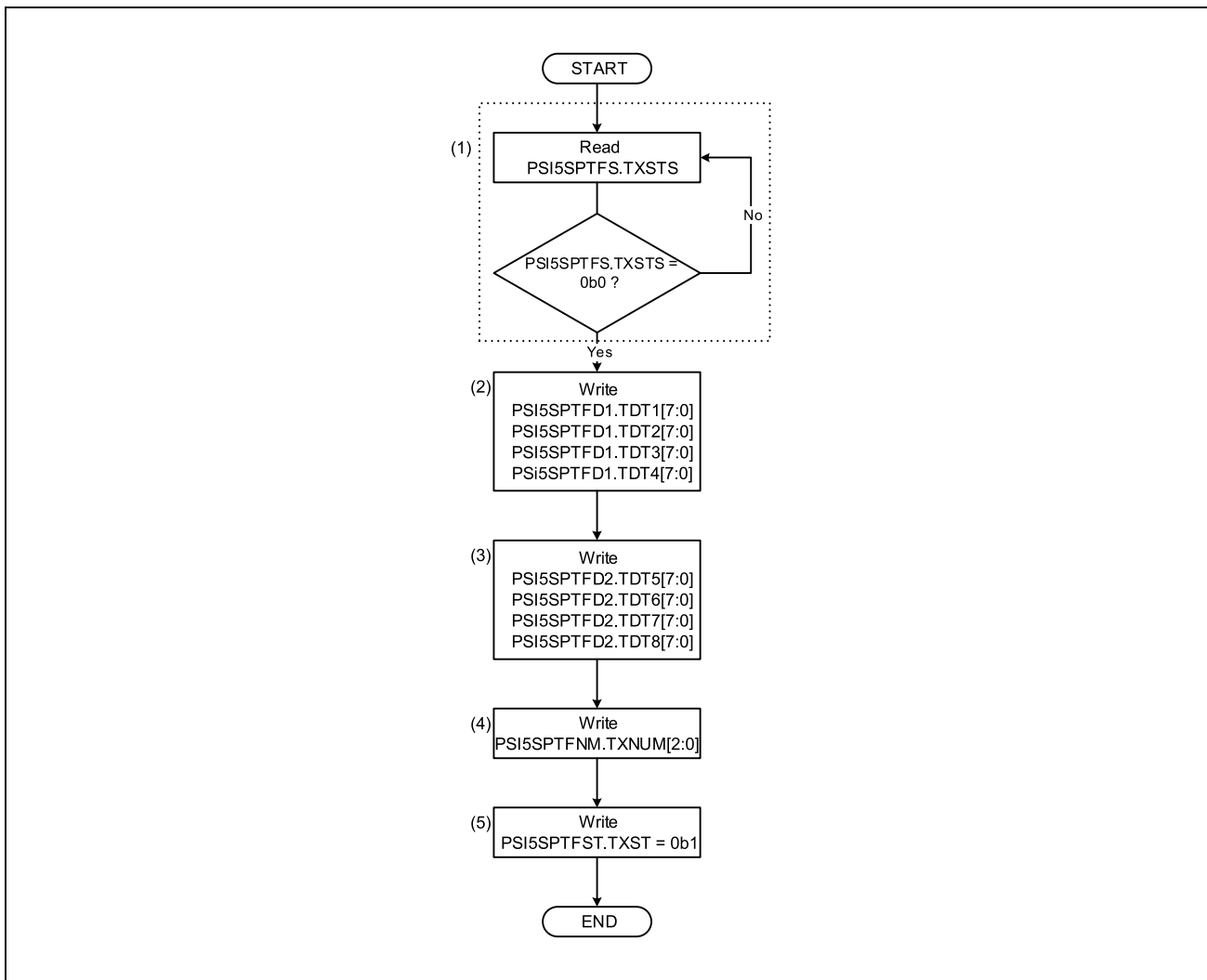


Figure 25.19 Transceiver Setting Procedure

- (1) PSI5SPTFS: PSI5-S Tx Frame Status register
Read the TXSTS bit to check that the UART is not busy with transmission (TXSTS = 0). If the UART is busy (TXSTS = 1), wait until it is not busy.
- (2) PSI5SPTFD1: PSI5-S Tx Frame Data1 register
Sequentially set the 1st to 4th bytes of the data to be transmitted continuously in this register. If the continuously transmitted data is less than 4 bytes, set only the data to be transmitted.
- (3) PSI5SPTFD2: PSI5-S Tx Frame Data2 register
Sequentially set the 5th to 8th bytes of the data to be transmitted continuously in this register. If the continuously transmitted data is less than 9 bytes, set only the data to be transmitted.

- (4) PSI5SPTFNM: PSI5-S Tx Frame Number register
Set the number of bytes to be transmitted continuously in this register.
- (5) PSI5SPTFST: PSI5-S Tx Frame Start register
Start continuous transmission.

As an example, **Table 25.91** below describes the settings of PSI5SPTFD1, PSI5SPTFD2, and PSI5SPTFNM registers to set up the external transceiver that uses the command format shown below in **Figure 25.20**.

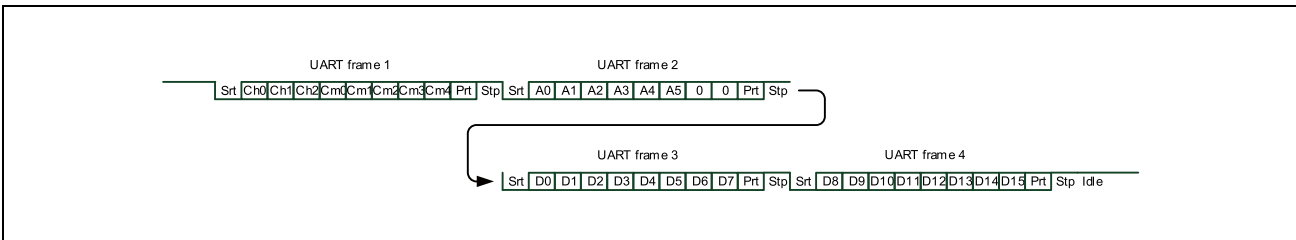


Figure 25.20 Example of Command Format

Table 25.91 List of Register Settings

Register Symbol	Bit Symbol	Value
PSI5SPTFD1	TDT1[7:0]	{Cm[4:0],Ch[2:0]}
PSI5SPTFD1	TDT2[7:0]	{0b00,A[5:0]}
PSI5SPTFD1	TDT3[7:0]	D[7:0]
PSI5SPTFD1	TDT4[7:0]	D[15:8]
PSI5SPTFD2	TDT5[7:0]	N/A
PSI5SPTFD2	TDT6[7:0]	N/A
PSI5SPTFD2	TDT7[7:0]	N/A
PSI5SPTFD2	TDT8[7:0]	N/A
PSI5SPTFNM	TXNUM[2:0]	0x3

25.5.5 DDSR Transmission Procedure

This section describes how to perform DDSR transmission.

Perform the procedure shown in **Figure 25.21** below for channels 1 to 7.

This procedure must be performed when the operation mode is PSI5S mode.

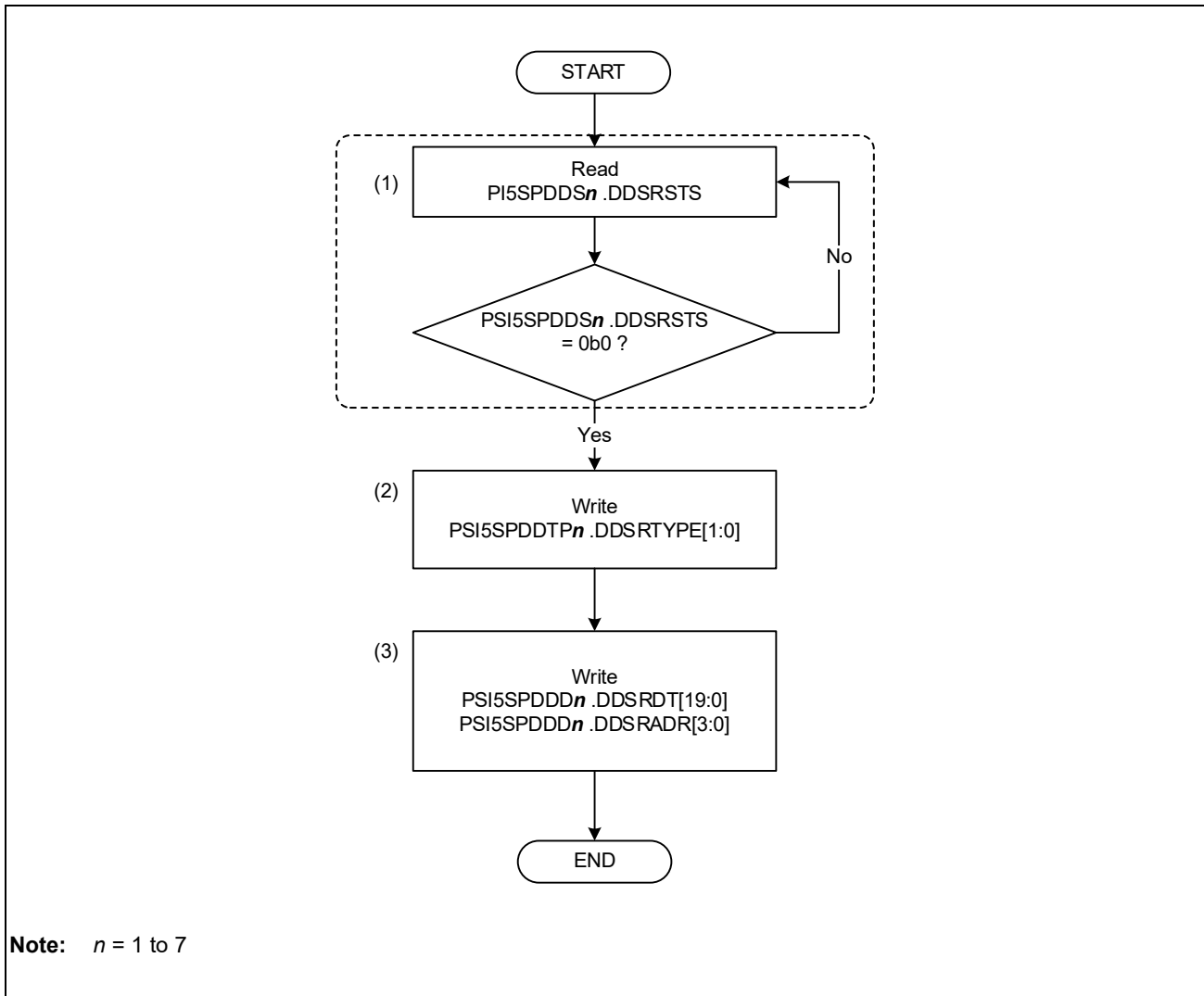


Figure 25.21 DDSR Transmission Procedure

- (1) PSI5PDDSn.DDSRSTS: PSI5-S DDSR Status Chn register ($n: 1$ to 7)
Check the DDSR transmission status. When the DDSRSTS bit is 1, wait until the value changes to 0.
Note that if DDSR transmit data is written to the PSI5PDDDDn register when the DDSRSTS bit is 1, an overwrite error occurs. For details, see **Section 25.6.2.9, Abnormal Transmission (Transmission Error)**.
- (2) PSI5PDDTPn: PSI5-S DDSR Type Chn register ($n: 1$ to 7)
Specify the type of ECU-to-sensor data.
- (3) PSI5PDDDDn: PSI5-S DDSR Data Chn register ($n: 1$ to 7)
Set DDSR transmit data in this register.

Figure 25.22 shows the procedure of DDSR overwrite error processing.

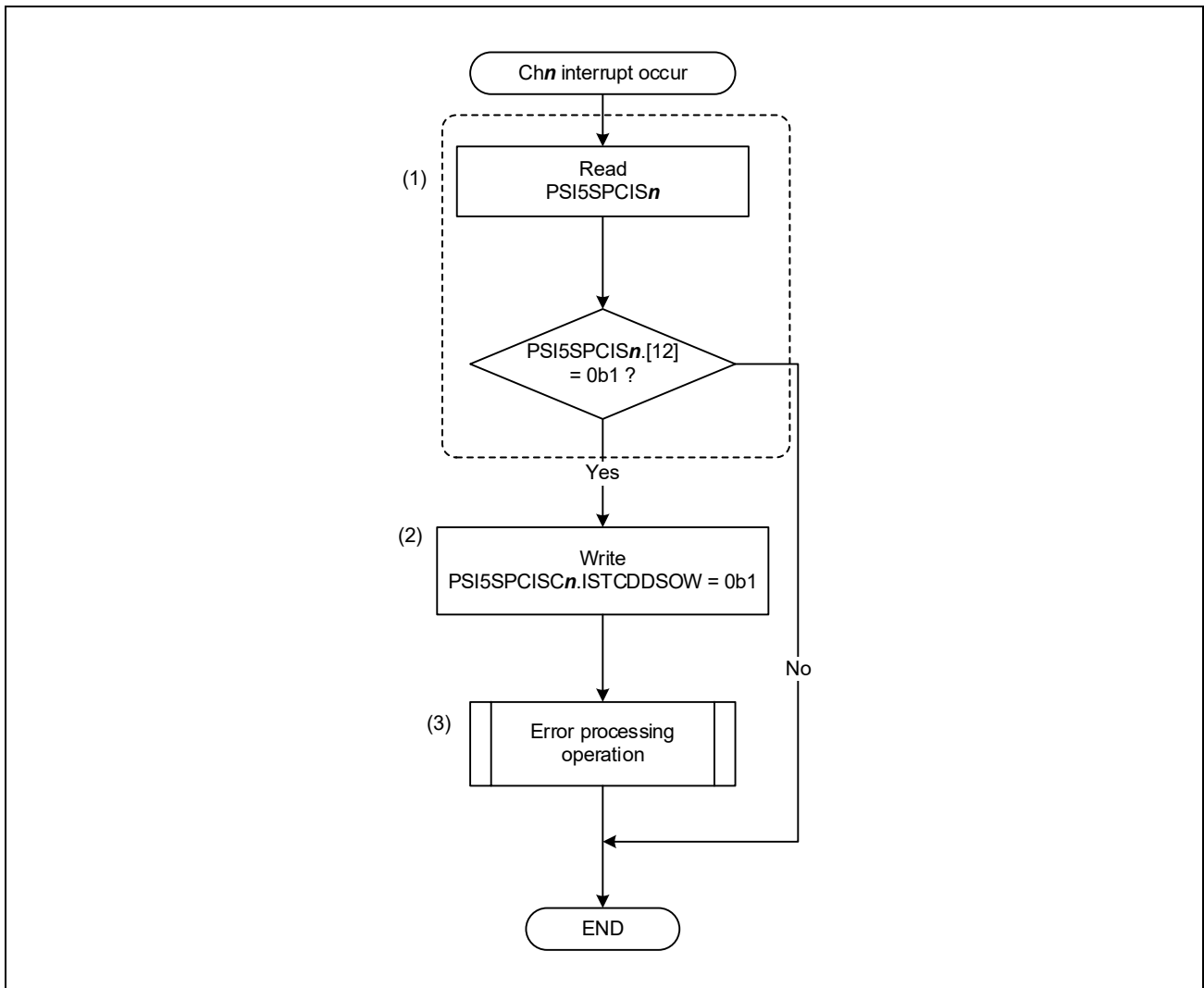


Figure 25.22 DDSR Overwrite Error Processing Procedure

- (1) PCISn: PSI5-S CPU Interrupt Status chn register. (n: 0 to 7)
Read PSI5SPCISn to check whether the DDSR overwrite error is set (PSI5SPCISn = 1). If the DDSR overwrite error is not set (PSI5SPCISn = 0), finish the processing.
- (2) PSI5SPCISn: PSI5-S CPU Interrupt Status chn register. (n: 0 to 7)
Clear the DDSR overwrite error.
- (3) Perform the processing to be done when a DDSR overwrite error occurs.

25.5.6 PSI5 Frame Reception Procedures

25.5.6.1 Normal Reception

This section describes the procedure of PSI5 frame reception.

PSI5-S can receive PSI5 frames only in PSI5S mode.

Figure 25.23 shows the operations to normally receive PSI5 frames.

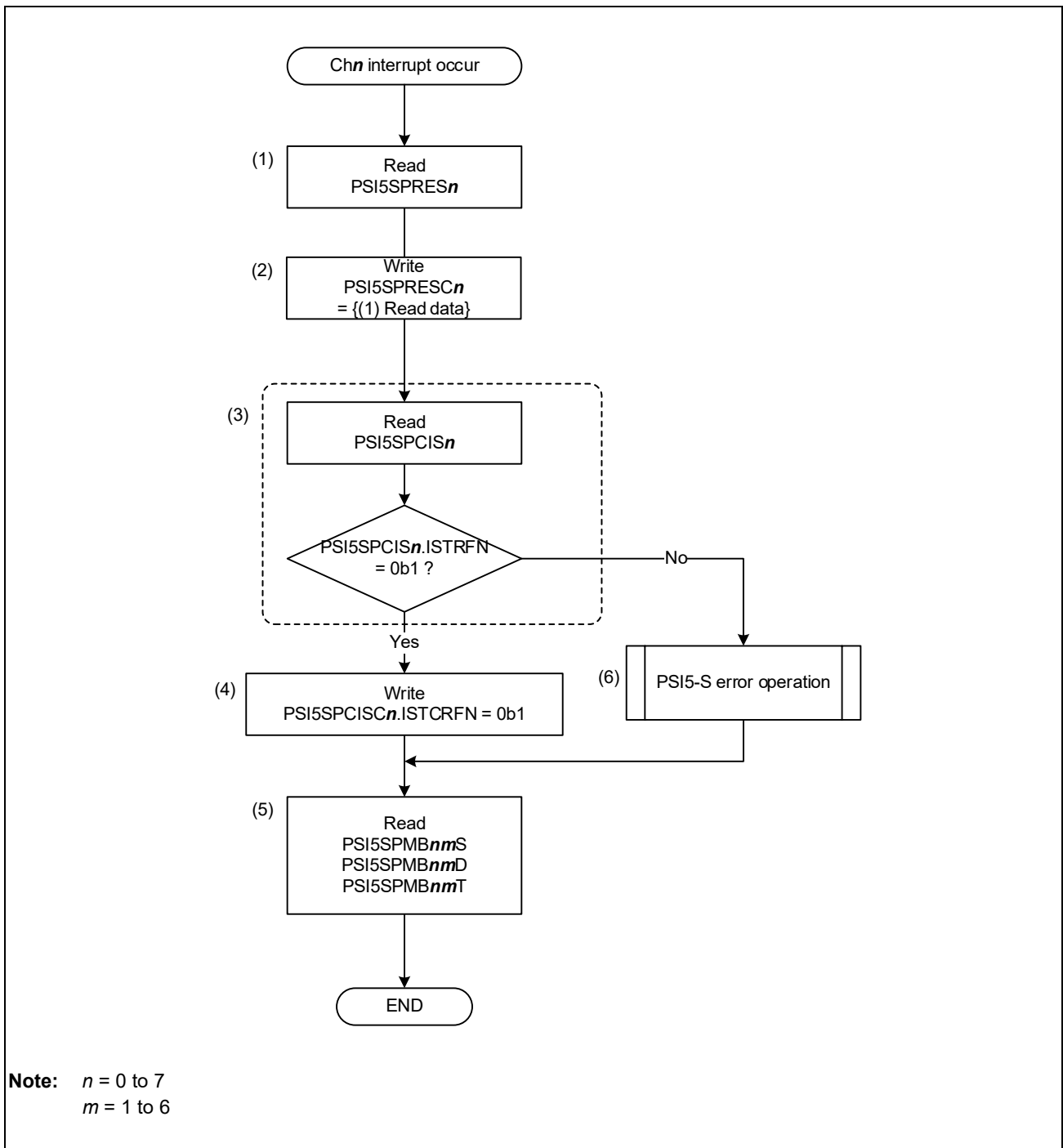


Figure 25.23 Procedure of Normal Reception of PSI5 Frames

- (1) **PSI5SPRES n** : PSI5-S Receive Error Status chn register (n : 0 to 7)
Check the frame that caused the interrupt. (This step is optional.)
- (2) **PSI5SPRESC n** : PSI5-S Receive Error Status Clear chn register (n : 0 to 7)
Clear the flag indicating the status of the frame that caused the interrupt. (This operation is optional.)
- (3) **PSI5SPCIS n** : PSI5-S CPU Interrupt Status chn register (n : 0 to 7)
Read the PSI5-S interrupt status register to check that the ISTRFN bit is set.
- (4) **PSI5SPCISC n** : PSI5-S CPU Interrupt Status chn register (n : 0 to 7)
Clear the flag indicating the end of frame reception.
- (5) **PSI5SPMB nm S**: PSI5-S receive MailBox chn $frmm$ Status register (n : 0 to 7, m : 1 to 6)
PSI5SPMB nm D: PSI5-S receive MailBox chn $frmm$ Data register (n : 0 to 7, m : 1 to 6)
PSI5SPMB nm T: PSI5-S receive MailBox chn $frmm$ Timestamp register (n : 0 to 7, m : 1 to 6)
Read the PSI5 frame status, data, and timestamp stored in the mailbox.
- (6) If the ISTRFN bit is not set, perform error processing.

25.5.6.2 Abnormal Reception

This section describes how to process an error during PSI5 frame reception.

For the processes from the occurrence of an interrupt to the reading of the interrupt status, see processes (1) and (2) in **Figure 25.24**.

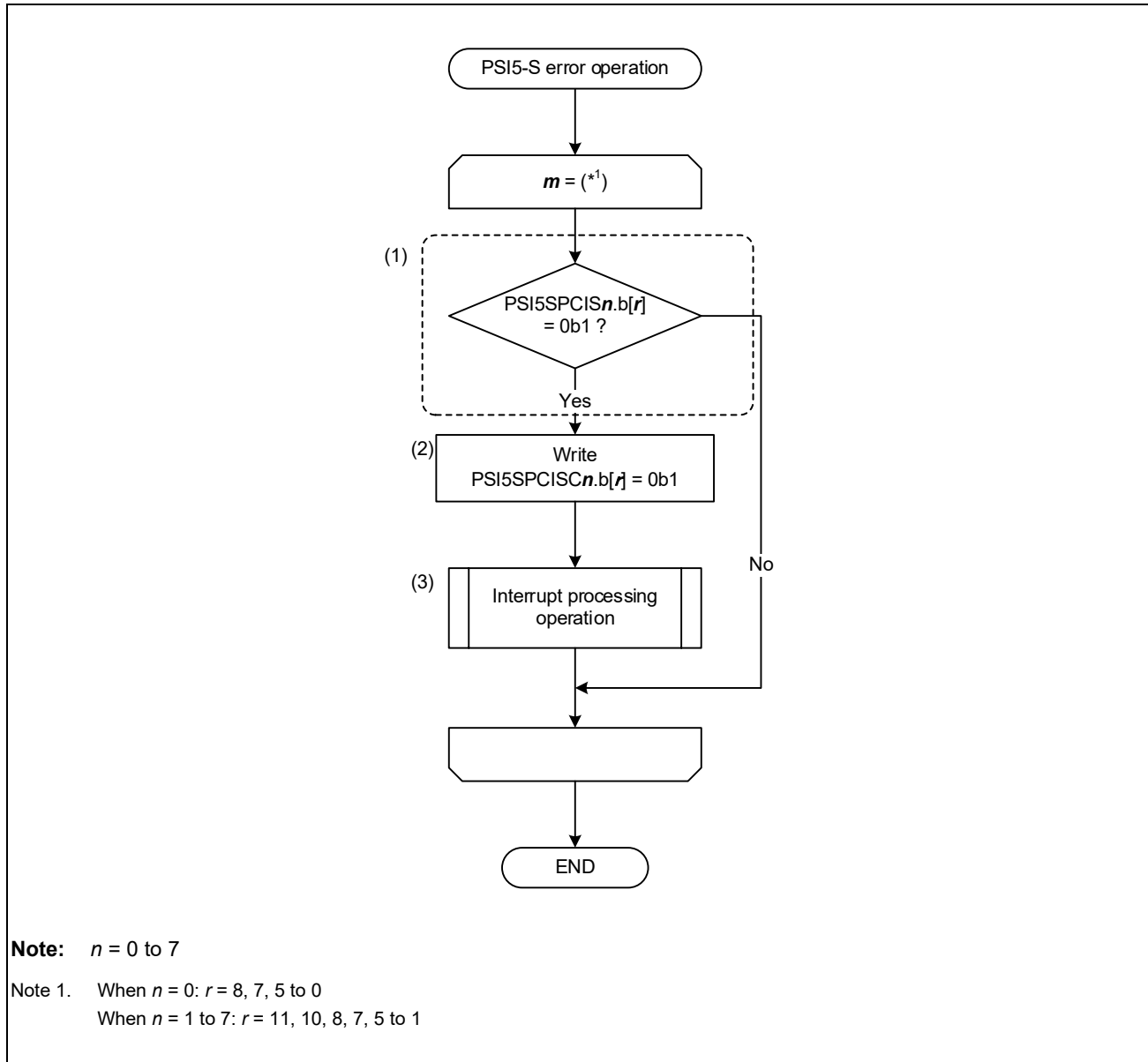


Figure 25.24 Procedure of Error Processing in PSI5 Frame Reception

- (1) Check the cause of the error indicated in the PSI5-S interrupt status register.
- (2) PSI5SPCISC n : PSI5-S CPU Interrupt Status ch n register (n : 0 to 7)
If an error-cause flag is set, clear the flag.
- (3) Perform the error processing that corresponds to the indicated cause of the error.

Figure 25.72 shows the occurrence timing of errors other than WDT errors during PSI5 frame reception.

For the occurrence timing of WDT errors, see **Section 25.6.2.8, Abnormal Reception (WDT Error)**.

25.5.6.3 Response Frame from Transceiver Reception

This section describes the procedure of the response frame from the transceiver reception.

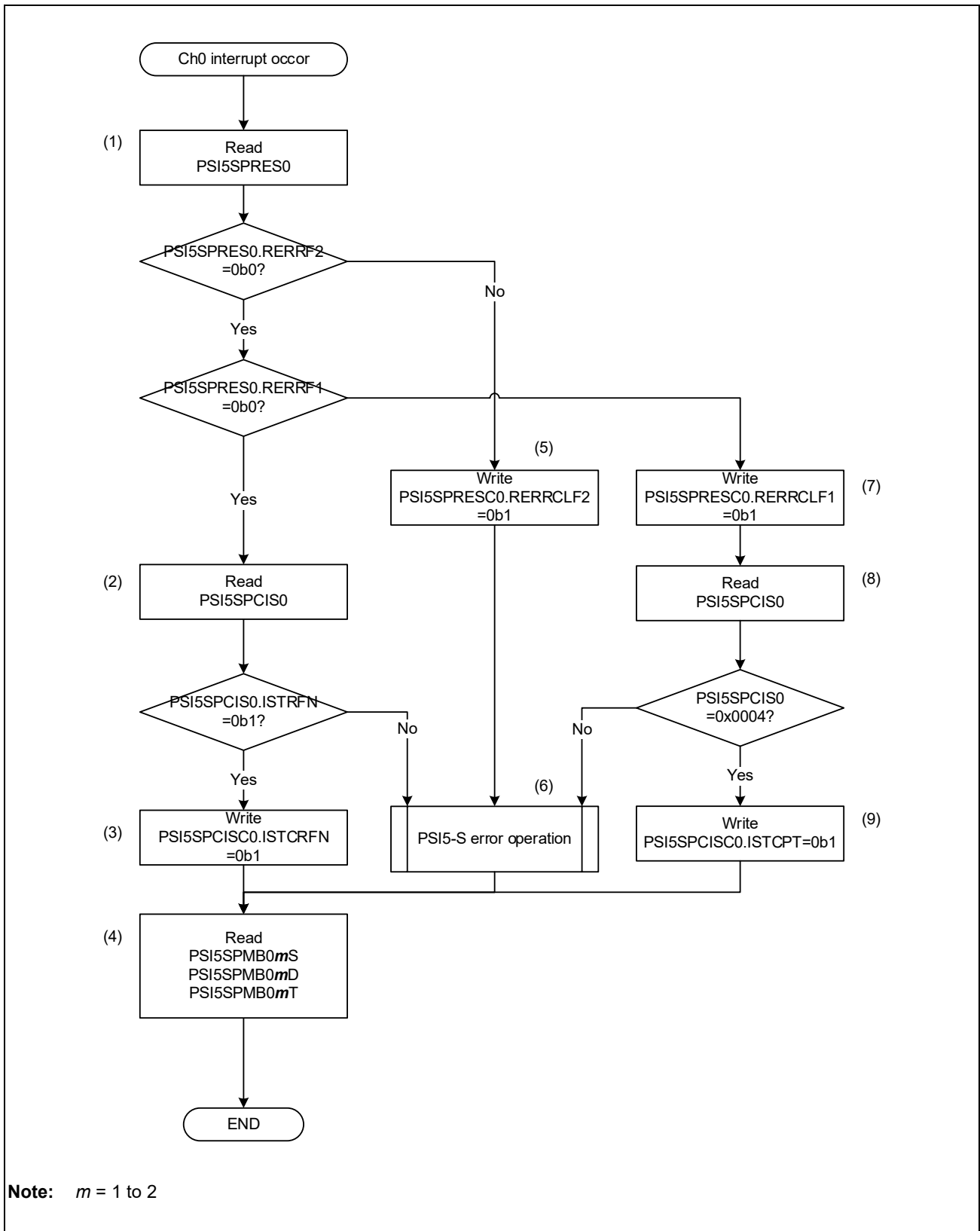


Figure 25.25 Procedure of Response Frame from Transceiver Reception

- (1) PSI5SPRES0: PSI5-S Receive Error Status ch0 register.
Check the error frame that caused the interrupt.
- (2) PSI5SPCIS0: PSI5-S CPU Interrupt Status ch0 register.
Read the PSI5-S interrupt status register to check that the ISTRFN bit is set.
- (3) PSI5SPCISC0: PSI5-S CPU Interrupt Status ch0 register.
Clear the flag indicating the end of frame reception.
- (4) PSI5SPMB0mS: PSI5-S receive MailBox ch0 frmm Status register (*m*: 1 to 2)
PSI5SPMB0mD: PSI5-S receive MailBox ch0 frmm Data register (*m*: 1 to 2)
PSI5SPMB0mT: PSI5-S receive MailBox ch0 frmm Timestamp register (*m*: 1 to 2)
Read the PSI5 frame status, data, and timestamp stored in the mailbox.
- (5) PSI5SPRESC0: PSI5-S Receive Error Status Clear ch0 register.
Clear the flag indicating the error of frame 2.
- (6) When the following cases, perform error processing.
 - The flag indicating the end of reception is not set.
 - The error frame occurs in frame 2 of ch0.
 - The error frame occurs in frame 1 of ch0, and it has been set to other than a payload data parity error.
- (7) PSI5SPRESC0: PSI5-S Receive Error Status Clear ch0 register.
Clear the flag indicating the error of frame 1.
- (8) PSI5SPCIS0: PSI5-S CPU Interrupt Status ch0 register.
Read the PSI5-S CPU Interrupt Status ch0 register to check that none of the flags except the ISTPT bit have been set.
- (9) PSI5SPCISC0: PSI5-S CPU Interrupt Status ch0 register.
Clear the flag indicating the payload data parity error.

For the setting of frame reception from a transceiver, see **Section 25.5.2.1(1), PSI5 Frame Reception Setting (No Payload Checksum in Reception Frame)**.

25.5.7 Initialization and Operation Stop Procedures

25.5.7.1 Initialization by Software Reset

This section describes how to initialize PSI5-S by a software reset.

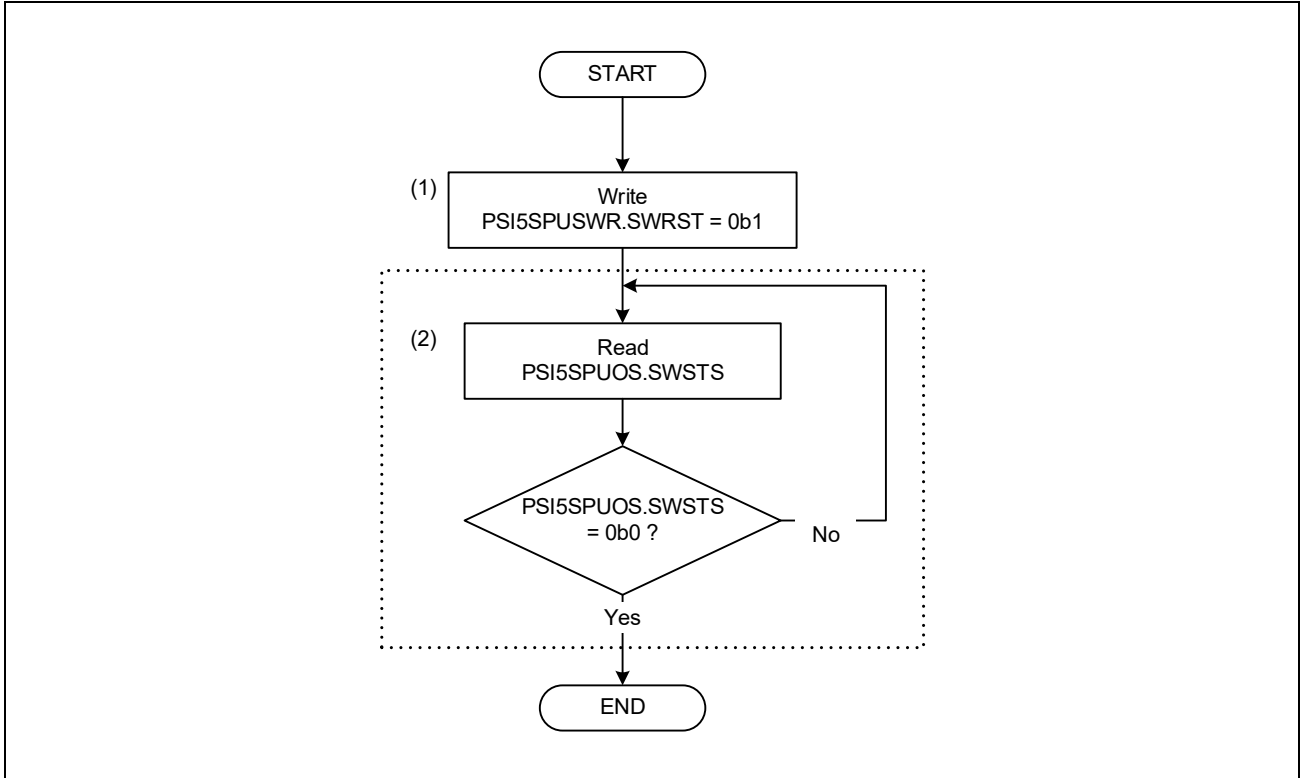


Figure 25.26 Initialization by Software Reset Procedure

- (1) PSI5SPUSWR: PSI5-S/UART Software Reset register
Issue a software reset.
- (2) PSI5SPUOS: PSI5-S/UART Operation Status register
Wait until the SW reset bit changes to 0.

25.5.7.2 DDSR Transmission Stop

This section describes how to stop DDSR transmission.

The transmission stop operation can be performed for individual channels.

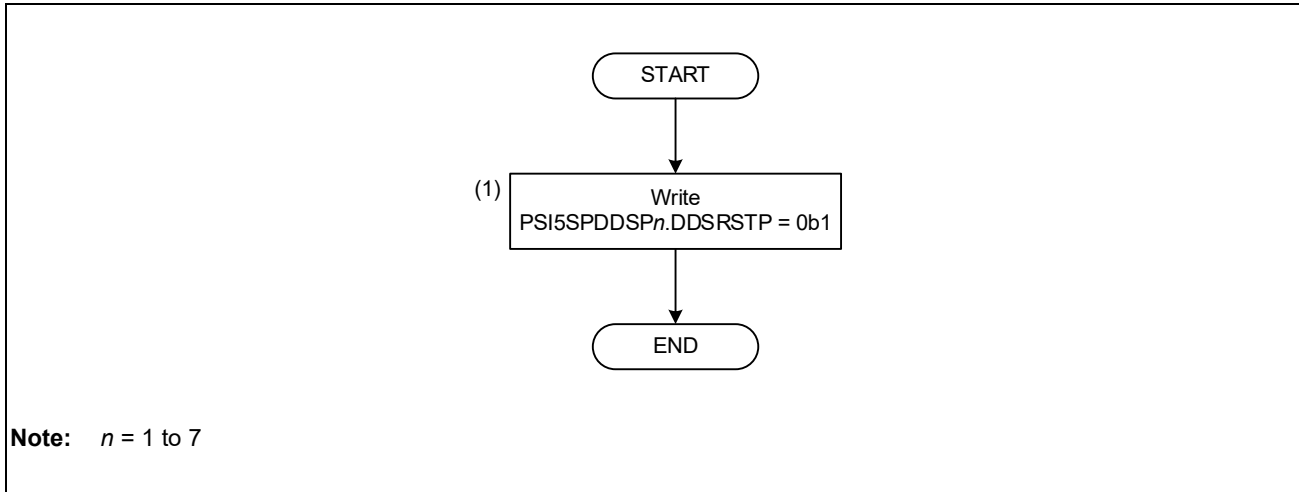


Figure 25.27 DDSR Transmission Stop Operation Procedure

- (1) PSI5SPDDSP n : PSI5-S DDSR Stop ch n register (n : 1 to 7)
Stop data transmission from DDSR.

25.5.7.3 MailBox Initialization

This section describes the operation to initialize mailboxes (MBs).

This operation initializes the MBs for all channels together.

This operation must be performed when PSI5-S is in configuration mode.

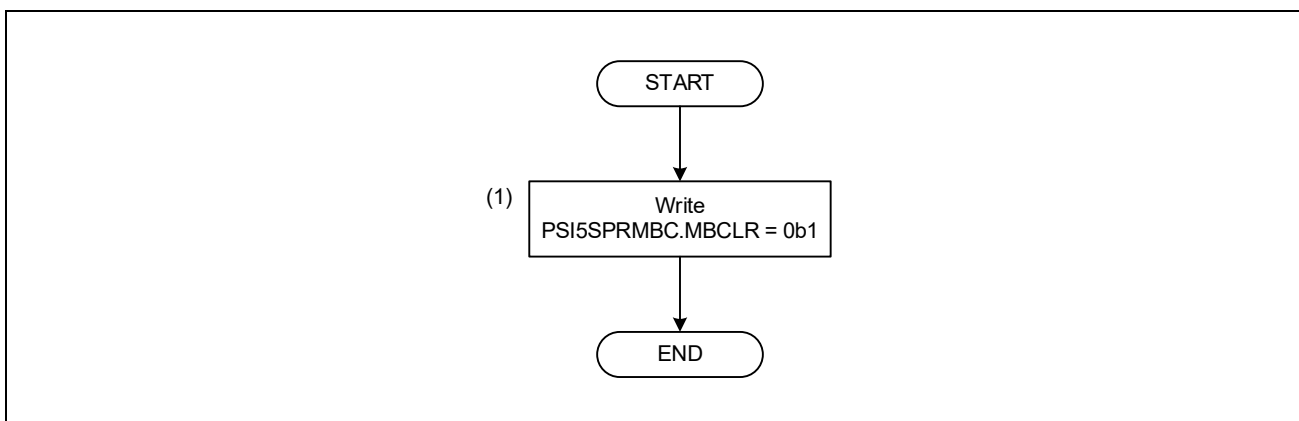


Figure 25.28 Mailbox Initialization Procedure

- (1) PSI5SPRMBC: PSI5-S Receive MailBox Data Clear register
Write 1 to the MBCLR bit to clear the mailboxes.

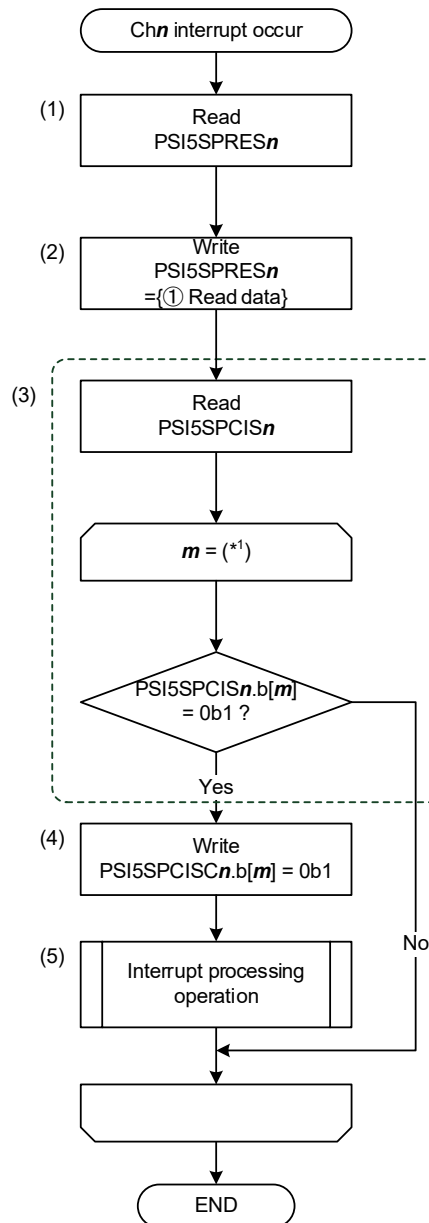
25.5.8 Interrupt Processing Procedures

25.5.8.1 Interrupt Processing in PSI5S Mode

This section describes how to process interrupts in PSI5S mode.

Interrupts occur on individual channels.

This interrupt processing operation can be used for channels 0 to 7.



Note: $n = 0$ to 7

Note 1. When $n = 0$: $m = 9$ to 7, 5 to 0
When $n = 1$ to 7: $m = 11$ to 7, 5 to 1

Figure 25.29 Interrupt Processing Procedure

- (1) **PSI5SPRES n** : PSI5-S Receive Error Status chn register (n : 0 to 7)
Check the frame that caused the reception error. (This step is optional.)
- (2) **PSI5SPRESC n** : PSI5-S Receive Error Status Clear chn register (n : 0 to 7)
Clear the flag indicating the status of the frame that caused the reception error. (This step is optional.)
- (3) **PSI5SPCIS n** : PSI5-S CPU Interrupt Status chn register (n : 0 to 7)
Read the interrupt status register, and check all interrupt factors.
- (4) **PSI5SPCISC n** : PSI5-S CPU Interrupt Status chn register (n : 0 to 7)
Clear the bit corresponding to the relevant interrupt factor.
- (5) Perform the interrupt process that corresponds to the interrupt factor.

Note that processes (3) and (4) must be performed for each interrupt factor bit.

25.5.8.2 Interrupt Processing in UART Mode

This section describes how to process interrupts in UART mode.

Interrupts can occur during both UART reception and UART transmission.

For the processing of interrupts during UART reception, see **Section 25.5.10.1, UART Reception**.

For the processing of interrupts during UART transmission, see **Section 25.5.10.2, UART Transmission**.

25.5.9 DMA Request Processing Procedures

25.5.9.1 DMA Request (Reception) Processing in PSI5S Mode

This section describes how to process DMA requests (reception) in PSI5S mode.

DMA requests (reception) can occur on individual channels.

This request processing operation can be used for channels 0 to 7.

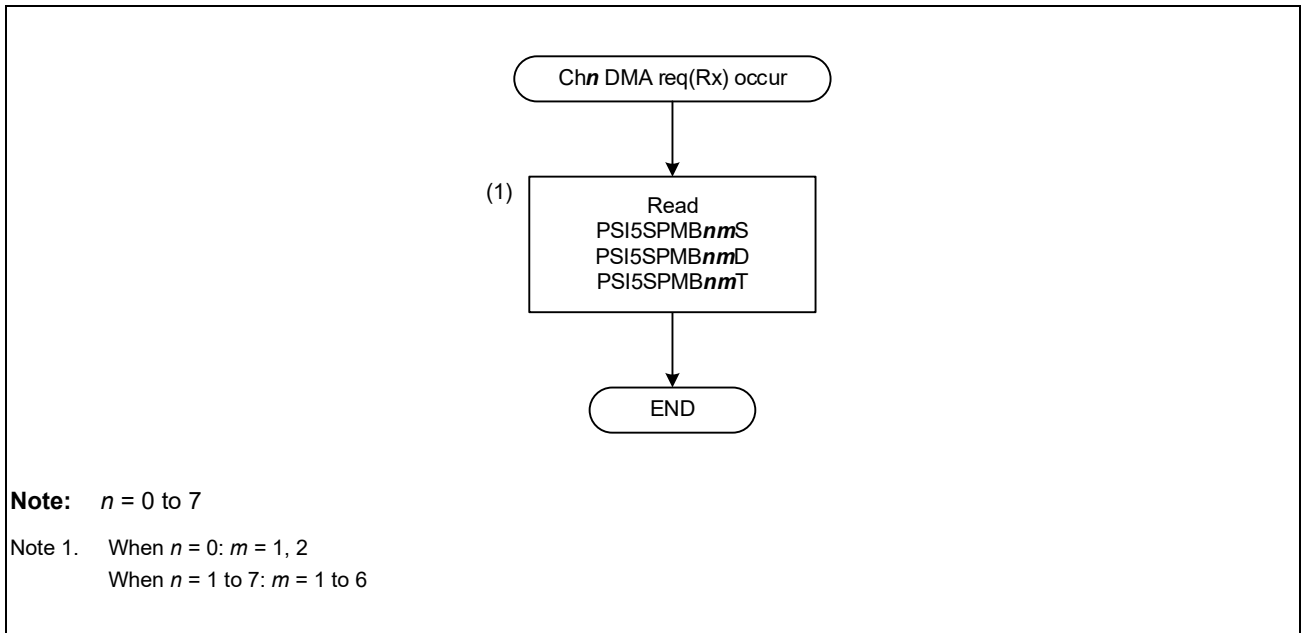


Figure 25.30 Procedure of DMA Request (Reception) Processing in PSI5S Mode

- (1) PSI5SPMBnmS: PSI5-S Receive MailBox ch0 Frm1 Status register (when $n = 0$, $m = 1$ to 2; when $n = 1$ to 7, $m = 1$ to 6)
 PSI5SPMBnmD: PSI5-S Receive MailBox ch0 Frm1 Data register (when $n = 0$, $m = 1$ to 2; when $n = 1$ to 7, $m = 1$ to 6)
 PSI5SPMBnmT: PSI5-S Receive MailBox ch0 Frm1 Timestamp register (when $n = 0$, $m = 1$ to 2; when $n = 1$ to 7, $m = 1$ to 6)
 Check the mailbox data corresponding to the relevant frame.

NOTE

In asynchronous mode of Ch1 to Ch0, the data storage position is switched from frm1 to frm2, ..., frm6, and frm1, in this order, each time a PSI5 frame is received. Therefore, when received data is to be read by DMA, storage addresses must be managed by software. (Data storage position of Ch0 is always frm1.)

25.5.9.2 DMA Request (Transmission) Processing in PSI5S Mode

This section describes how to process DMA requests (transmission) in PSI5S mode.

DMA requests (transmission) can occur on individual channels.

This request processing operation can be used for channels 0 to 7.

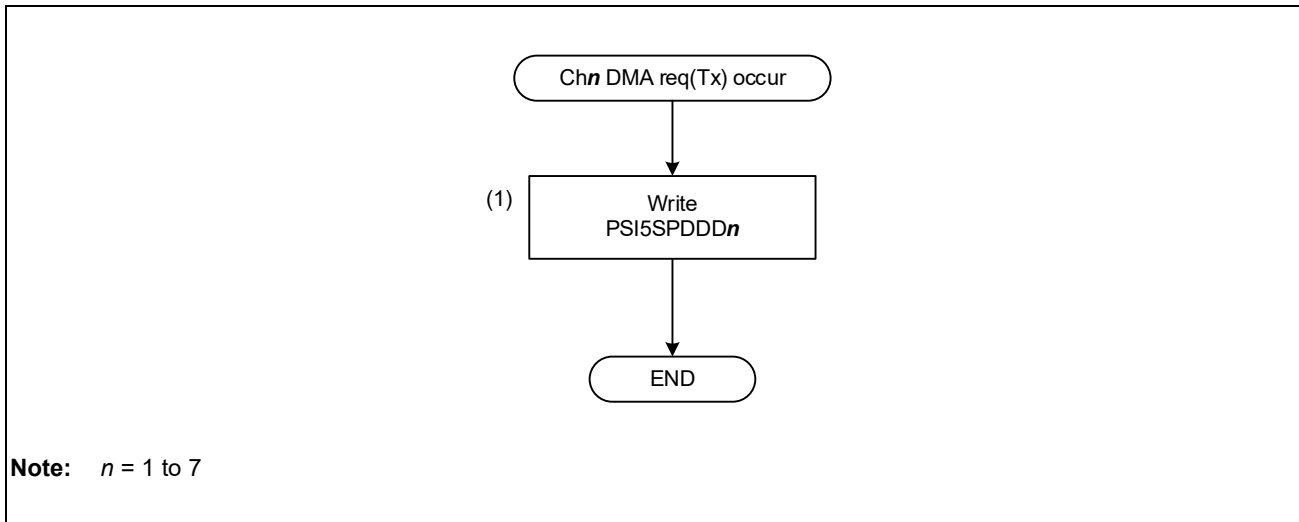


Figure 25.31 Procedure of DMA Request (Transmission) Processing in PSI5S Mode

- (1) PSI5SPDDD n : PSI5-S DDSR Data Ch n register (n : 1 to 7)
Set transmit data in DDSR.

25.5.9.3 DMA Request (Reception) Processing in UART Mode

This section describes how to process DMA requests (reception) in UART mode.

DMA requests (reception) occur on channel 7.

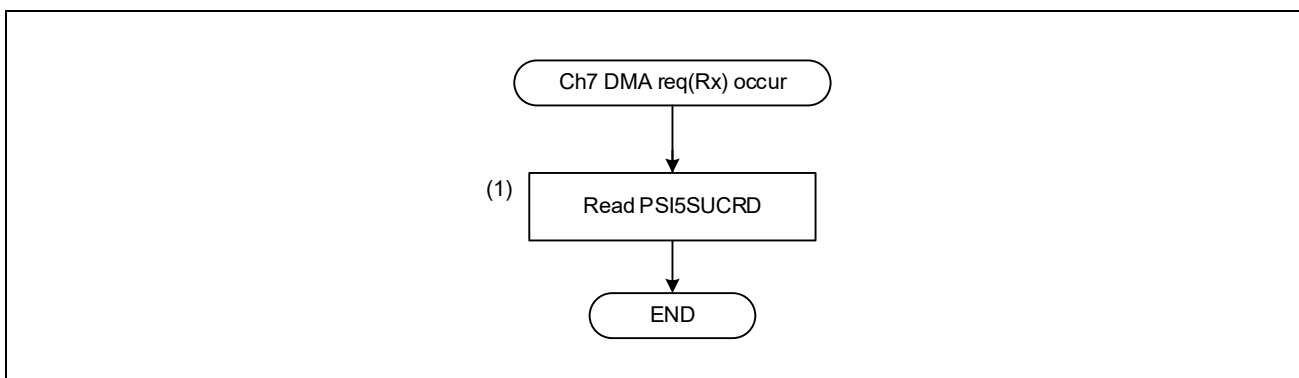


Figure 25.32 Procedure of DMA Request (Reception) Processing in UART Mode

- (1) PSI5SUCRD: UART Communication Rx Data register
Read receive data from this register.

25.5.9.4 DMA Request (Transmission) Processing in UART Mode

This section describes how to process DMA requests (transmission) in UART mode.

DMA requests (transmission) occur on channel 7.

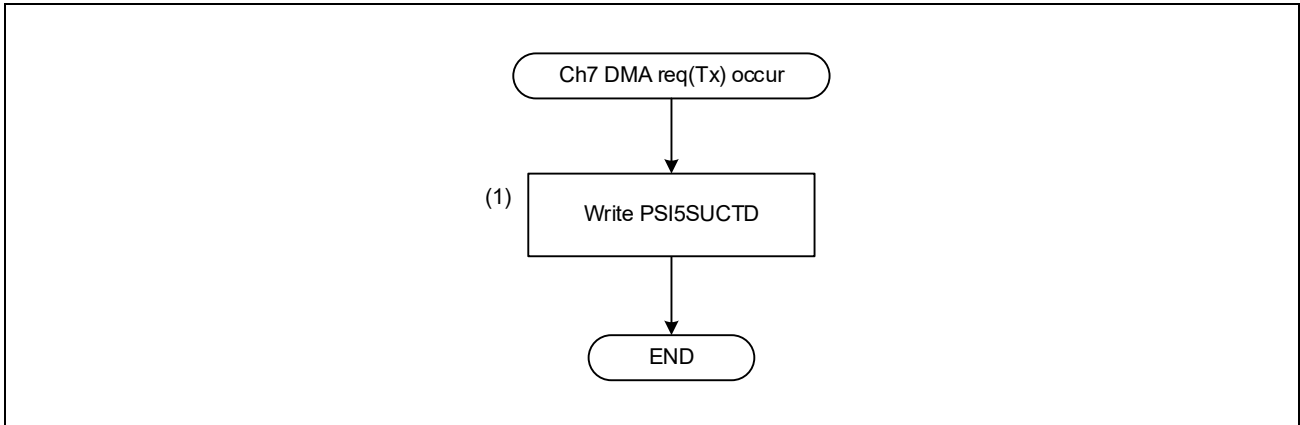


Figure 25.33 Procedure of DMA Request (Transmission) Processing in UART Mode

- (1) PSI5SUCTD: UART Communication Tx Data register
Write transmit data to this register.

25.5.10 UART Mode Communication Procedures

This section describes UART communication procedures.

UART communication can be performed only in UART mode.

25.5.10.1 UART Reception

Figure 25.34 below shows the processing procedure of normal UART reception.

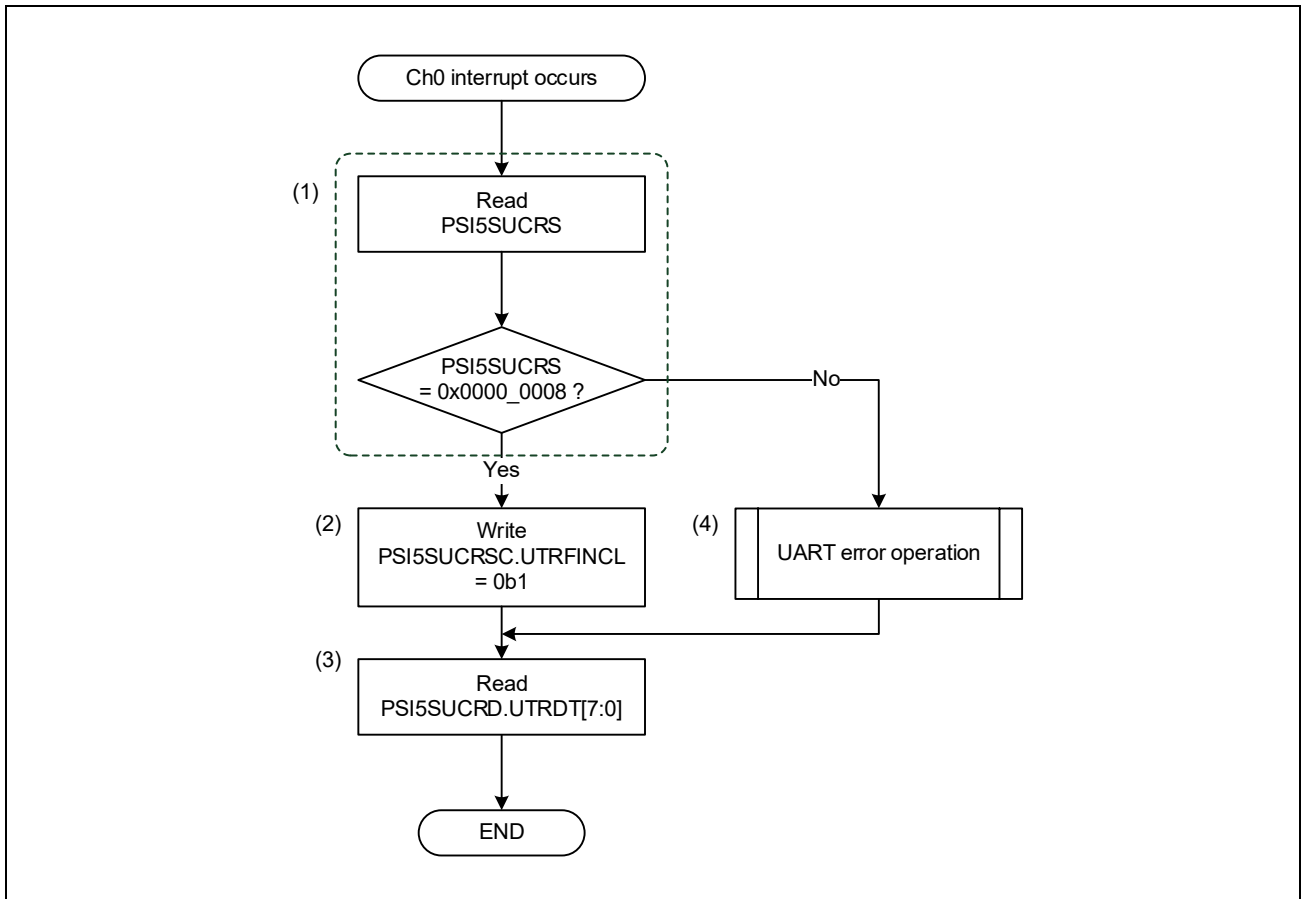


Figure 25.34 UART Reception Procedure

- (1) PSI5SUCRS: UART Communication Rx Status register
Read the UART reception status register, and check that the flag indicating the end of reception is set.
- (2) PSI5SUCRSC: UART Communication Rx Status Clear register
If the flag indicating the end of reception is set, clear it.
- (3) PSI5SUCRD: UART Communication Rx Data register
Read receive data from this register.
Note: If this step is skipped, a reception overrun error occurs at the time of receiving the next data.
- (4) If the flag indicating the end of reception is not set, perform error processing.

Figure 25.35 shows the processing to be performed when a reception overrun error, receive framing error, or receive parity error occurs.

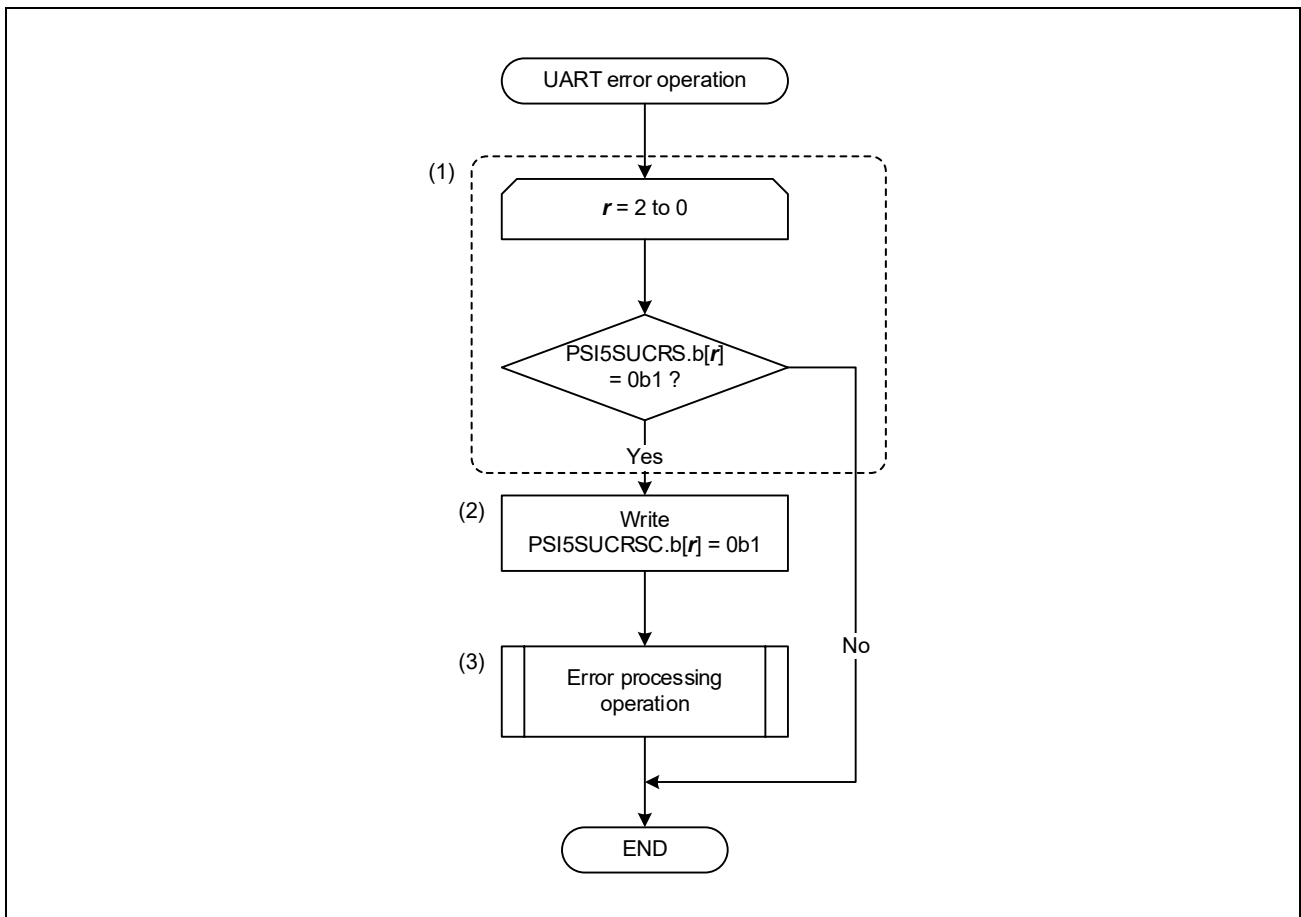


Figure 25.35 UART Reception Error Processing Procedure

- (1) Check the states of error factor bits in the UART reception status register.
- (2) PSI5UCRSC: UART Communication Rx Status Clear register
If an error-cause flag is set, clear the flag.
- (3) Perform the processing corresponding to the indicated error cause.

25.5.10.2 UART Transmission

The procedure of UART transmission uses the UART transmission busy flag or the transmission end interrupt.

Figure 25.36 shows the UART transmission procedure using the UART transmission busy flag.

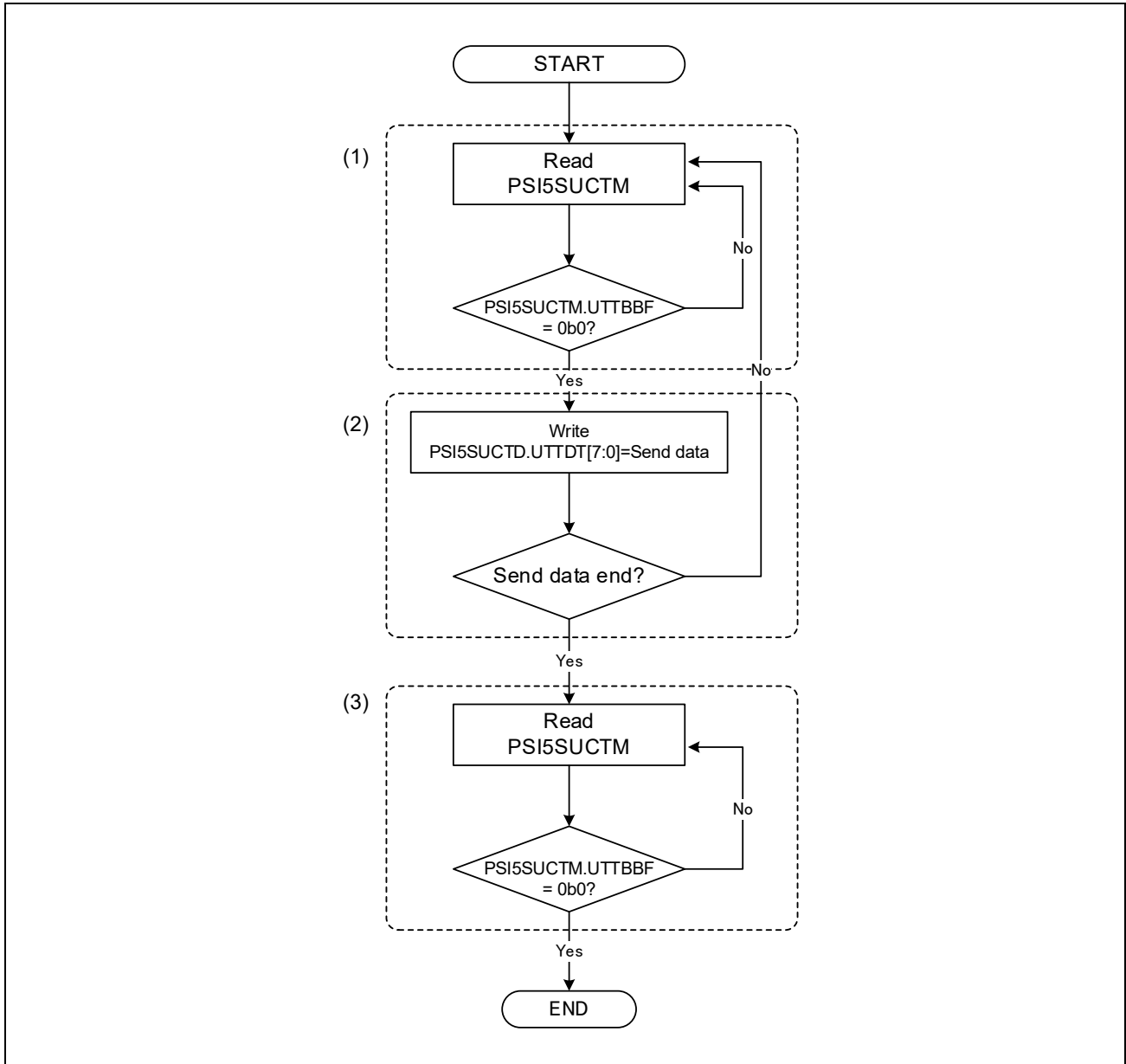


Figure 25.36 UART Transmission Procedure (Using UART Transmission Busy Flag)

- (1) PSI5SUCTM: UART Communication Tx Monitoring register
Read the UART communication Tx monitoring register to check that the UART Tx buffer is not busy.
- (2) PSI5SUCTD: UART Communication Tx Data register
Write transmit data to this register.
- (3) PSI5SUCTM: UART Communication Tx Monitoring register
Read the UART communication Tx monitoring register to check that the UART Tx buffer is not busy.

Figure 25.37 below shows the UART transmission procedure using the transmission end interrupt.

Because no transmission end interrupt of the previous transmission is generated for the first transmission operation, the UART transmission busy flag must be used for the first transmission operation and the transmission end interrupt thereafter.

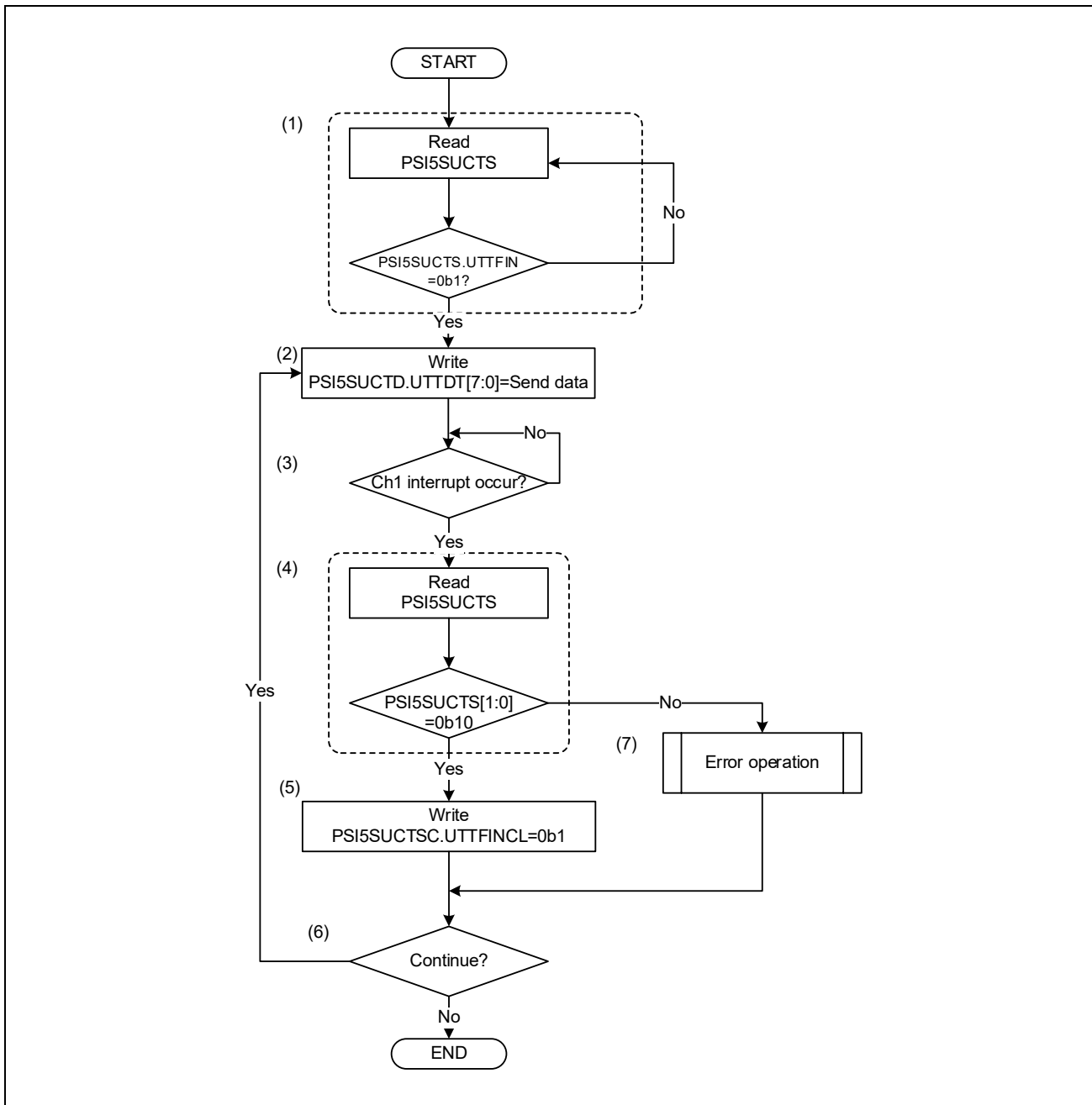


Figure 25.37 UART Transmission Procedure (Using Transmission End Interrupt)

- (1) PSI5SUCTS: UART Communication Tx Status register
Read the UART communication status register to check that the UART is ready for transmission.
- (2) PSI5SUCTD: UART Communication Tx Data register
Write transmit data to this register.
- (3) Wait until a channel 1 interrupt occurs. When it occurs, proceed to step (4).

- (4) PSI5SUCTS: UART Communication Tx Status register
Read the UART communication status register to check that the transmission end flag is set and the overwrite error flag is not set.
- (5) PSI5SUCTSC: Uart Communication Tx Status Clear register
Clear the flag indicating the end of transmission.
- (6) To continue the processing, proceed to step (2). If not, finish the processing.
- (7) Perform the corresponding error processing, see **Figure 25.38**.

Figure 25.38 below shows the procedure of the processing to be performed when a transmission overwrite error occurs.

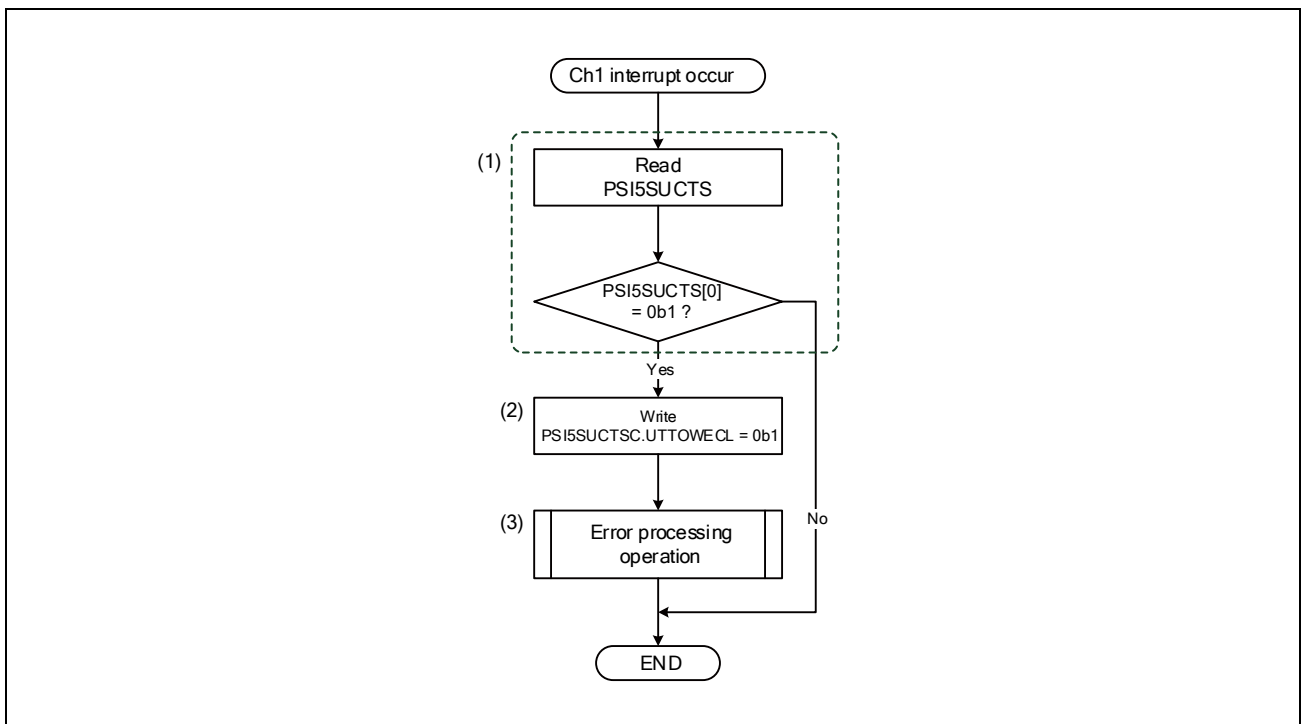


Figure 25.38 UART Transmission Error Processing Procedure

- (1) PSI5SUCTS: UART Communication Tx Status register
Read the UART communication status register to check that the transmission overwrite error flag is set.
If the flag is not set, finish the processing.
- (2) PSI5SUCTSC: UART Communication Tx Status Clear register
Clear the transmission overwrite error flag.
- (3) Perform the processing to be done when a transmission overwrite error occurs.

25.5.11 Loopback Test

25.5.11.1 Loopback Test Setting Procedure

This setting procedure must be performed after the common setting procedure (**Section 25.5.1, Common Setting Procedure**).

Figure 25.39 below shows the Loopback test setting procedure.

PSI5-S enters the test mode only when the values below are written to the register in the order shown in the figure three times.

If a different value is written midway, the whole procedure becomes invalid. In such a case, perform three new writes from the beginning.

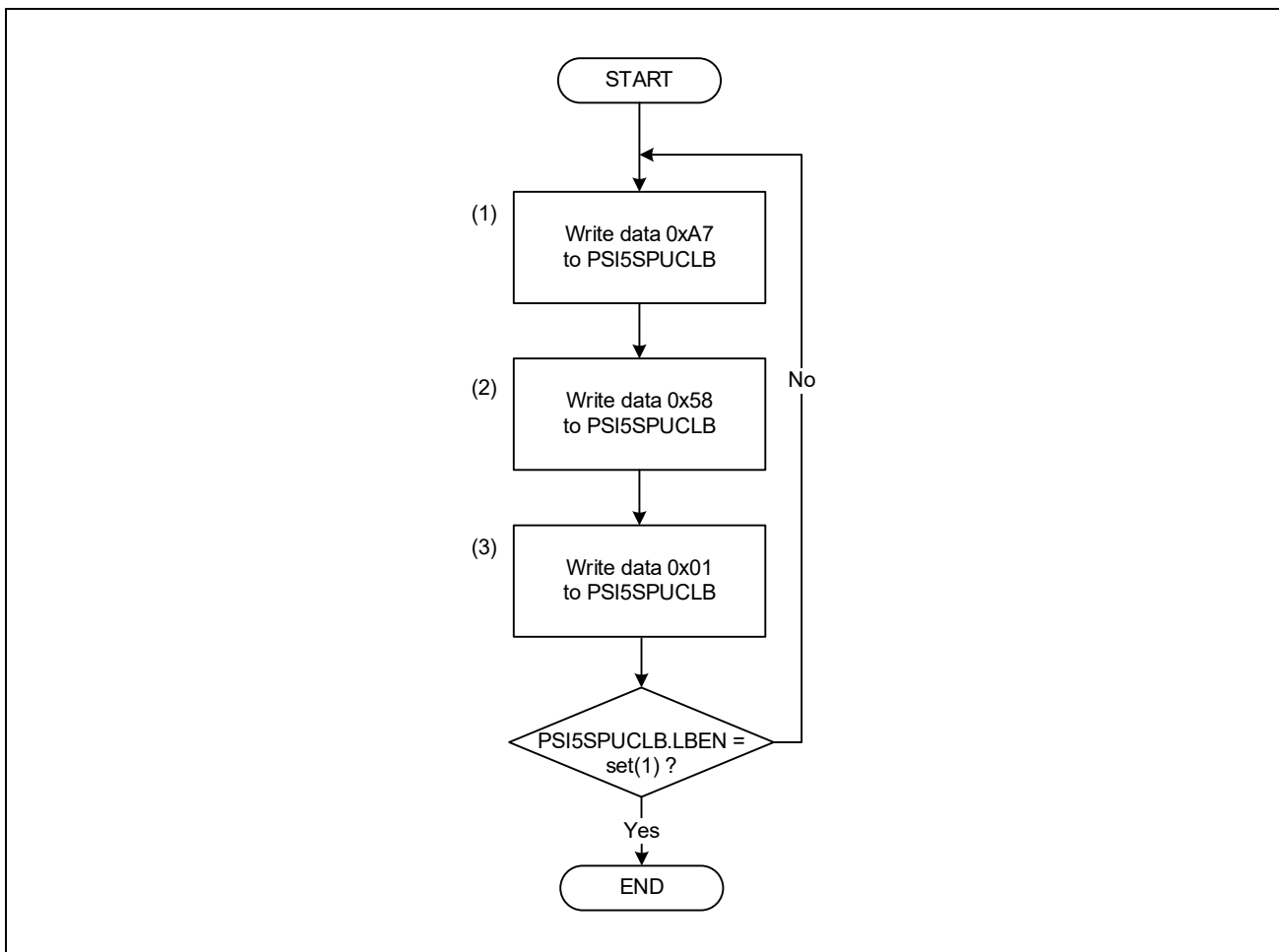


Figure 25.39 Loopback Test Setting Procedure

Then PSI5S mode setting (**Section 25.5.2, PSI5S Mode Setting Procedure**) is performed.

The loopback test is performed under the following conditions:

- Communication mode is asynchronous mode.
- Sync commands are not transmitted (sync pulses are not generated).

25.5.11.2 Loopback Test Procedure

This section describes the procedure to perform a loopback test.

As an example, **Figure 25.40** below shows the procedure of the loopback test in which a PSI5 frame (five packets) on pseudo channel n (n : 1 to 7) are turned back to the reception block and the reception of the frame is checked by using a special function register (SFR).

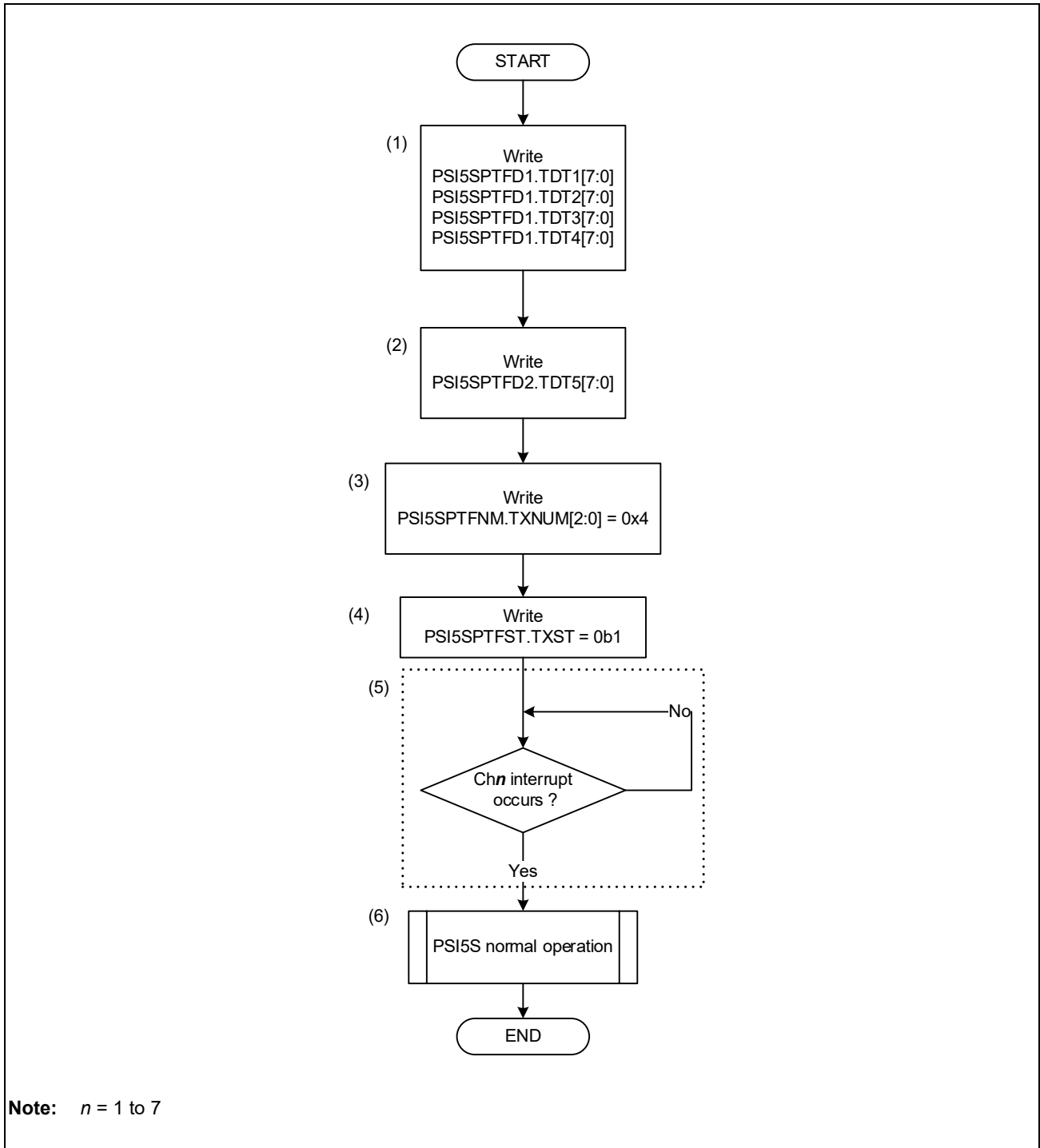


Figure 25.40 Loopback Test Procedure

- (1) PSI5SPTFD1: PSI5-S Tx Frame Data1 register
In this register, set the first to fourth bytes of the data (PSI5 frame on pseudo channel n [n : 1 to 7]) to be transmitted continuously.
- (2) PSI5SPTFD2: PSI5-S Tx Frame Data2 register
In this register, set the fifth byte of the data (PSI5 frame on pseudo channel n [n : 1 to 7]) to be transmitted continuously.
- (3) PSI5SPTFNM: PSI5-S Tx Frame Number register
Specify 5-packet transmission (0x4) in this register.
- (4) PSI5SPTFST: PSI5-S Tx Frame Start register
Write 1 to this register to start transmission.
- (5) Wait until an interrupt outputs on channel n (n : 1 to 7).
- (6) Subsequent processes follow the PSI5 frame reception procedure described in **Section 25.5.6, PSI5 Frame Reception Procedures**.

25.5.12 Configuration Mode Entering Procedure

Figure 25.41 below shows the configuration mode entering procedure.

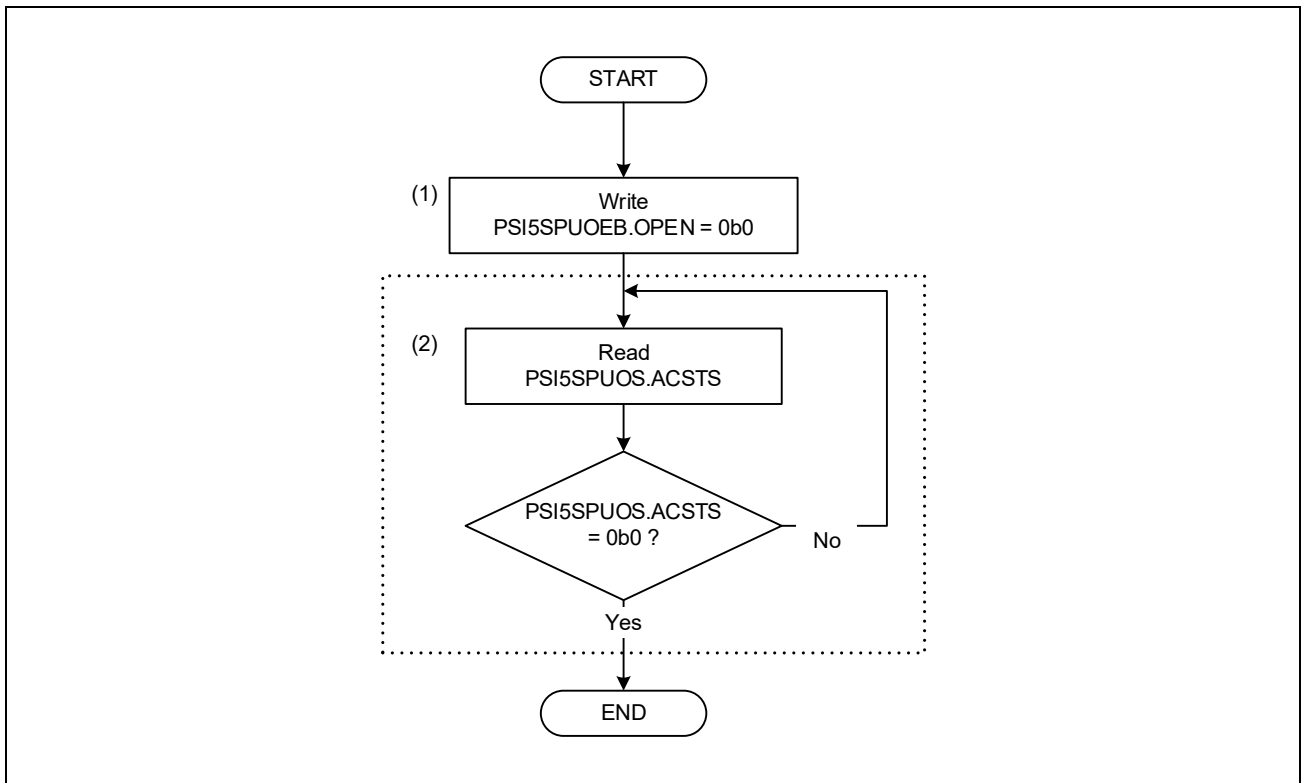


Figure 25.41 Configuration Mode Entering Procedure

- (1) PSI5SPUOEB: PSI5-S/UART Operation EnaBle register
Stop the operation of PSI5-S.
- (2) PSI5SPUOS : PSI5-S/Uart Operation Status register
Wait until the Active Status bit change to 0.

The timing of return to the configuration mode, see **Section 25.4, Operation Modes**.

25.6 Operations

This chapter describes the functional configuration of PSI5-S and PSI5-S operations in individual operation modes.

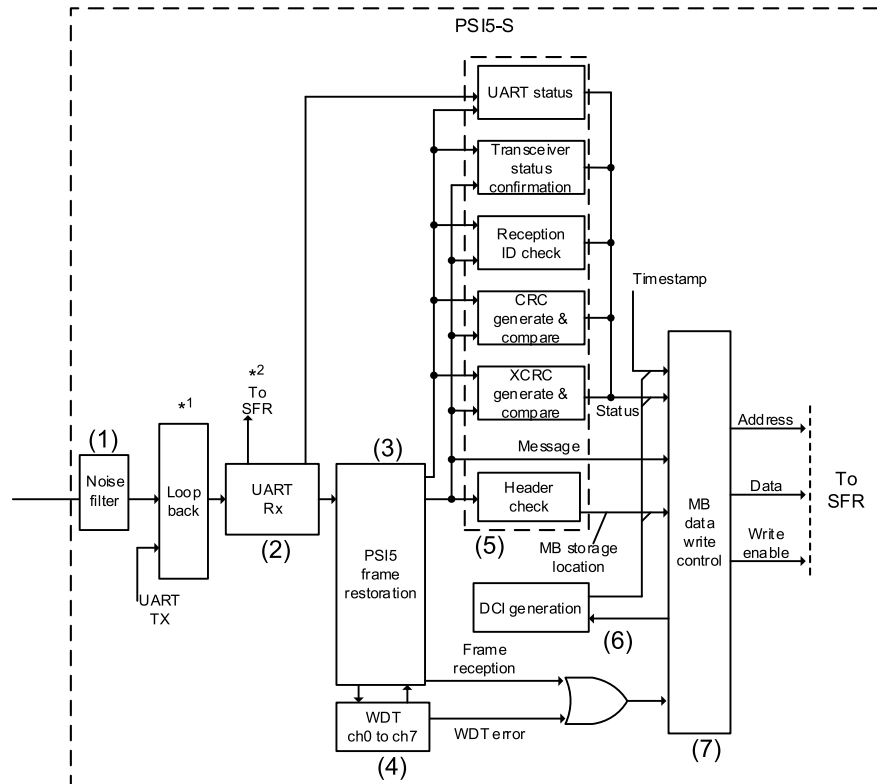
25.6.1 Functional Configuration

The sections that follow describe the functions of PSI5-S.

25.6.1.1 PSI5-S Reception Function

PSI5-S receives sensor-to-ECU data, and stores the status (if a reception error is detected), received data, and timestamp in SFR.

Figure 25.42 below shows the block diagram for PSI5-S reception.



- (1) Take the majority of received signals to remove the noise.
- (2) Perform UART reception, and output 1-byte data and reception error.
- (3) Receive UART frames. If the idle time longer than the specified value has elapsed, the next frame is regarded as a header.
After the header reception, check the number of frames by ChID and FmID. If the specified number of frames has been received, it is restored as PSI5 frames.
- (4) Monitor whether PSI5 frames of each channel receive a specified number of packets within the time.
WDT errors are generated if the time expires without receiving.
- (5) Generate the followings from the restored PSI5 frame data.
 - After generating XCRC/CRC, the error determination status value, UART status flag, and status value from the header
 - Message data
 - MB storage location address from ChID and FmID in the reception header
- (6) Generate the 4-bit DCI value and update it after MB data for 3 addresses is written.
MB data is configured by combining the DCI value with status/message/timestamp data.
- (7) When the WDT error or frame is received, it is stored to the MB storage location in the order of status, message, and timestamp.

Note 1. When loopback is set, UART transmit data is wrapped.

Note 2. When data is received in UART mode, the received data and status are output to SFR.

Figure 25.42 Block Diagram for PSI5-S Reception

For the timing of reception, see Figure 25.72.

The following figure shows the flow of data reception.

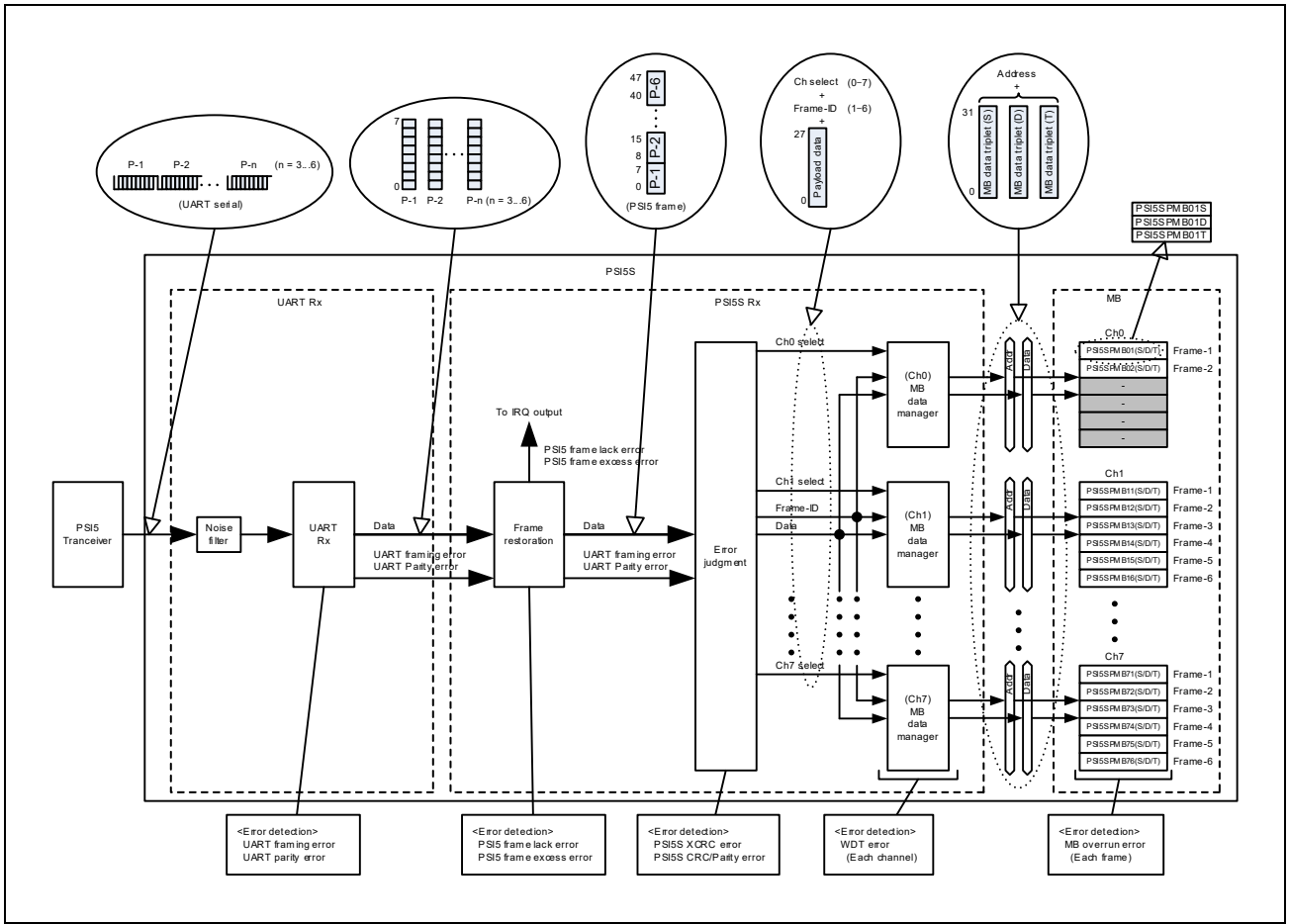


Figure 25.43 PSI5-S Reception Data Flow

25.6.1.2 PSI5-S Transmission Function

PSI5-S generates ECU-to-sensor data, and transmits a Sync command when it is triggered by a synchronization pulse on individual channels.

PSI5-S also has a function to transmit the data set by the CPU via SFR.

The following figure shows the block diagram for PSI5-S transmission.

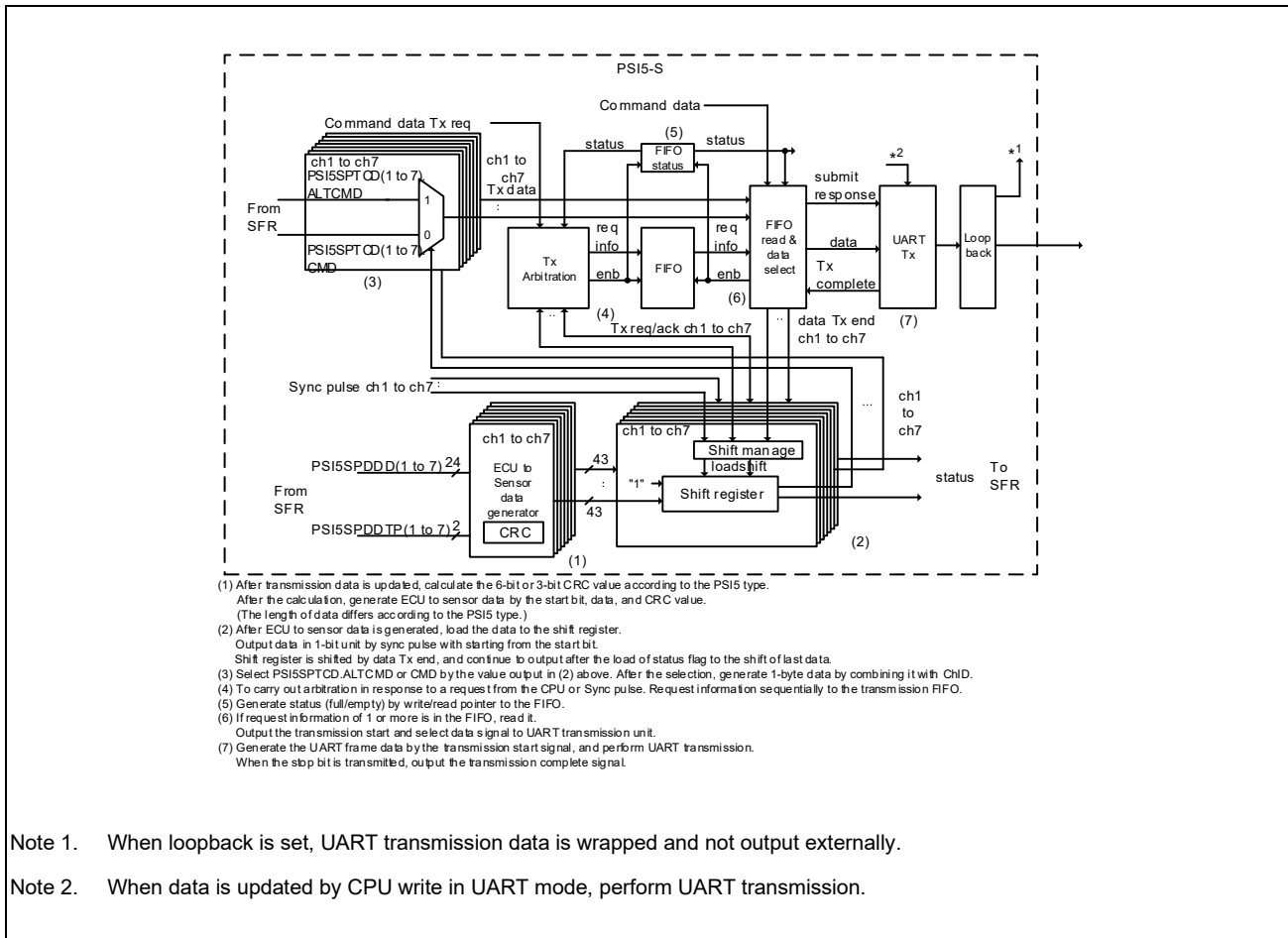


Figure 25.44 Block Diagram for PSI5-S Transmission

For the timing of the transmission, see **Figure 25.66**.

Transmit requests between command data (**Section 25.6.2.3, Normal Transmission (Command Data)**) and ECU-to-sensor data (**Section 25.6.2.4, Normal Transmission (ECU-to-Sensor Data)**) are arbitrated, and then UART starts transmission.

Figure 25.45 below shows the block configuration related to the transmit request arbitration function.

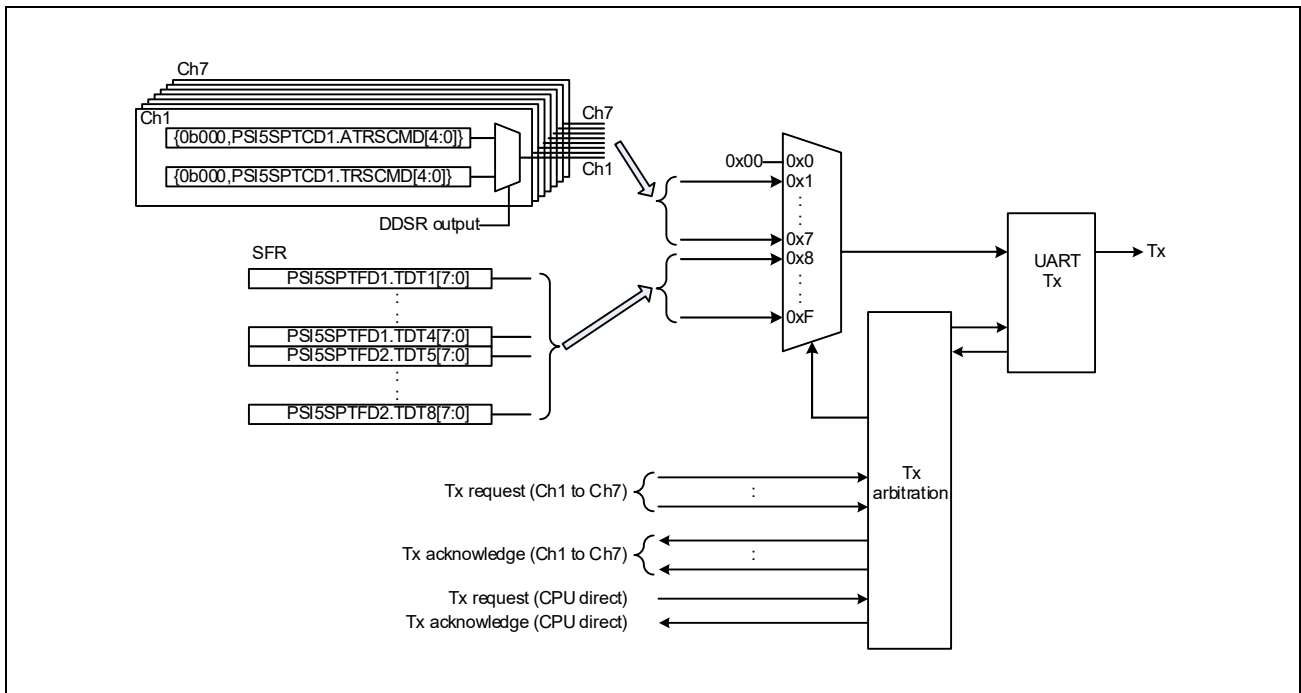


Figure 25.45 Block Diagram for Transmit Request Arbitration

25.6.1.3 Timestamp Generation Function

Based on the timing of clock input from the GTM or the timing of a divided clock (1 to 1023 μ s in 1- μ s steps) generated from PCLK, a timestamp counter runs to measure time. PSI5-S has two timestamp counters for timestamps A and B, and the timestamp counters can be enabled and cleared together or separately.

In synchronous mode of Ch1 to Ch7, the timestamp value is captured at the timing of PSI5 frame header reception or the synchronization pulse. In asynchronous mode of Ch1 to Ch7, the timestamp value is captured in the timing of PSI5 frame header reception. In asynchronous mode of Ch0, the timestamp value is captured in the timing of PSI5 frame header reception. (Ch0 is always asynchronous mode.) The timestamp value of Ch0 is also captured at the timing of a WDT error occurs in any of the channels.

Timestamp capture values are available for channels 0 to 7. When PSI5 frame reception ends, one of the timestamp capture values is selected according to the channel ID in the header, and used as a timestamp value. If the received frame is an invalid frame, the timestamp capture value for channel 0 is selected. Determination of invalid frames follows the specifications listed in **Table 25.93** and **Table 25.95**. If a WDT error occurs, the timestamp capture value of Ch0 is used as the timestamp value.

Note that timestamp capture values are all zeros when the timestamp capture enable bit (PSI5SPRCF1[0:7].TSEN) is 0 (disable). Also, the timestamp capture values can be cleared by the clear bit (PSI5SPTCDC[0:7]).

Figure 25.46 below shows the block diagram for the timestamp generation function.

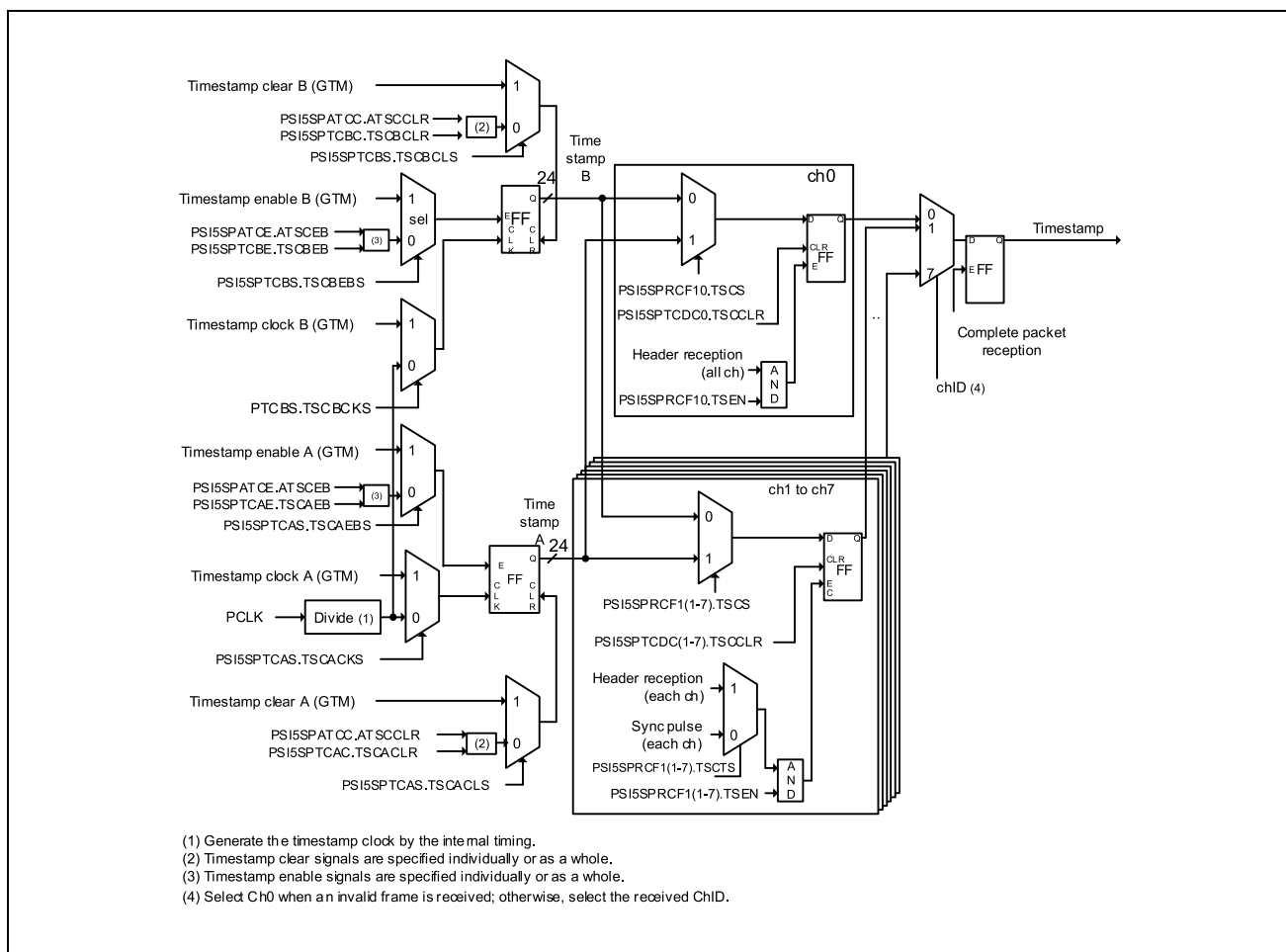


Figure 25.46 Block Diagram for Timestamp Generation

Figure 25.47 below shows an operation in which a timestamp capture value is fetched when the header of a PSI5 frame is received, and a timestamp value is generated when PSI5 frame reception ends.

The same operation can be performed in both synchronous and asynchronous modes.

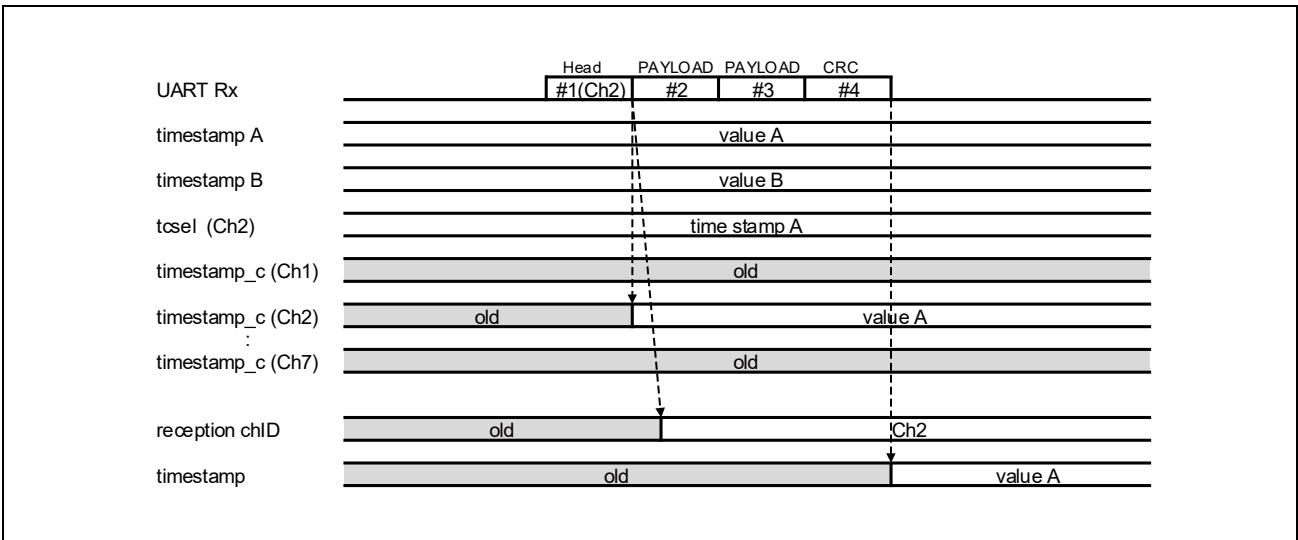
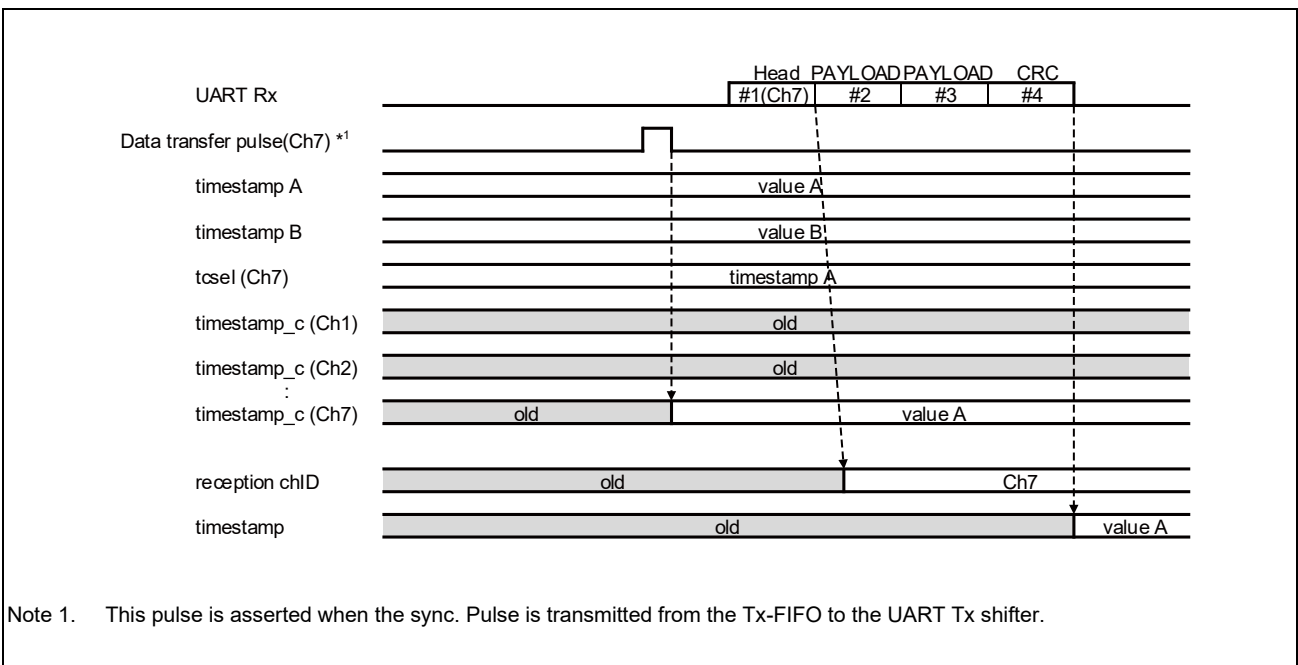


Figure 25.47 Timestamp Generation Operation (Timestamp Capture Triggered by Header Reception)

Figure 25.48 below shows an operation in which a timestamp capture value is fetched when a synchronization pulse outputs, and a timestamp value is generated when PSI5 frame reception ends.

This operation can be performed only in synchronous mode.



Note 1. This pulse is asserted when the sync. Pulse is transmitted from the Tx-FIFO to the UART Tx shifter.

Figure 25.48 Timestamp Generation Operation (Timestamp Capturing Triggered by Synchronization Command Transmission)

Figure 25.49 below shows an operation in which a timestamp capture value is fetched when a WDT error occurs, and a timestamp value is generated immediately.

The same operation can be performed in both synchronous and asynchronous modes.

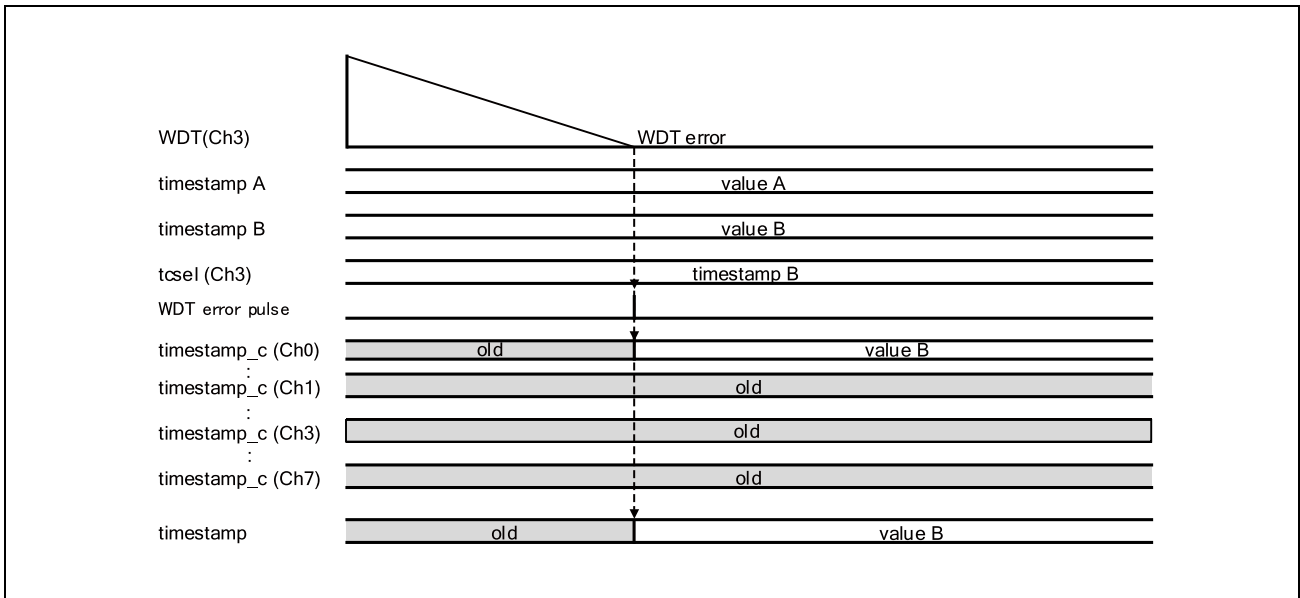


Figure 25.49 Timestamp Generation Operation (Capturing Triggered by WDT Error)

25.6.1.4 Synchronization Pulse Generation Function

The timing of the synchronization pulse for channels 0 to 7 is selected from the rising edge of the synchronization trigger signal for each channel input from the GTM or the timing of a divided clock (1 to 1024 μ s in 1- μ s step) generated from PCLK.

The divided clock timing generation function operates when the function is selected in SFR and its operation is enabled (PUOEB.OPEN = 0x1). With other settings, counters stop with a counter value of 0.

The following figure shows a block diagram for the synchronization pulse generation circuit:

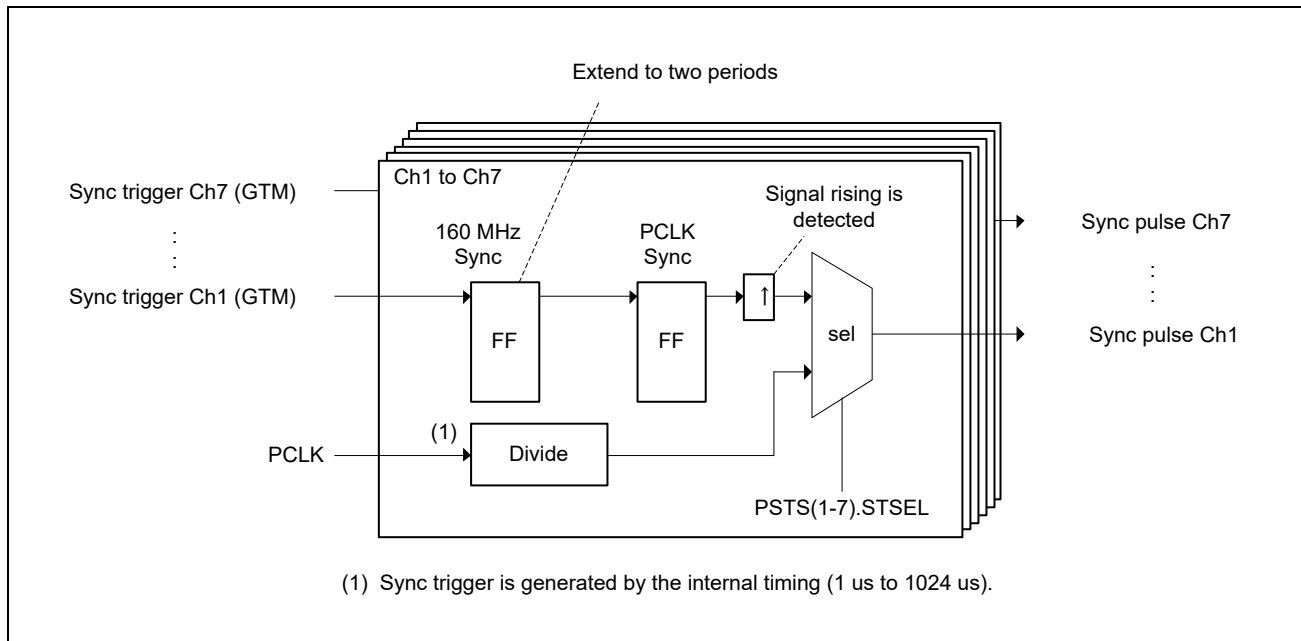


Figure 25.50 Block Diagram for Synchronization Pulse Generation

25.6.2 Operations in PSI5S Mode

PSI5-S performs UART reception of four to six frames when the operation mode is set to PSI5S mode (PSI5SPUOMD.OPMD = 0x1) and the module operation is enabled (PSI5SPUOEB.OPEN = 0x1).

PSI5-S restores PSI5 frame data with corresponding functions shown in **Figure 25.42**.

After restoring the data, PSI5-S checks for reception errors, and then stores data (such as the status value generated from the received frame data, payload value of the received frame data, and the timestamp value generated as described in **Section 25.6.1.3, Timestamp Generation Function**) as mailbox data in SFR.

The following figure shows the flow of reception.

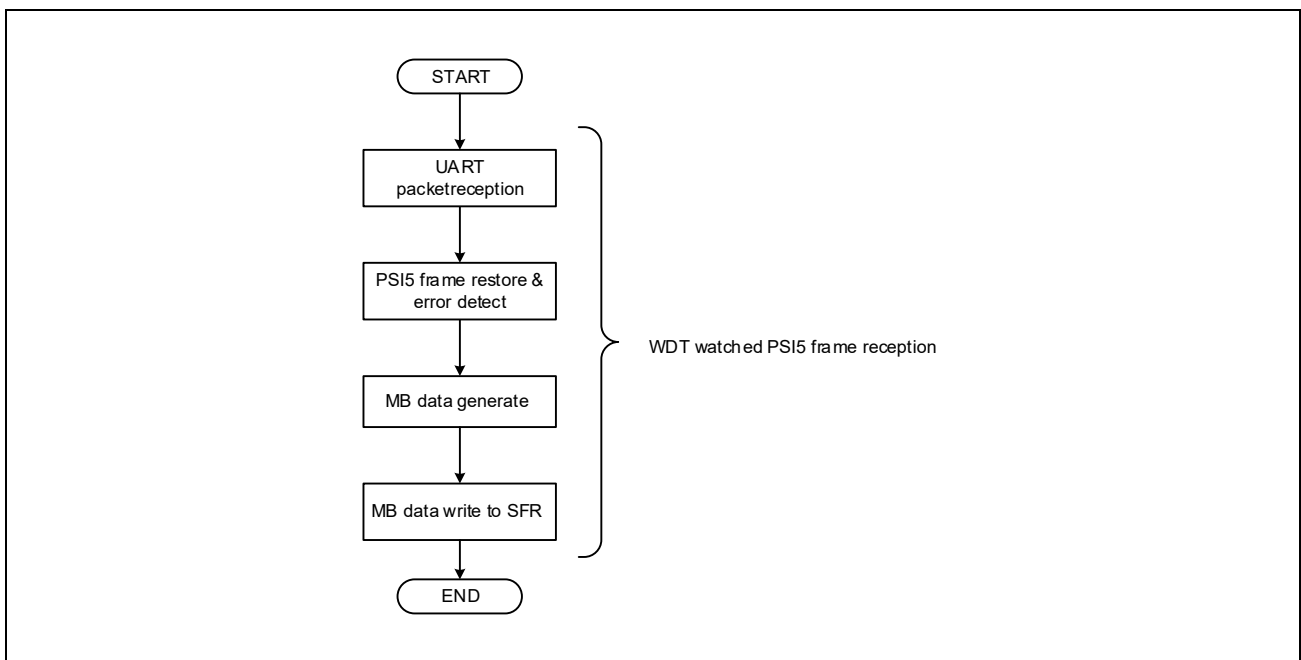


Figure 25.51 Flow of Reception

25.6.2.1 Normal Reception (in Synchronous Mode)

PSI5-S performs reception according to the flow shown in **Figure 25.51**.

In synchronous mode, PSI5-S starts reception with the synchronization pulse for each channel used as the trigger, and receives a specified number of PSI5 frames.

A WDT installed for each channel is used to monitor reception to determine whether the specified number of PSI5 frames are received within a specified time.

(1) Receiving UART Frames

PSI5-S receives UART frames, and restores PSI5 frame data.

For the UART specifications, see **Section 25.2.1, Functional Overview**.

UART receive data is filtered by a noise filter (filtering of three prescaler cycles of data signal based on a majority decision) according to the settings.

Sampling enable generate counter and Sampling counter is reset and UART reception starts when the start bit is detected at a sampling point that starting from the falling edge detection After noise filtering.

NOTE

A sampling point is different in oversample number (odd or even).

Figure 25.52 and **Figure 25.53** below show examples of UART reception.

The following example assumes that the oversample number is set to 5.

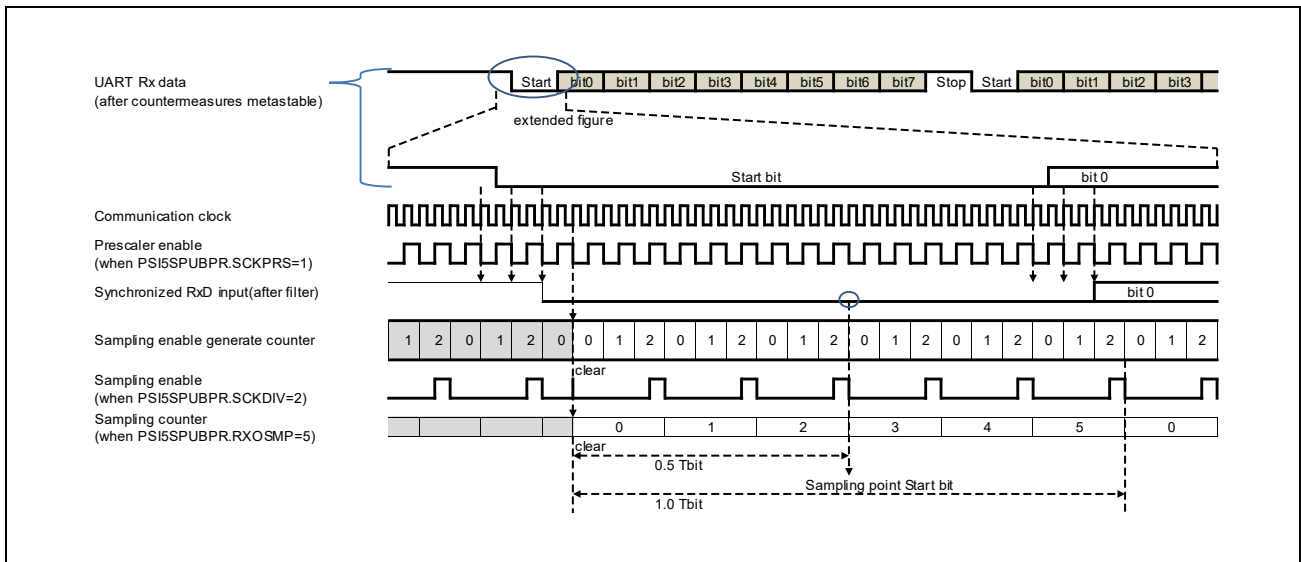


Figure 25.52 UART Frame Reception Start (PSI5SUBPR.RXOSMP = 5 [ODD])

The following example assumes that the oversample number is set to 4.

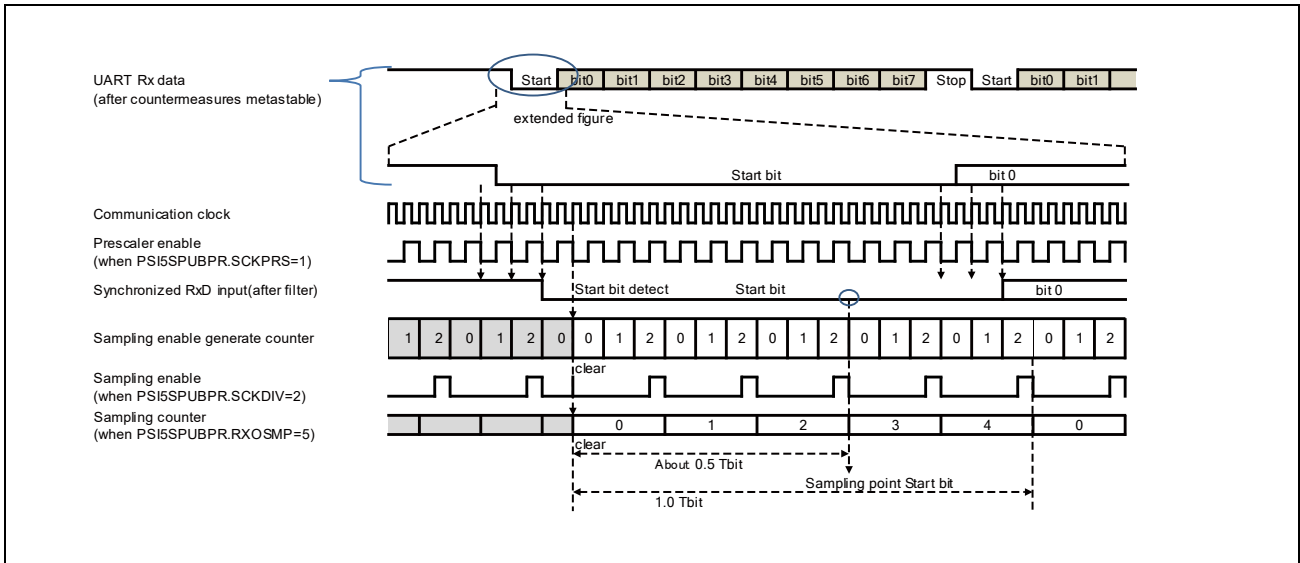


Figure 25.53 UART Frame Reception Start (PSI5SUBPR.RXOSMP = 4 [EVEN])

Figure 25.54 below shows an example of an operation in which UART frames are received continuously.

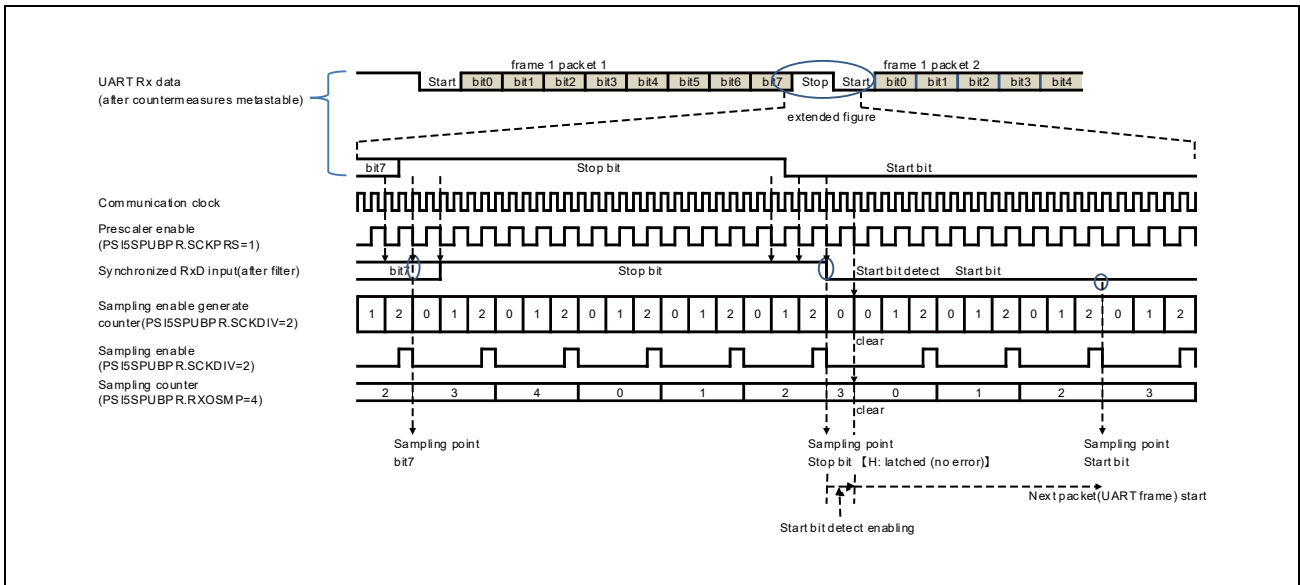


Figure 25.54 UART Frame Reception End (Frame Continue) (PSI5SPRCF10.PFRMIDLE = 0)

Figure 25.55 below shows an example of an operation in which the last packet frame is received.

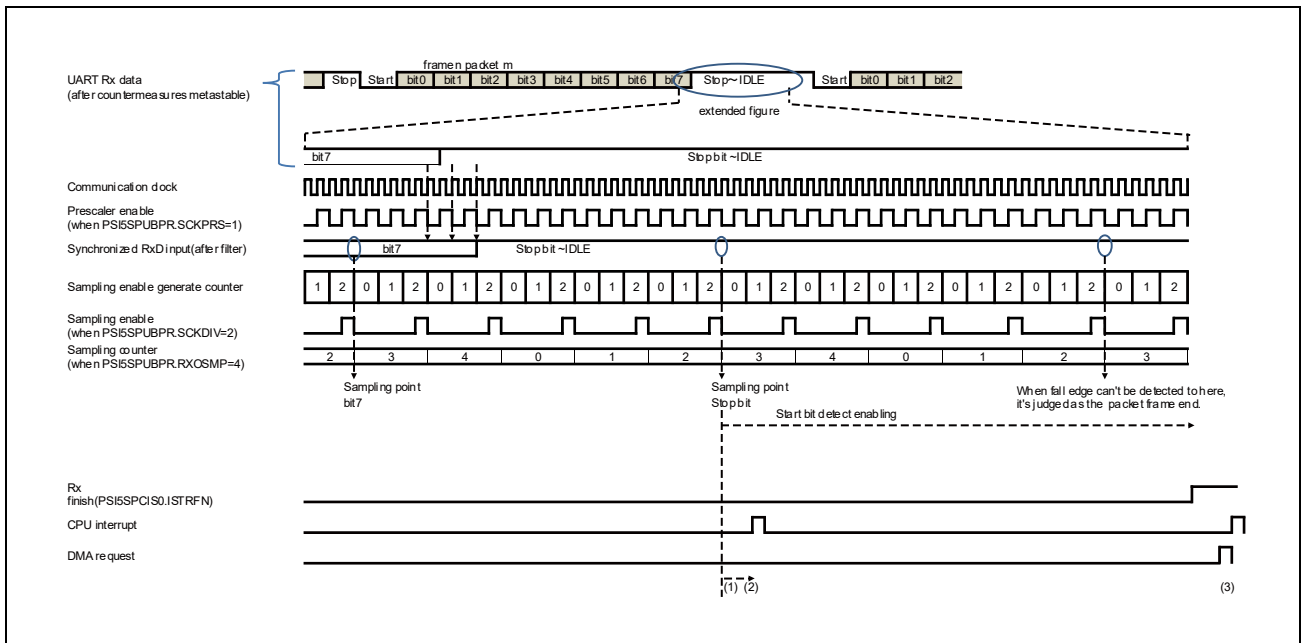


Figure 25.55 UART Frame Reception End (Packet Frame End) (PSI5SPRCF10.PFRMIDDLE = 0)

- (1) The stop bit of the last packet is detected.
- (2) The asynchronous transfer processing based on PCLK and the communication clock, causes a delay of several clock cycles.
- (3) After 50-60 cycles of PCLK (for restoring PSI5 frame data, determining errors and storing mailbox data), the flag indicating the end of reception is set (PSI5SPRCF10.PFRMIDDLE = 1) and an interrupt is output.

Figure 25.56 below shows an example of an operation in which the end of packet frame is detected.

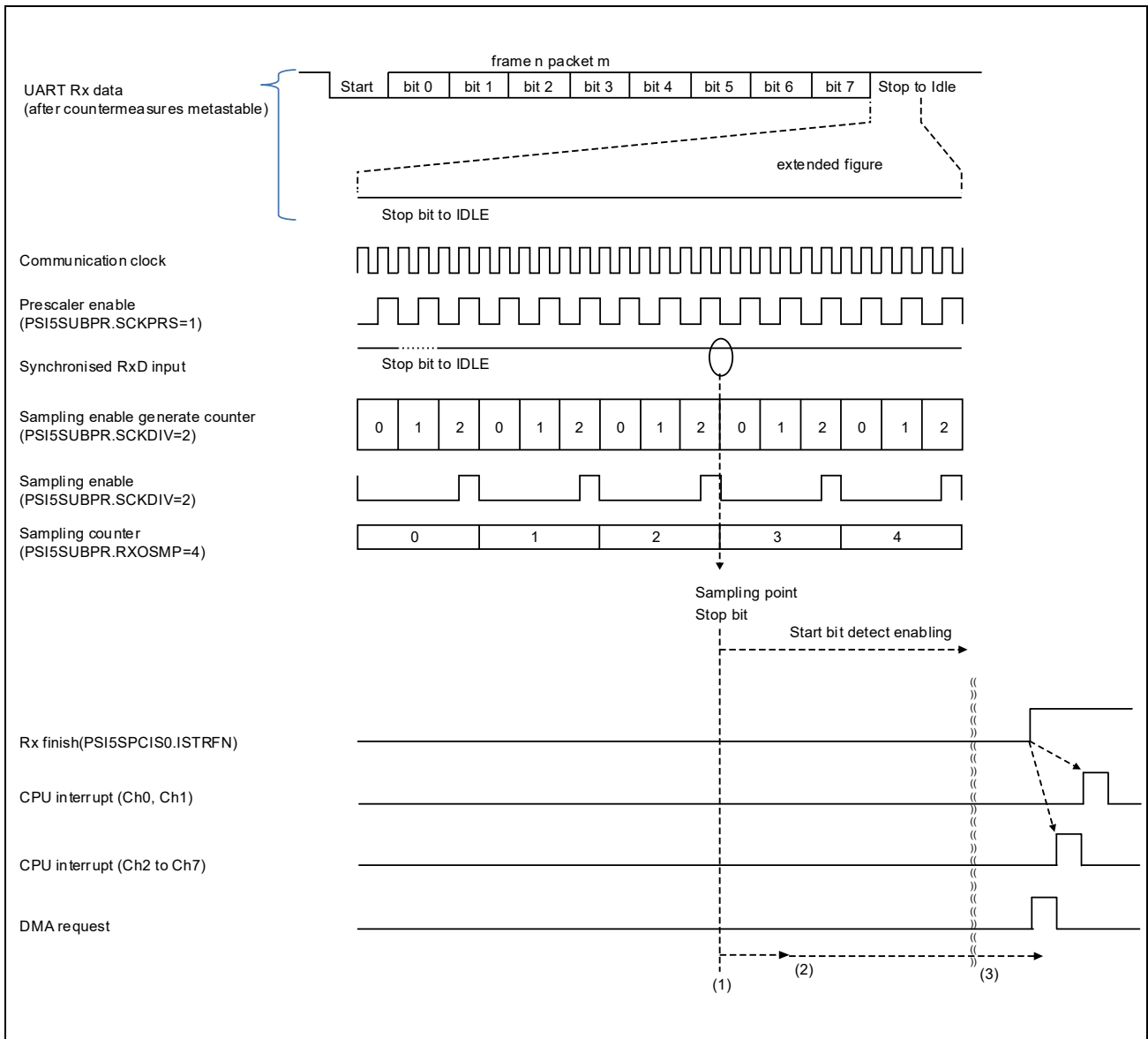


Figure 25.56 UART Frame Reception (Channel End) (PSI5SPRCF10.PFRMIDLE=0)

- (1) When the stop bit of the last packet is detected, the flag indicating the end of reception is set (PSI5SPCIS0.ISTRFN = 1).
When reception of all packet frames in a channel ends, an interrupt and a DMA request are output.
- (2) The asynchronous transfer processing based on PCLK and the communication clock causes a delay of several clock cycles.
- (3) After 50-60 cycle of PCLK (for Restoring PSI5 frame data, determining errors and storing mailbox data), the flag indicating the end of reception is set (PSI5SPCIS0.ISTRFN = 1) and an interrupt outputs.
When reception of all packet frames in a channel ends, an interrupt and a DMA request are output.

PSI5 frame data consists of a header, payload, and XCRC bits (**Figure 25.3**).

UART frames are input successively, and the maximum interval to the next packet frame is specified in the PSI5SPRCF10.PFRMIDLE[3:0] bits.

If the actual interval between frames exceeds the specified value, the UART frame that is received next is treated as a header.

After the header of a frame is received, the number of received packets in the frame (PSI5SPRCF1(0-7).F(1-6)PKT) is referenced according to the channel ID and frame ID in the header. The number of packets to be received is determined from the number of already received packets. When FrmID is not 0 in ch0, the number of packets to be received is 6 (maximum value). When FrmID is an illegal value (6 or 7) in ch1 to ch7, the number of packets to be received is 6 (maximum value).

If the number of packets to be received is too small or too large, a reception error occurs. For details, see **Section 25.6.2.7, Abnormal Reception (Reception Error)**.

If too few packets are received, the received packets are not stored.

If a WDT error occurs, the received packets at the time of the WDT error occurrence are stored.

If too many packets are received, the received packets over the specified number are not stored.

Table 25.92 below shows the correspondence between the settings of PSI5SPRCF1(0-7).F(1-6)PKT and the number of packets to be received.

Table 25.92 PSI5 Frame Reception Packet Number

Value of PSI5SPRCF1(0 to 7).F(1 to 6)PKT	Specification
0	Mask the corresponding frame. (Receive data is not stored in SFR)
1, 2	Setting prohibited
3 to 6	Packet number is PSI5SPRCF1(0-7).F(1-6)PKT value(3 to 6)
7	Setting prohibited

The following figure shows the data format of a PSI5 frame when the payload bit length is min/max:

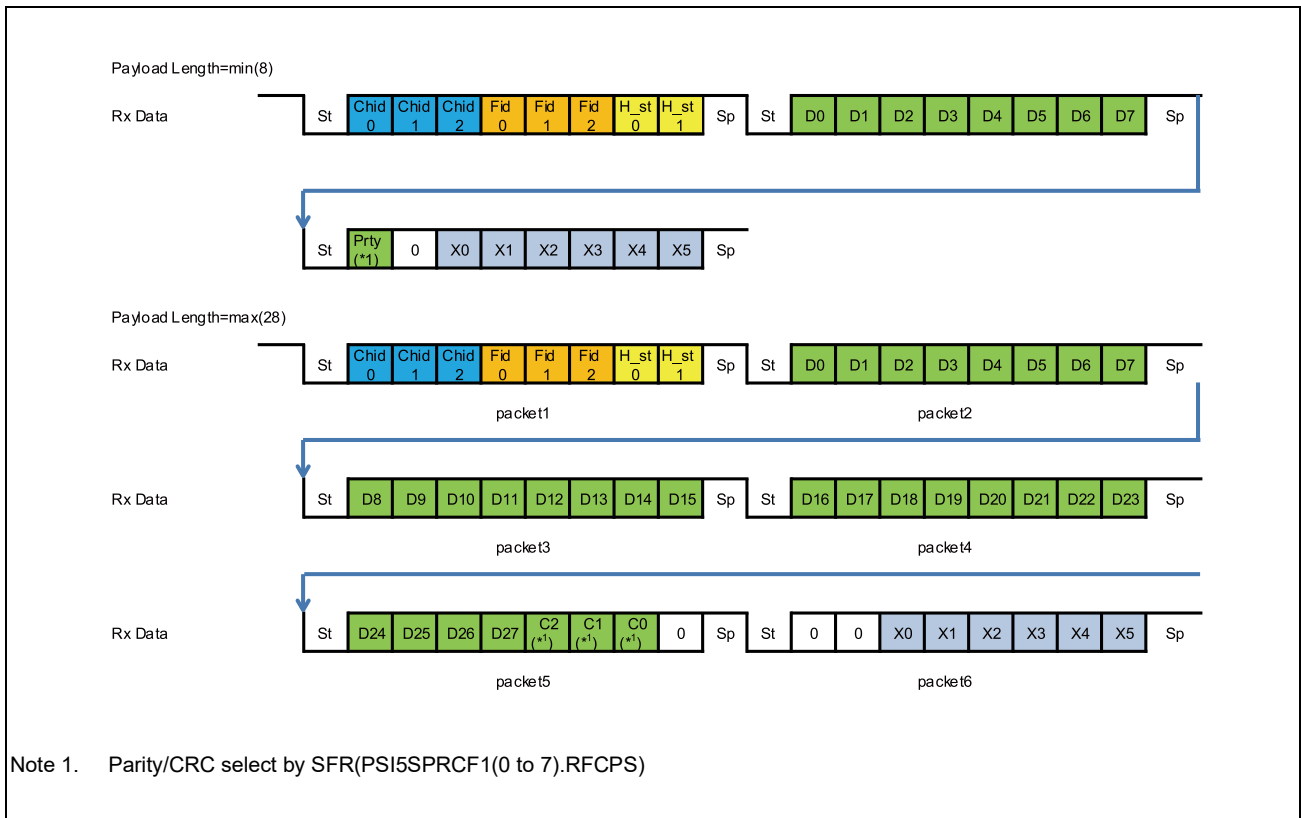


Figure 25.57 PSI5 Frame Data Format (Payload = 8 Bits (min.) to 28 Bits (max.))

(2) Restoring PSI5 Frame Data and Determining Errors

After receiving all the packets to be received, PSI5-S finishes packet frame reception and restores the PSI5 frame data.

For how to determine errors when restoring the PSI5 frame data, see **Section 25.6.2.7, Abnormal Reception (Reception Error)**.

(3) Generating MailBox Storage Data

After determining errors in PSI5 frame data, PSI5-S generates three types of mailbox storage data (triplet).

The three types of mailbox storage data contain the status, data, and timestamp, respectively, and have the same value of data consistency indicator (DCI) in the upper four bits.

The DCI value installed for each channel is generated by a 4-bit up-counter. The up-counter value is incremented each time PSI5 data is restored. PSI5-S is initialized by an asynchronous reset or software reset.

The following figure shows the timing of generating the DCI value.

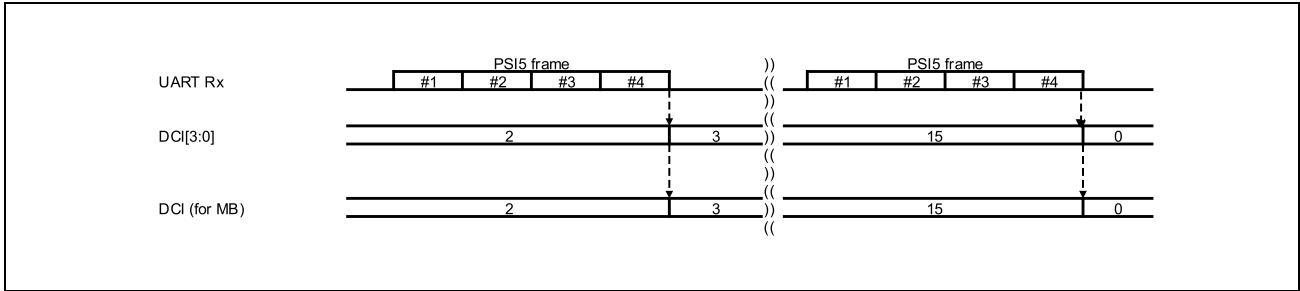


Figure 25.58 DCI Generation Timing

Figure 25.59 below shows the bit format of the mailbox data (triplet).

For details, see **Section 25.3.12, Ch 0 Frm m MB Data (m: 1, 2)** or **Section 25.3.13, Ch n Frm m MB Data (n: 1 to 7) (m: 1 to 6)**.

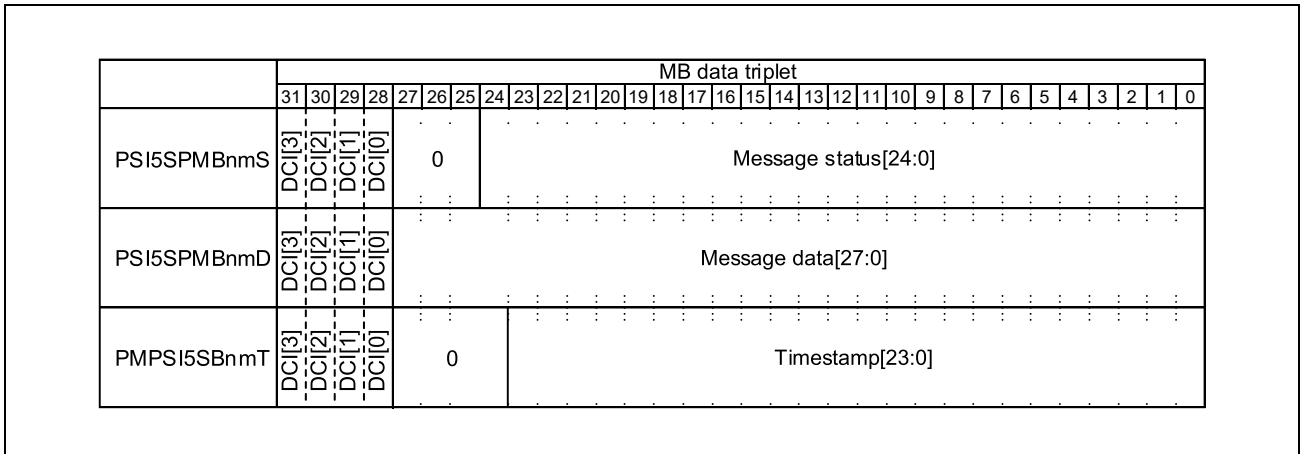


Figure 25.59 MB Data Triplet

(4) Writing MailBox Data to SFR

After generating mailbox storage data, PSI5-S obtains the storage destination of the mailbox data in SFR by using the channel ID and frame ID in the header.

The conditions for deciding the storage destination are as follows.

Channel 0 is a virtual channel. If an XCRC error or UART reception error occurs, the received frame is determined to be an invalid frame, and the mailbox data is stored in frames on channel 0. The mailbox data is also stored in frames on channel 0 when the frame ID is invalid or a WDT error occurs.

Table 25.93 PSI5 Frame MB Target Generate (Sync)

XCRC error, UART parity error, UART framing error	FrmID	MB target
No	0≤SFR data≤5	Ch: Receive ChID Frm: Receive FrmID
	6, 7	Ch: 0
Yes	Not related	Frm: 2

The storage data and the storage destination when a WDT error occurs in synchronous mode are as shown in **Table 25.98**.

Table 25.94 below lists the 1st addresses of the frames to write mailbox data that is generated from the received channel ID and frame ID.

Table 25.94 MB write 1st Address Generate (by Frame ID)

Frm ID	Frame	Write 1st Address							
		Ch0	Ch1	Ch2	Ch3	Ch4	Ch5	Ch6	Ch7
0	Frame1	500 _H	548 _H	590 _H	5D8 _H	620 _H	668 _H	6B0 _H	6F8 _H
1	Frame2	50C _H	554 _H	59C _H	5E4 _H	62C _H	674 _H	6BC _H	704 _H
2	Frame3	50C _H	560 _H	5A8 _H	5F0 _H	638 _H	680 _H	6C8 _H	710 _H
3	Frame4	50C _H	56C _H	5B4 _H	5FC _H	644 _H	68C _H	6D4 _H	71C _H
4	Frame5	50C _H	578 _H	5C0 _H	608 _H	650 _H	698 _H	6E0 _H	728 _H
5	Frame6	50C _H	584 _H	5CC _H	614 _H	65C _H	6A4 _H	6EC _H	734 _H
6	Error frame	50C _H							
7	Error frame	50C _H							

The mailbox data storage addresses must be in the range from 500_H to 73C_H shown in the address map (**Table 25.11**).

When mailbox data is stored in frame 3 on channel 1, the 1st address of the storage destination is 560_H.

After deciding the storage destination, PSI5-S stores the status in the area beginning with the 1st address, data in the area beginning with the address “1st address + 4”, and timestamp in the area beginning with the address “1st address + 8”.

(5) Monitoring PSI5 Frame Reception with the WDT

With the WDT installed for each channel, PSI5-S monitors reception of the specified number of PSI5 frames to be received.

When the synchronization pulse is transmitted to the Tx shifter register after synchronization pulse input from GTM, the WDT counter fetches a specified expiration value and starts counting. When the specified number of frames are received or the WDT counter value becomes 0, the WDT counter stops, and waits for the rising edge of the next synchronization pulse. At the rising edge of the next synchronization pulse, the WDT counter fetches a specified expiration value, and starts counting. If the WDT counter value becomes 0, a WDT error is assumed. For details, see **Section 25.6.2.8, Abnormal Reception (WDT Error)**.

The WDT expiration value (PSI5SPWDEVn.WDTEX) should be shorter than the interval time of synchronization pulse.

NOTE

Frame reception time < WDT expiration time < {(the interval time of synchronization pulse) - (maximum arbitration delay time)}

(maximum arbitration delay time) = bit time * 154 + 50 * (PCLK period)

The following figure shows the WDT operation (without WDT error).

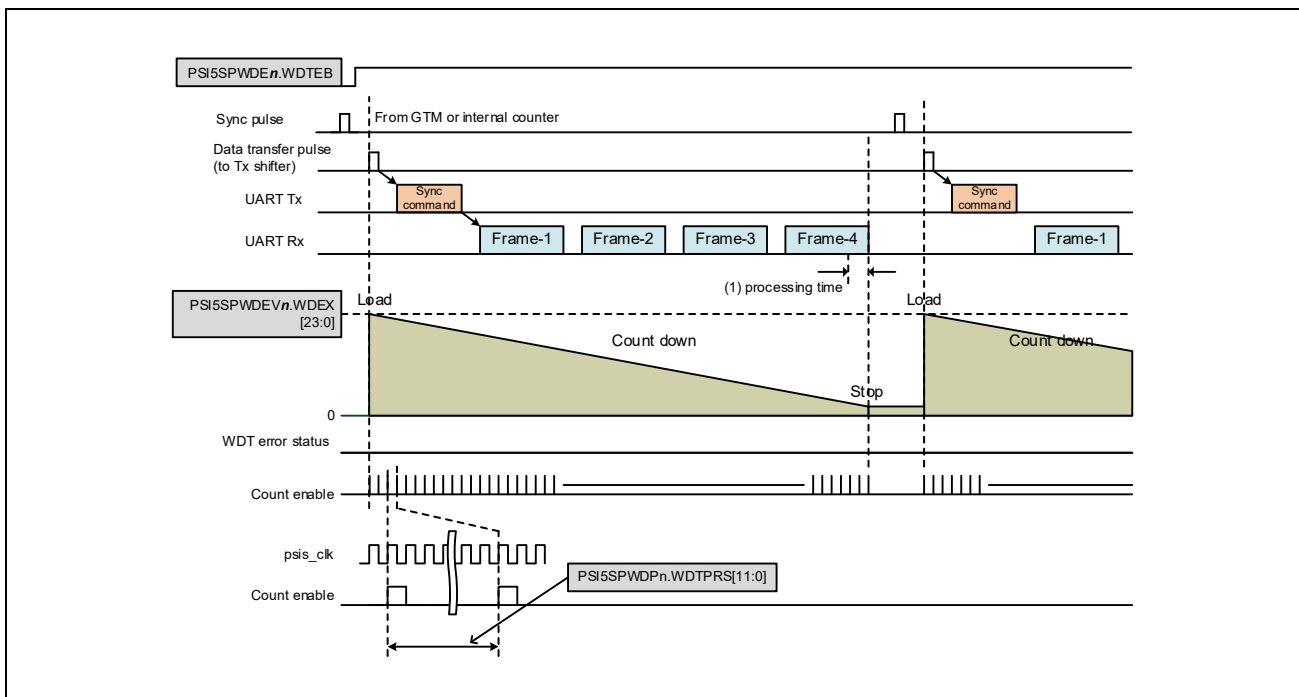


Figure 25.60 WDT Operation Timing (Sync Mode) (without WDT Error)

NOTE

The WDT counter stops in 50 cycles after a stop bit of the last frame is received. A WDT error occurs in a case that the WDT counter expires within the 50 clock cycles even if the stop bit of the last frame is received.

The number of packets to be received on each channel is obtained as follows:

- When the value set in PSI5SPRCF1(0-7).F(1-6)PKT is not 0, the frame is monitored.
- When the value set in PSI5SPRCF1(0-7).F(1-6)PKT is 0, the frame is not monitored.

25.6.2.2 Normal Reception (in Asynchronous Mode)

Unlike synchronous communication, asynchronous communication does not transfer PSI5 frames in synchronization with synchronizing pulses. In asynchronous mode, PSI5-S receives PSI5 frames from the transceiver according to the packets transmitted from the sensor.

(1) Receiving UART Frames

This operation is the same as that described in **Section 25.6.2.1(1), Receiving UART Frames**.

(2) Restoring PSI5 Frame Data and Determining Errors

This operation is the same as that described in **Section 25.6.2.1(2), Restoring PSI5 Frame Data and Determining Errors**.

(3) Generating MailBox Storage Data

This operation is the same as that described in **Section 25.6.2.1(3), Generating MailBox Storage Data**.

(4) Writing MailBox Data to SFR

This operation is different from that described in **Section 25.6.2.1(4), Writing MailBox Data to SFR** in the following points.

After generating mailbox storage data, PSI5-S obtains the storage destination of the mailbox data by using the channel ID in the header and the frame ID counter*1 installed for each channel.

In asynchronous mode, the frame ID is fixed to 0. Therefore, the frame ID is replaced with the frame ID counter in order to always store the data of the six latest frames.

Note 1. The frame ID counter is a hexadecimal counter that operates only in asynchronous mode and is incremented each time mailbox data is stored in SFR.

The following figure shows an outline of the counter operation:

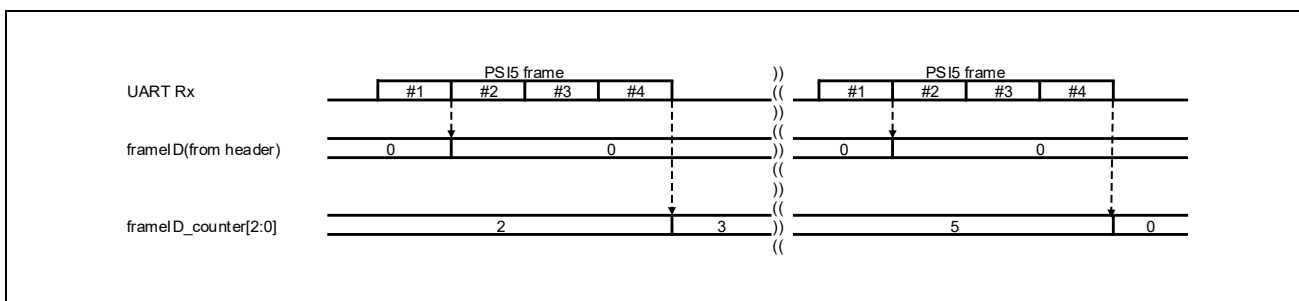


Figure 25.61 Frame ID Counter (Asynchronous Mode Only)

Table 25.95 below shows the conditions for obtaining the storage destination.

Table 25.95 PSI5 Frame MB Target Generate (Async)

XCRC error, UART parity error, UART framing error	FrmID	MB target
No	= 0	Ch: Receive ChID Frm: Frame ID counter data (Active Asynchronous mode)
	≠ 0	Ch: 0
Yes	Not related	Frm: 2

The storage data and the storage destination when a WDT error occurs in asynchronous mode are as shown in **Table 25.99**.

(5) Monitoring PSI5 Frame Reception with the WDT

With the WDT installed for each channel, PSI5-S checks that PSI5 frames are received frame by frame.

When the WDT enable bit (PSI5SPWDE(0-7).WDTEB) is set to 1 in PSI5S mode, the WDT counter fetches a specified expiration value and starts counting. Each time a frame is received or the WDT counter value becomes 0, the WDT counter fetches the specified expiration value again and starts counting.

If the WDT counter value becomes 0, occurrence of a WDT error is assumed. For details, see **Section 25.6.2.8, Abnormal Reception (WDT Error)**.

The following figure shows the WDT operation (without a WDT error).

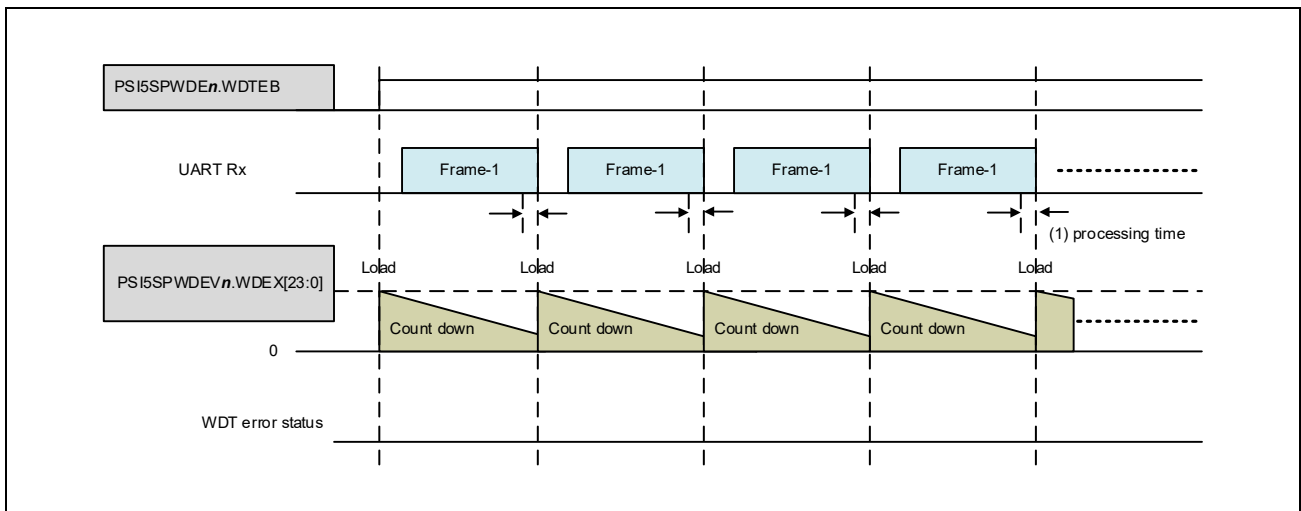


Figure 25.62 WDT Operation Timing (Asynchronous Mode) (without WDT Error)

NOTE

The WDT counter restart in 50 cycles after a stop bit of the frame is received. A WDT error occurs in a case that the WDT counter expires within the 50 clock cycles even if the stop bit of the frame is received.

25.6.2.3 Normal Transmission (Command Data)

After transmitting data (PSI5SPTFD1 and PSI5SPTFD2) and the number of transmit packets (PSI5SPTFNM.TXNUM) is set, PSI5-S transmits UART frames according to the specified number of transmit packets when the transmission start bit (PSI5SPTFST.TXST) is set. No idle time is set between UART frames. To maintain continuous transmission, packet frames are transmitted without idle time between them.

For the UART specifications, see **Section 25.2.1, Functional Overview**.

For 8-byte communication, data is transmitted in order of PSI5SPTFD1.TDT1[7:0], PSI5SPTFD1.TDT2[7:0], ..., and PSI5SPTFD2.TDT8[7:0].

The following figure shows the operation of command data transmission.

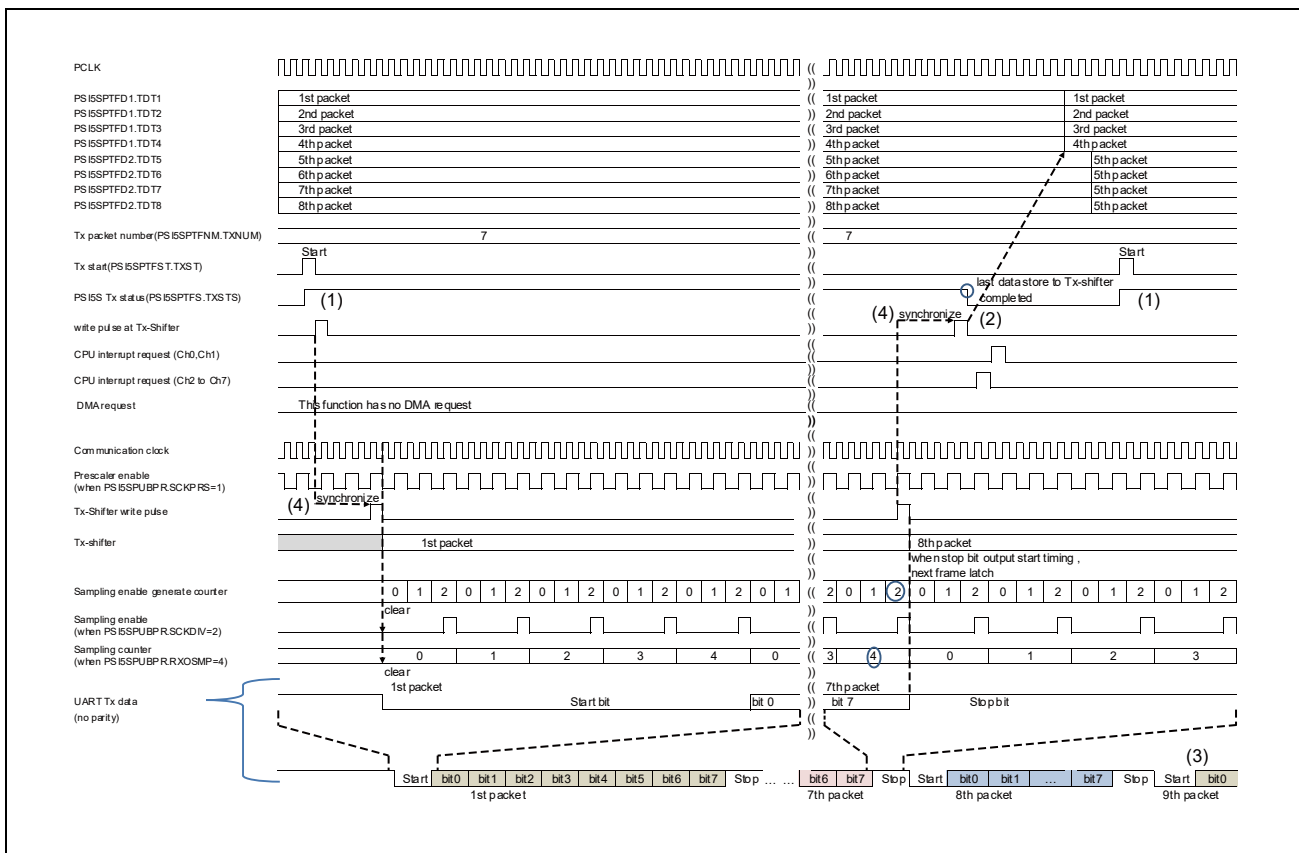


Figure 25.63 Command Data Transmission (8-byte)

- (1) The PSI5SPTFS.TXSTS bit is 1 when TXST is 1.
- (2) In the timing of stop bit output, next packet data is set in the transmit data shift register (Tx shifter). The PSI5SPTFS.TXSTS bit is cleared to 0 to enable writing of the next data by the CPU.
- (3) If the CPU writes transmit data continuously, packets are transmitted without idle time.
- (4) The asynchronous transfer processing based on PCLK and the communication clock causes a delay of several clock cycles.

25.6.2.4 Normal Transmission (ECU-to-Sensor Data)

In PSI5S mode, PSI5-S transmits a Sync command in synchronization with the synchronization pulse generated by the GTM or an internal counter. Because the initial value of ECU-to-sensor data is set to all 1's, the output value of the Sync command is 1 (command data (PSI5SPTCD n (n : 1-7).ATRSCMD[4:0] and PSI5SPTCD n (n : 1-7).ACHID[2:0])) unless the transmit data is updated.

After the frame type (frame 1 to frame 4) of ECU-to-sensor data and the ECU-to-sensor data (maximum data size: 20 bits, address: 4 bits) to be transmitted are set, PSI5-S generates frame data by adding CRC bits at the end of ECU-to-sensor data. Note that all unused bits for ECU-to-sensor data must be 1.

The generating polynomial for CRC is “ $g(x) = x^6 + x^4 + x^3 + 1$ ” (initial value: 010101_B) when the number of CRC bits is 6, or “ $g(x) = x^3 + x + 1$ ” (initial value: 0b111) when the number of CRC bits is 3. The range of CRC calculation is zero extension of the number of CRC bit in the MSB of ECU-to-sensor data (maximum of 24 bits). Data bits are transmitted LSB first.

Because the channels set to asynchronous mode do not use synchronization pulses, ECU-to-sensor transmission is not performed on those channels.

Table 25.96 below shows the frame types available for ECU-to-sensor data.

Table 25.96 ECU-to-Sensor Frame Types

Value of PSI5SDDSRTYPE[1:0]	Specification	Start Bit	CRC	CRC Calculation Scope
00 _B	Frame1 (Short) Frame length = 15 ECU to Sensor data = data[2:0],address[2:0]	010 _B	3 bits x3+x+1	000 _B , data[2:0], address[2:0] (9bit)
01 _B	Frame2 (Long) Frame length=29 ECU to Sensor data = data[12:0], address [2:0]	010 _B	3 bits x3+x+1	000 _B , data[12:0], address [2:0] (19bit)
10 _B	Frame3 (XLong) Frame length = 37 ECU to Sensor data = data[18:0], address[2:0]	010 _B	3 bits x3+x+1	000 _B , data[18:0], address [2:0] (25bit)
11 _B	Frame4 (XXLong) Frame length = 43 ECU to Sensor data = data[19:0], address[3:0]	011111110 _B	6 bits x6+x4+x3+1	000000 _B , data[19:0], address[3:0] (30bit)

The following figure shows the format of frame data.

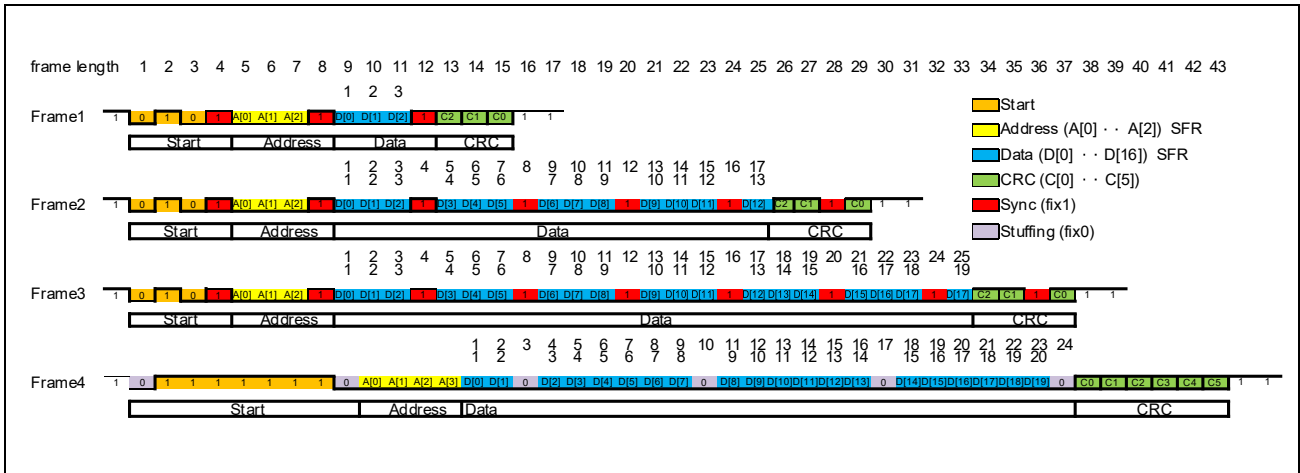


Figure 25.64 ECU-to-Sensor Data Format

After ECU-to-sensor data is generated, the data shifts bit by bit beginning with the start bit at the rising edge of a synchronization pulse for each channel.

Based on the value output by shifting, the command data in SFR is selected (from “PSI5SPTCDn(n: 1-7).ATRSCMD[4:0] and one bit of PSI5SPTCDn(n: 1-7).ACHID[2:0]” and “PSI5SPTCDn(n: 1-7).TRSCMD[4:0] and one bit of PSI5SPTCDn(n: 1-7).CHID[2:0]”).

When the value output by shifting is 0, the command data in SFR is “PSI5SPTCDn(n: 1-7).TRSCMD[4:0] and one bit of PSI5SPTCDn(n: 1-7).CHID[2:0]”.

When the value output by shifting is 1, the command data in SFR is “PSI5SPTCDn(n: 1-7).ATRSCMD[4:0] and one bit of PSI5SPTCDn(n: 1-7).ACHID[2:0]”.

A UART frame is generated from the selected data, and output.

The following figure shows the format of the UART frame to be output.

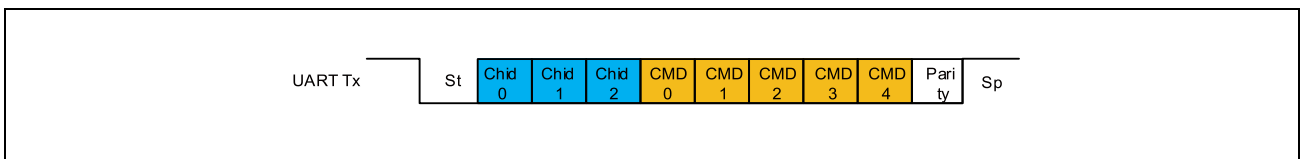
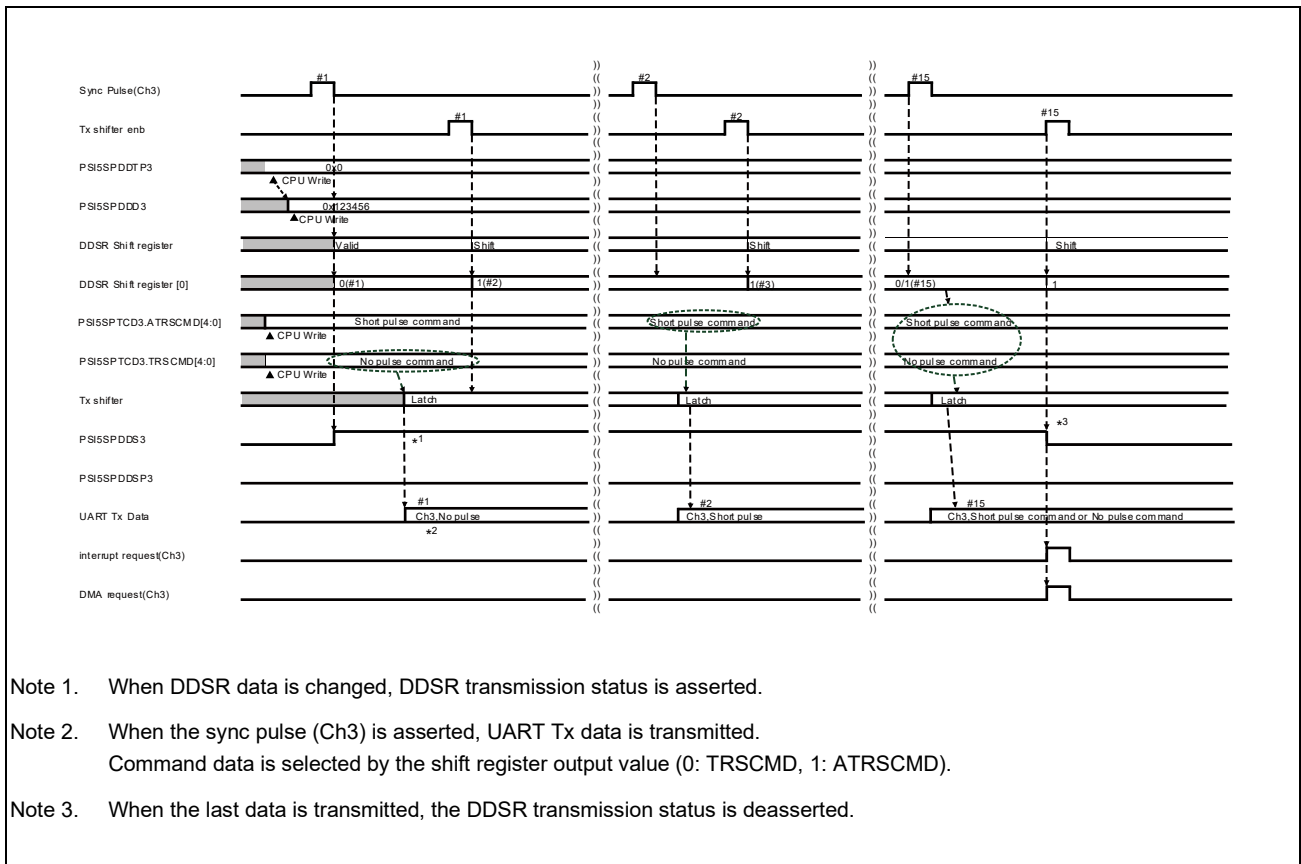


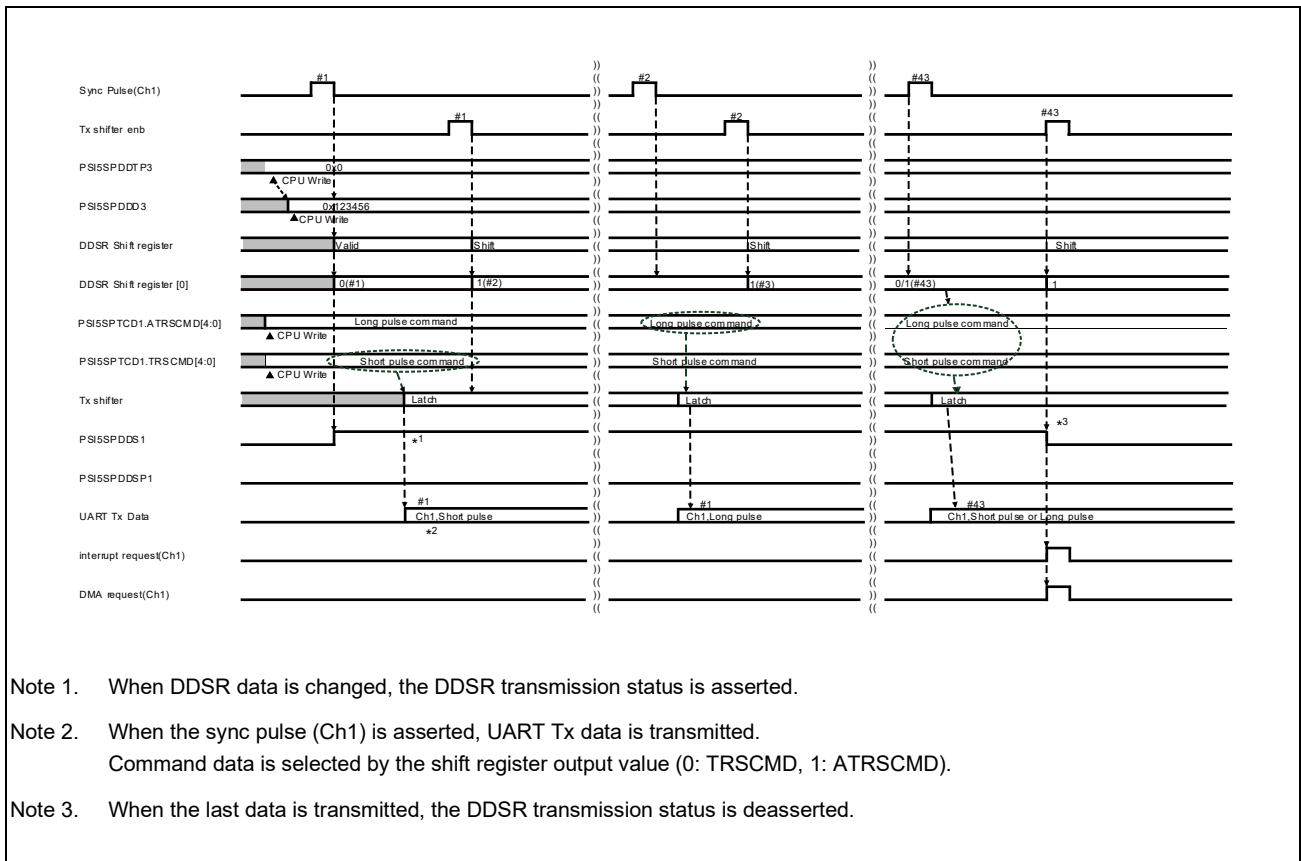
Figure 25.65 ECU-to-Sensor Data Format (UART Frame)

The following figure shows the operation of ECU-to-sensor data transmission.



- Note 1. When DDRSR data is changed, DDRSR transmission status is asserted.
- Note 2. When the sync pulse (Ch3) is asserted, UART Tx data is transmitted.
Command data is selected by the shift register output value (0: TRSCMD, 1: ATRSCMD).
- Note 3. When the last data is transmitted, the DDRSR transmission status is deasserted.

Figure 25.66 ECU-to-Sensor Data Transmission (Frame 1)



- Note 1. When DDSR data is changed, the DDSR transmission status is asserted.
- Note 2. When the sync pulse (Ch1) is asserted, UART Tx data is transmitted.
Command data is selected by the shift register output value (0: TRSCMD, 1: ATRSCMD).
- Note 3. When the last data is transmitted, the DDSR transmission status is deasserted.

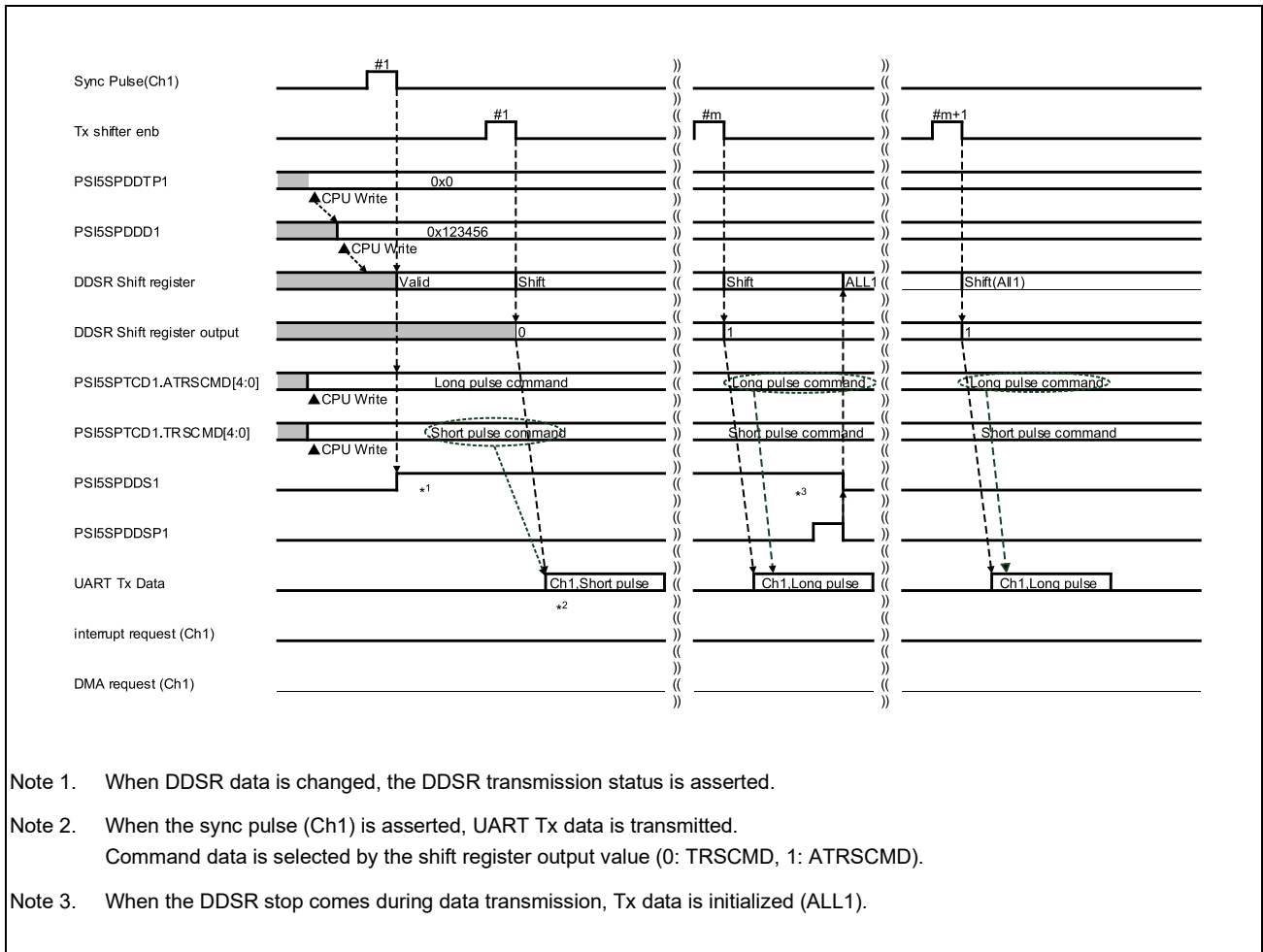
Figure 25.67 ECU-to-Sensor Data Transmit (Frame 4)

25.6.2.5 Normal Transmission (ECU-to-Sensor Data Transmission Stopped)

If transmission is stopped (PSI5SPDDSP(1-7).DDSRSTP = 1) during ECU-to-sensor data transmission, PSI5-S performs the following operations:

- Initializing the DDSR shift register (set all 1)
- De-asserting the DDSR transmission status signal

The following figure shows the operation that is performed when ECU-to-sensor data transmission is stopped.



- Note 1. When DDSR data is changed, the DDSR transmission status is asserted.
- Note 2. When the sync pulse (Ch1) is asserted, UART Tx data is transmitted.
Command data is selected by the shift register output value (0: TRSCMD, 1: ATRSCMD).
- Note 3. When the DDSR stop comes during data transmission, Tx data is initialized (ALL1).

Figure 25.68 ECU-to-Sensor Data Transmission Stop

25.6.2.6 Normal Transmission (Transmit Request Arbitration)

Transmit requests are arbitrated by using the function shown in **Figure 25.45** and output by UART transmission.

The priority of arbitration is: Ch1 (high priority) > Ch2 > Ch3 > Ch4 > Ch5 > Ch6 > Ch7 > Command data (low priority)

The following figure shows an operation in which requests are input after the lower priority requests are input.

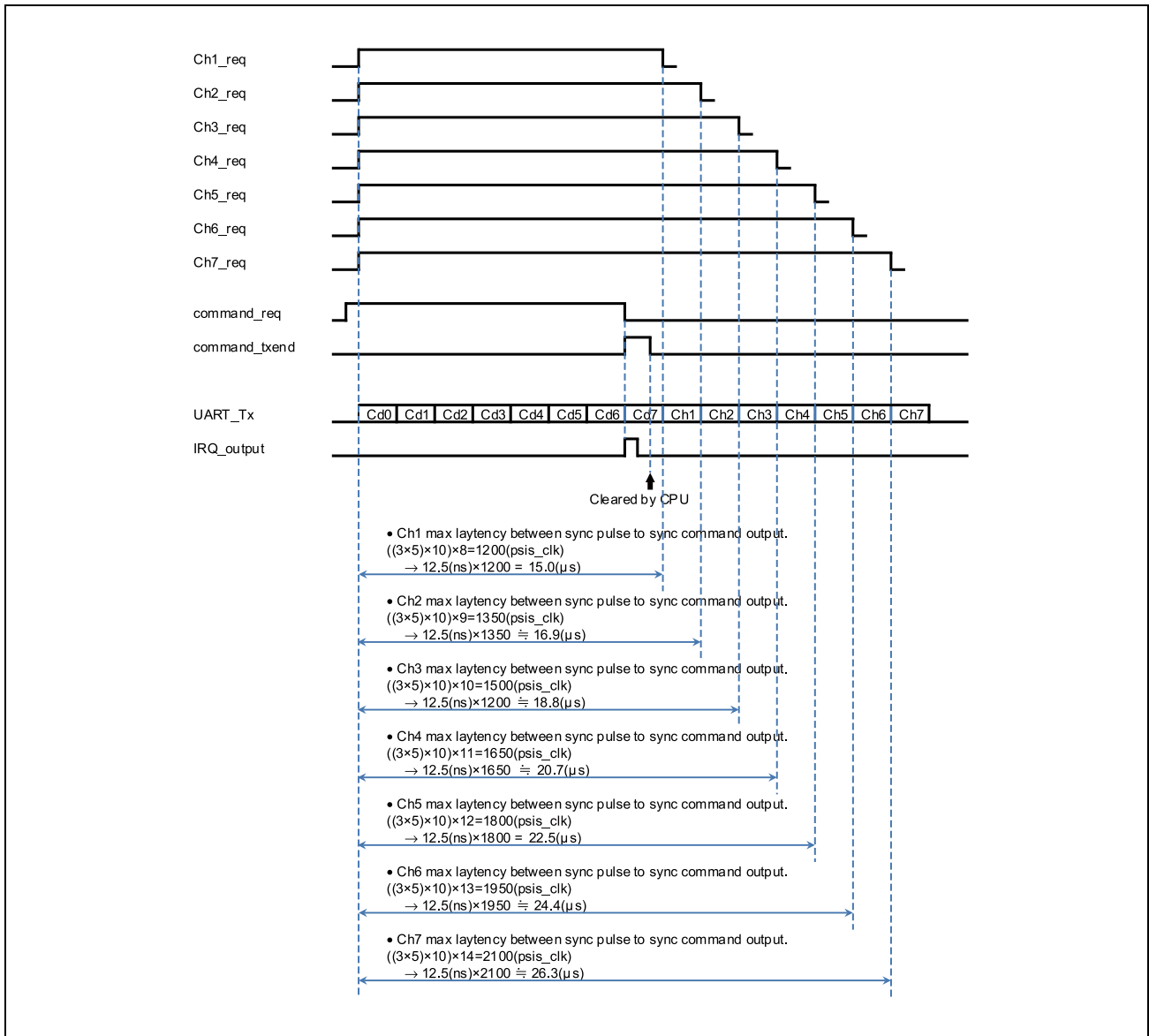


Figure 25.69 Arbitration Tx Request (max latency)

The following figure shows an operation in which all requests are input simultaneously.

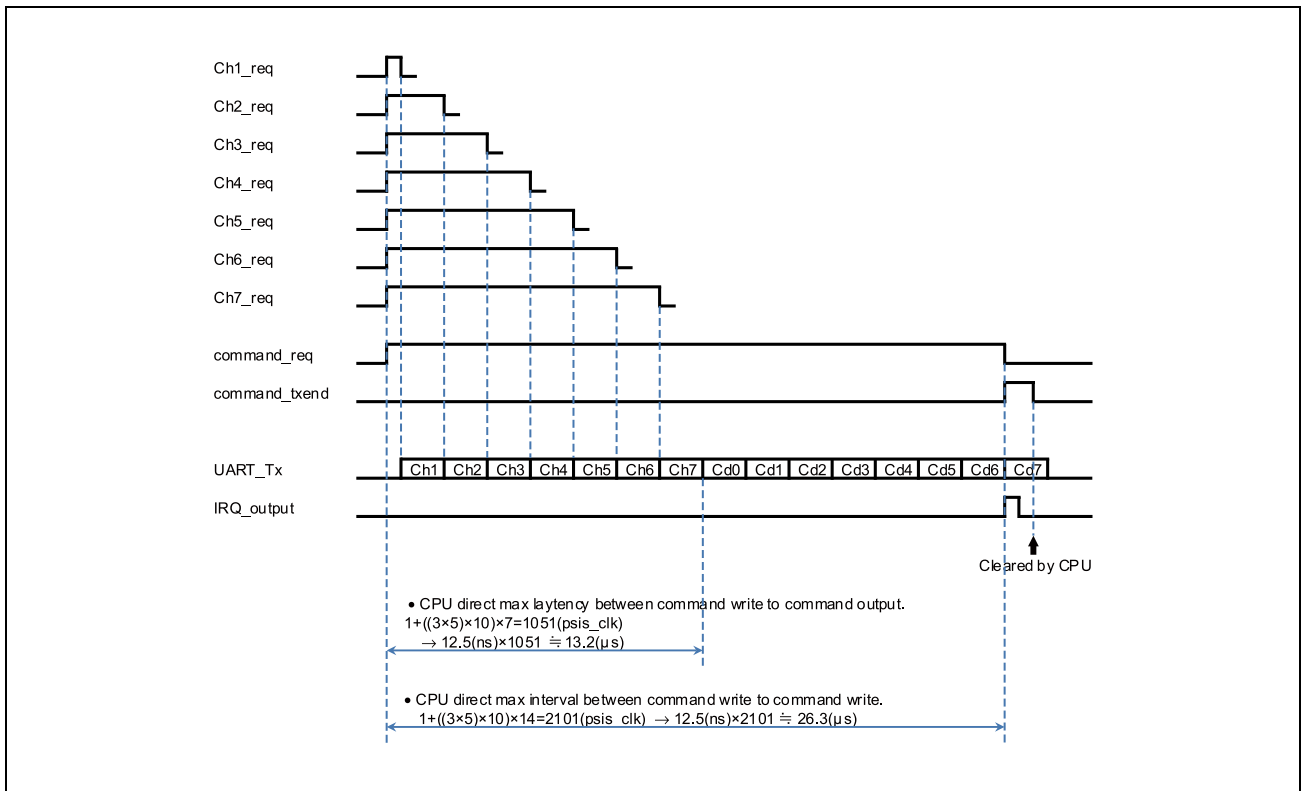


Figure 25.70 Arbitration Tx Request (All Requests Share the Same Timing)

Tx request in FIFO is cleared when the operation enable register (PSI5SPUOEB.OPEN) is set to 0.

25.6.2.7 Abnormal Reception (Reception Error)

After restoring a PSI5 frame, PSI5-S determines whether reception errors have occurred. Then, PSI5-S checks for XCRC errors, CRC errors, transceiver status errors, and mailbox overrun.

If mailbox overrun has occurred, the data in the mailbox will be unpredictable.

Other errors are checked for at different timings. The timing for determining other errors is as follows:

UART-related errors (parity and framing errors) are checked for when a UART frame is received.

A WDT error is determined to have occurred when the WDT counter value becomes 0. For details, see **Section 25.6.2.8, Abnormal Reception (WDT Error)**.

An error status is cleared by writing 1 to the corresponding bit in the reception status clear register. The error status is also cleared when the operation mode is switched to configuration mode.

Table 25.97 below lists PSI5 frame reception errors.

Table 25.97 PSI5 Frame Reception Errors

Error	Detect Timing	Detection Condition
XCRC	PSI5 frame received	The agreement of XCRC (6-bit) value which was calculated from received data and the reception XCRC (6-bit) value is confirmed.
CRC	PSI5 frame received & RFCPS=1	The agreement of CRC (3-bit) value which was calculated from received data and the reception CRC (3-bit) value is confirmed.
Parity	PSI5 frame received & RFCPS=0	The agreement of even parity value which was calculated from received data and the reception parity value is confirmed.
Transceiver status	PSI5 frame received & error in header≠0x0	When the error status value of the received header data is other than 0.
UART parity	UART frame received	The received parity value does not match.
UART framing	UART frame received	The received stop bit value is 0.
WDT	WDT counter value reaches 0	Same as timing of detection
MB overrun	PSI5 frame received	The previous data has not been read.
PSI5 frame lack	IDLE time is passed*1	The number of received packets is less than the number of packets to be received.
PSI5 frame excess	UART frame received*2	The number of received packets exceeds the number of packets to be received.

Note 1. When a PSI5 frame lack error is occurs, the new PSI5 frame data is not stored.
Then if a WDT error occurs, the received packets at the time of the WDT error occurrence is stored.

Note 2. When a PSI5 frame excess error is occurred, the extra data of the new PSI5 frame is not stored.

When the number of received packets exceeds the number of packets to be received, a PSI5 frame excess error, XCRC error and CRC error occur. In this case, a PSI5 frame excess error is detected for the received Ch, and the received data is stored as Ch 0 Frm 2 MB data.

The generating polynomial for XCRC is “ $g(x) = x^6 + x^4 + x^3 + 1$ ”. The initial value of the XCRC bits is 010101_B.

The generating polynomial for CRC is “ $g(x) = x^3 + x + 1$ ”. The initial value of the XCRC bits is 111_B.

The following figure shows the ranges of calculation for the XCRC and CRC values for a PSI5 frame that uses CRC and has a payload of 13 bits.

The calculation target of XCRC is PSI5 frame data excluding XCRC, and 0 extension of 6bits to MSB.

The calculation target of CRC is the payload data, and 0 extension of 3bits to MSB.

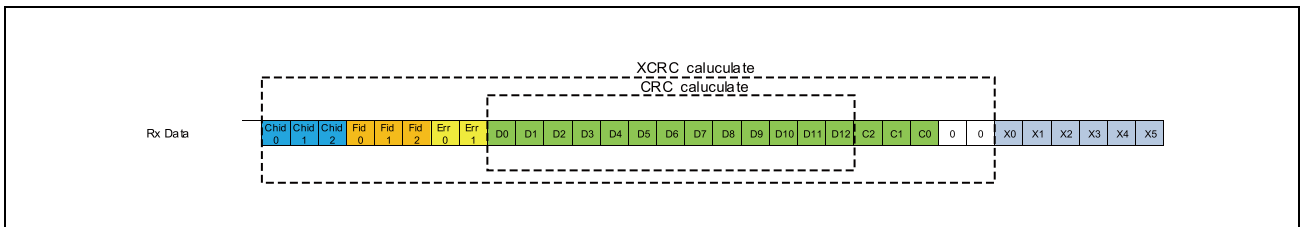
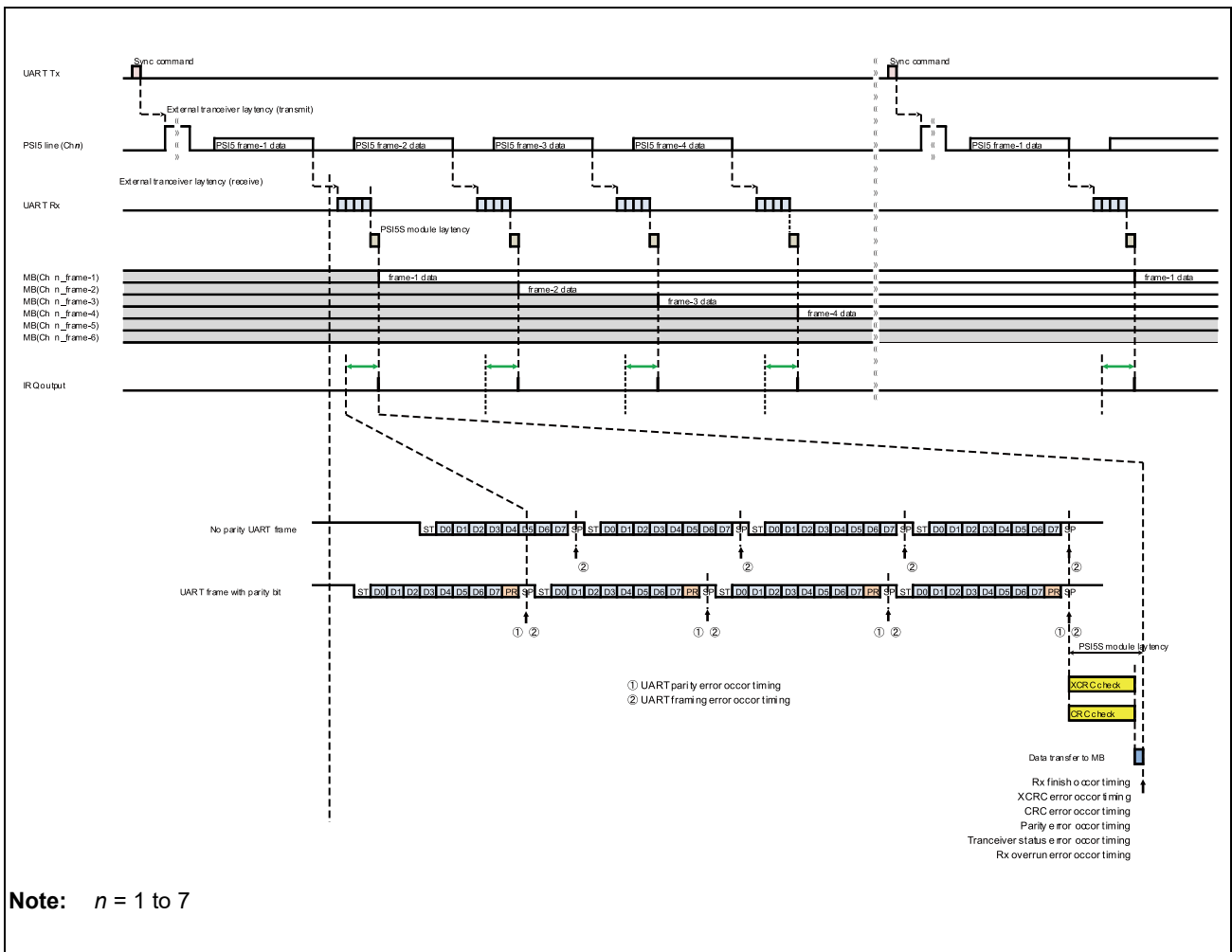


Figure 25.71 Calculate Area of XCRC, CRC

The following figure shows the detection timing of reception errors.



Note: n = 1 to 7

Figure 25.72 Sensor-to-ECU Data Is Received (Reception Error Is Detected)

The following figure shows the operation for error detection when too many or too few UART frames are received.

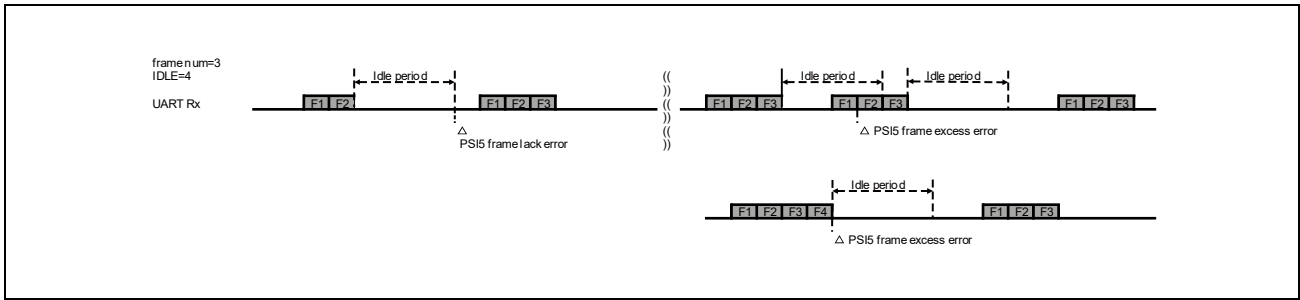


Figure 25.73 UART Frame Error Is Detected (Frame Lack or Frame Excess)

When the PSI5-S transitions to the PSI5-S during a PSI5 frame reception, any error of “UART framing error”, “UART parity error”, “payload data parity error”, “Frame lack error” and “packet frame XCRC error” possibly occurs. Because the PSI5-S doesn't detect a bus idle condition when the PSI5-S transitions to the PSI5S mode.

25.6.2.8 Abnormal Reception (WDT Error)

The synchronous and asynchronous modes differ from each other in WDT error determination, as follows.

(1) Synchronous Mode

When the synchronization pulse is transmitted to the Tx shifter after synchronization pulse input, the WDT value is reset to the expiration value, and, at the same time, the WDT starts the next operation. If the WDT counter reaches 0 before receiving all the packets to be received, a WDT error is assumed.

Note that the WDT starts operation again after the rising edge of the next synchronization pulse for each channel is detected.

If a WDT error occurs, an interrupt status (PSI5SPCIS[0:7].ISTRWDT) occurs. The interrupt status is cleared by writing 1 to the corresponding bit (PSI5SPCIS[0:7].ISTRWDT) in the interrupt status clear register.

The data received until the transmission of the synchronization pulse to the Tx shifter after a WDT error is detected is not stored.

The following figure shows the operation that is performed when a WDT error occurs in synchronous mode.

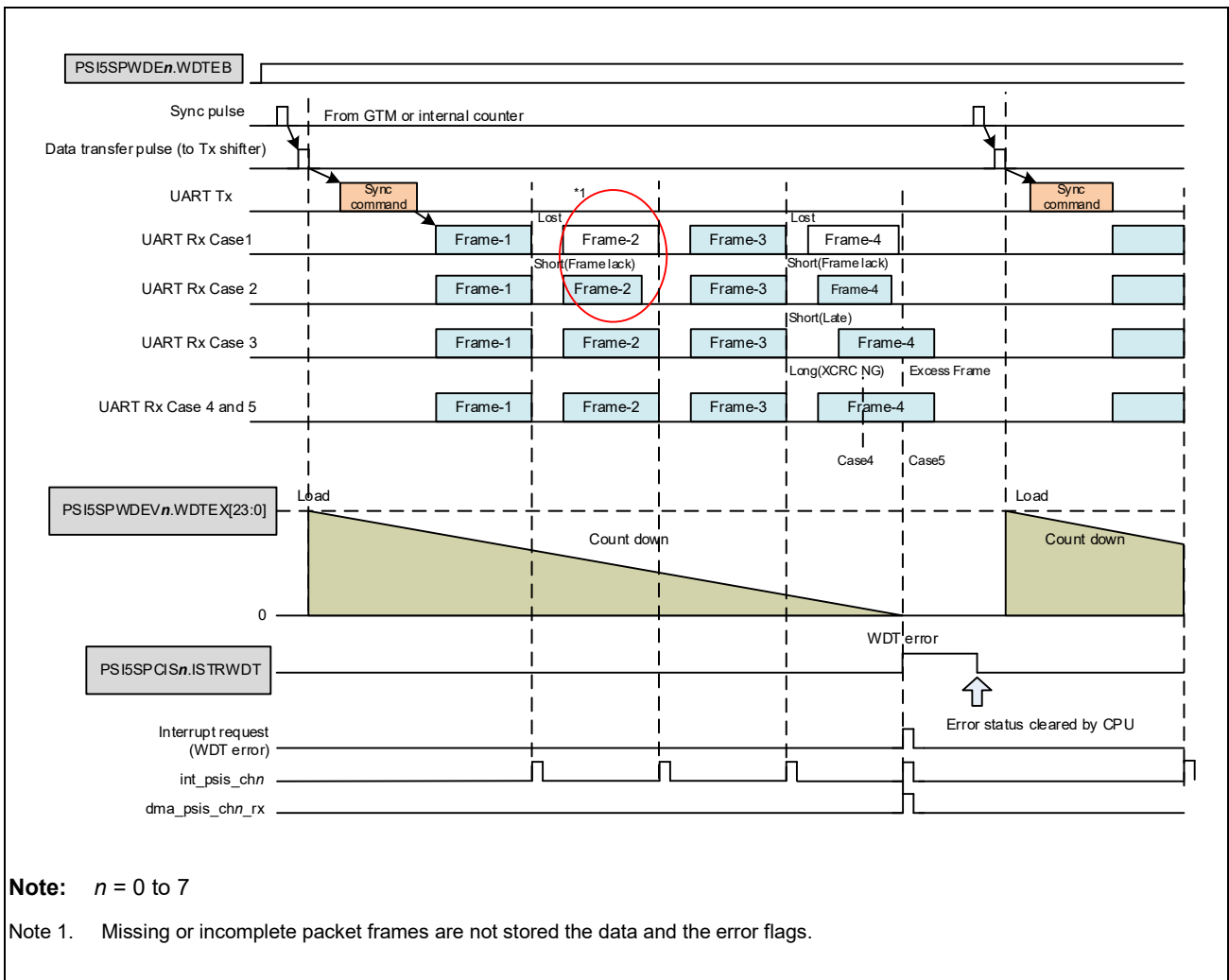


Figure 25.74 WDT Operation Timing (Sync Mode) — WDT Error

Below shows the MailBox triplet when a WDT error occurs in synchronous mode.

Table 25.98 MB Data Triplet - WDT Error (Synchronous Mode)

Case of UART Rx	Case1	Case2	Case3	Case4	Case5
Store location	ch0/frm2	ch0/frm2	ch0/frm2	ch0/frm2	ch0/frm2
DCI	Ch0	Ch0	Ch0	Ch0	Ch0
CHID	Channel where WDT error detected	Channel where WDT error detected	Channel where WDT error detected	CHID in the header of the reception frame	Channel where WDT error detected
FID	Packet Frame Counter of channel where WDT error detected	Packet Frame Counter of channel where WDT error detected	Packet Frame Counter of channel where WDT error detected	FrmID in the header of reception frame	Packet Frame Counter of channel where WDT error detected
MBORERR	MailBox overrun error	MailBox overrun error	MailBox overrun error	MailBox overrun error	MailBox overrun error
WDTERR	1	1	1	1	1
UTFRERR	0	UART framing error	UART framing error	UART framing error	UART framing error
UTPTERR	0	UART parity error	UART parity error	UART parity error	UART parity error
HEADERR	0	Header error	Header error	Header error	Header error
HEADST	0	HEADST in the header of the reception frame	HEADST in the header of the reception frame	HEADST in the header of the reception frame	HEADST in the header of the reception frame
CRCERR	0	0	0	CRC error	CRC error
CRC	000 _B	000 _B	000 _B	CRC of reception frame data	CRC of reception frame data
XCRCERR	0	0	0	1	1
XCRC	00_0000 _B	00_0000 _B	00_0000 _B	XCRC of reception frame data	XCRC of reception frame data
Payload data	000_0000 _H	Incomplete frame data *1,*2	Incomplete frame data *1,*2	Payload data of received frame data	Incomplete frame data *1,*2
Time stamp	At WDT error detection (Ch0)	At WDT error detection (Ch0)	At WDT error detection (Ch0)	At header reception (Ch0)	At WDT error detection (Ch0)

Note 1. Not received (missing or incomplete) packet bits are "0".

Note 2. Incomplete data is stored in packets 2 to 4 and the lower 4 bits of packet 5.

The following figure shows incomplete data format and store location.

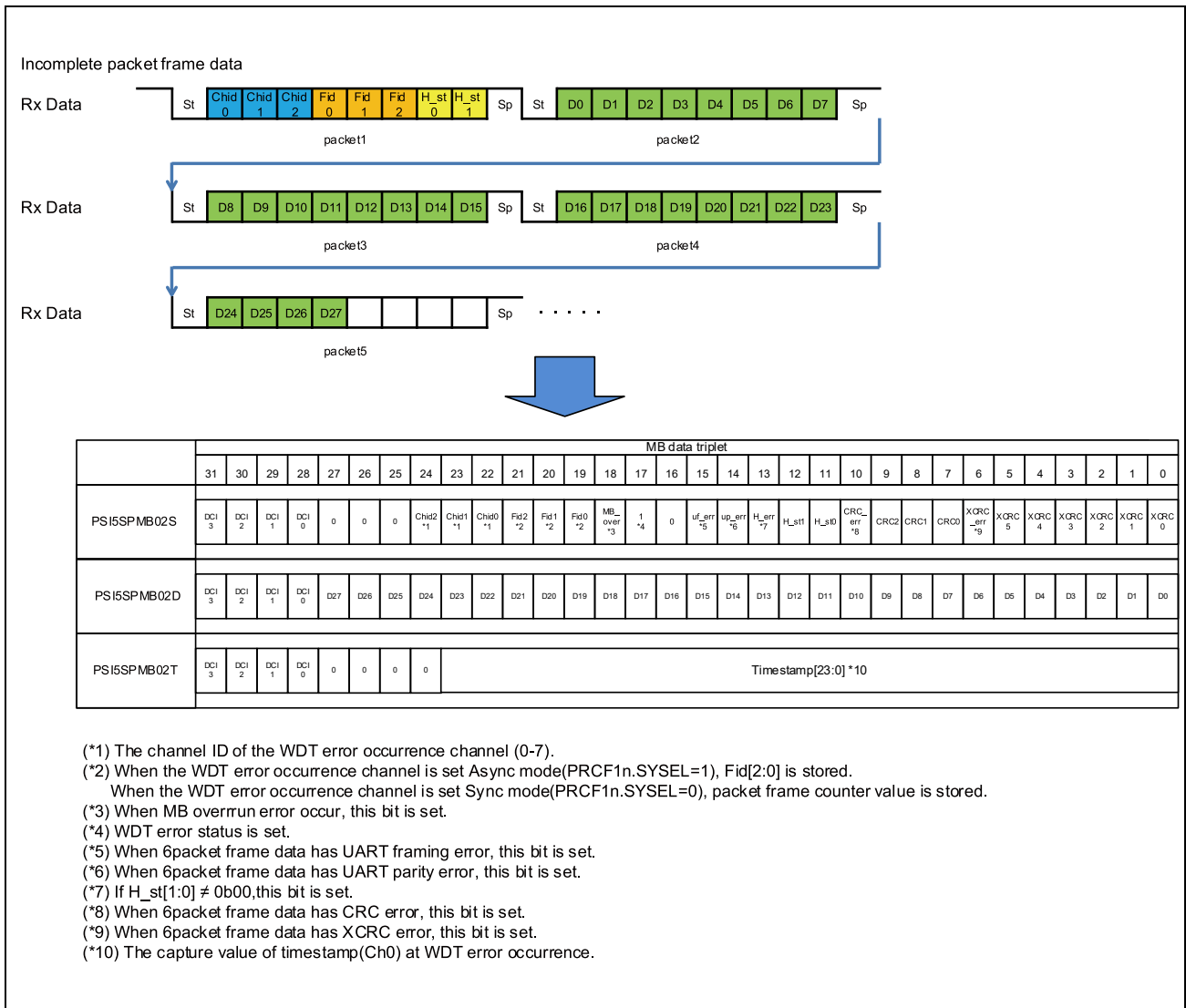


Figure 25.75 PSI5 Frame Data Format and MB Data Triplet at WDT Error

The following figure shows incomplete data format and store location.

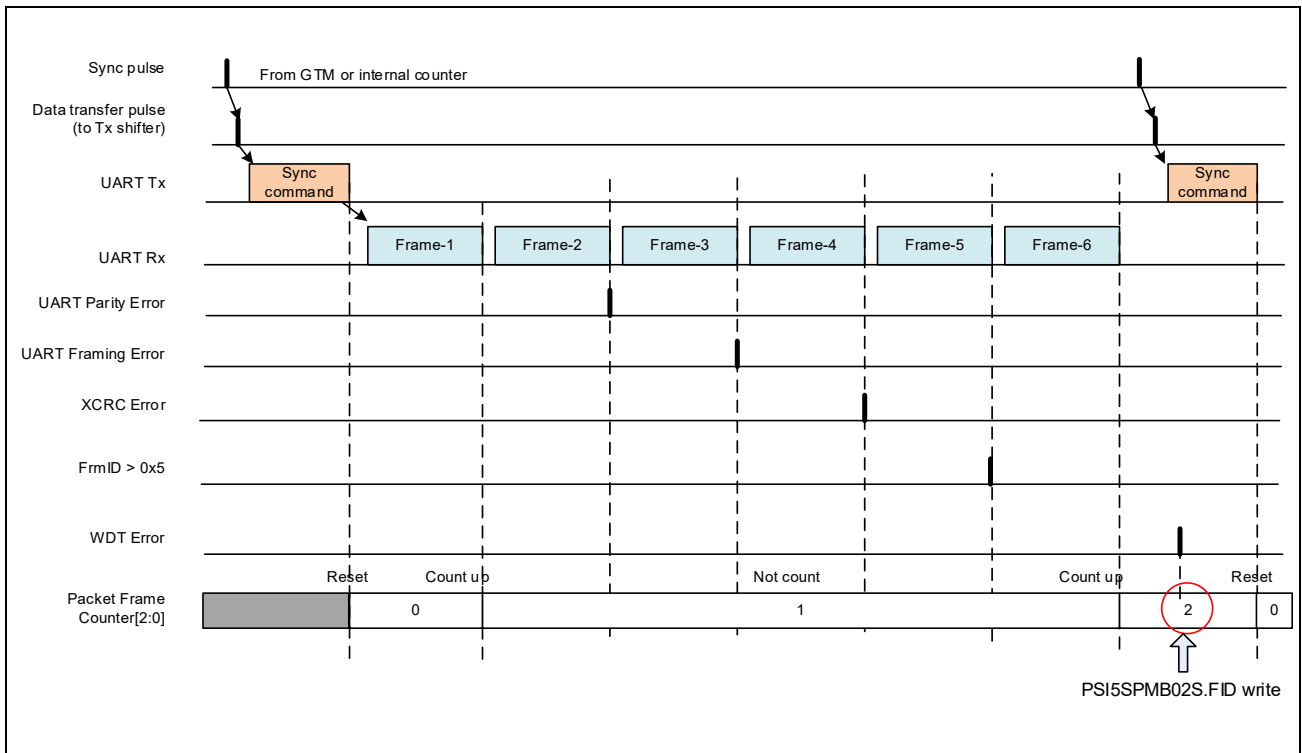


Figure 25.76 Packet Frame Counter

The following figure shows the arbitration of the write request to Ch0 and the operation of the interrupt and DMA request.

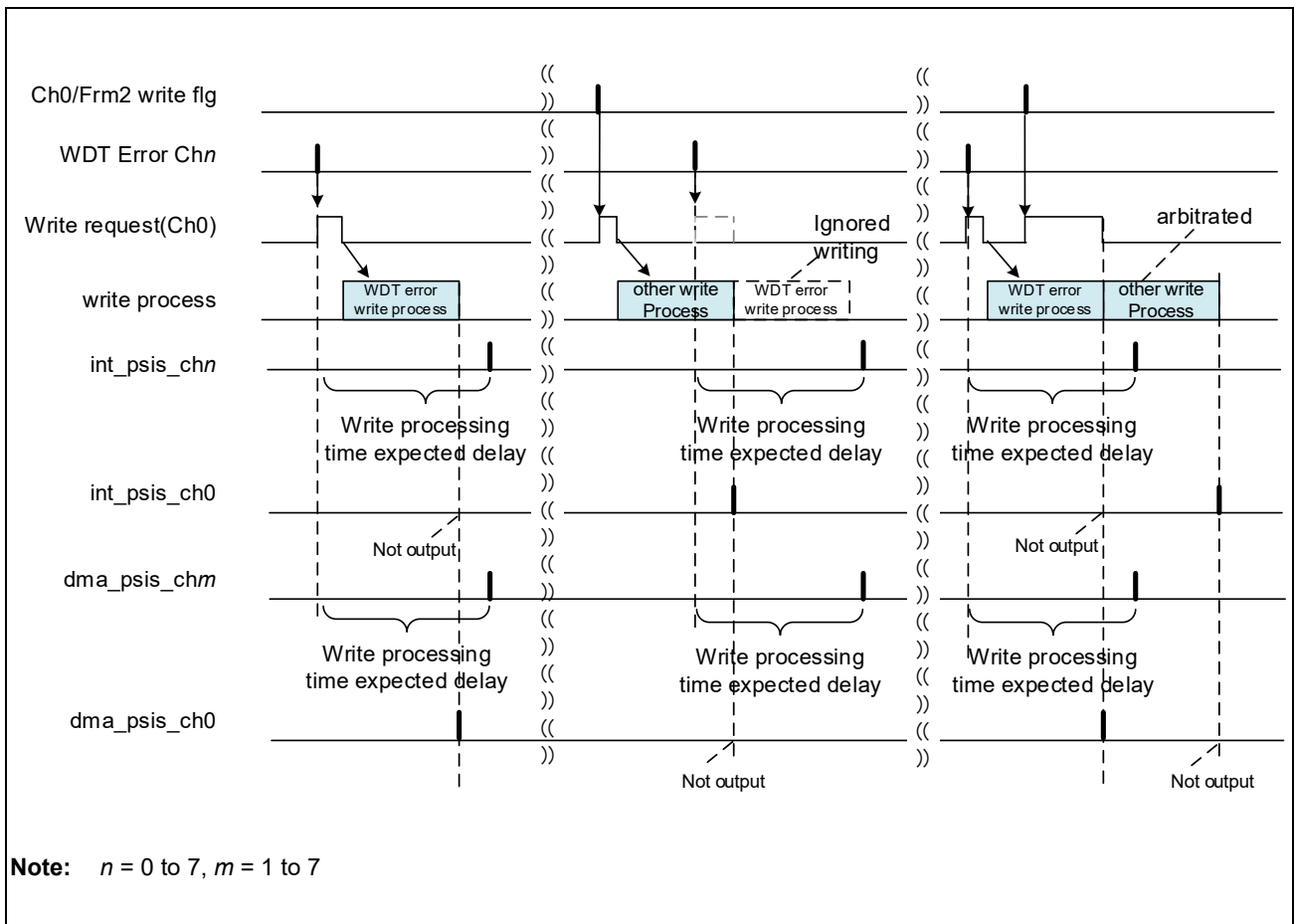


Figure 25.77 Write Request Arbitrated and Interrupt/DMA Output (Synchronous Mode)

The interrupt / DMA request by WDT error outputs from detection Ch after writing the MailBox of Ch0.

When a write request by WDT error during the write processing to Ch0/frm2, interrupt/DMA request outputs and the MB data triplet is discarded.

When a write request by another error (not WDT error) during the write processing to Ch0/frm2, the interrupt/DMA request outputs and the MB data triplet write after the write processing to Ch0/frm2.

(2) Asynchronous Mode

When reception of a frame ends, the WDT value is reset to the expiration value, and, at the same time, the WDT starts the next operation. If the WDT counter reaches 0 before reception of the next frame ends, occurrence of a WDT error is assumed.

In such a case, the WDT value is reset to the initial value, and the WDT starts operation again.

The conditions for starting the first operation of the WDT are as follows.

When the WDT enable bit is set to 1, an expiration value is set in the WDT counter. When the operation mode is subsequently switched to PSI5S mode, the WDT starts operation. If the WDT enable bit is 0 (disable), the WDT does not start operation.

If a WDT error occurs, an interrupt status (PSI5SPCIS1.ISTRWDT) occurs. The interrupt status is cleared by writing 1 to the corresponding bit (PSI5SPCISC1.ISTRWDT) in the interrupt status clear register.

The following figure shows the operation that is performed when a WDT error occurs in asynchronous mode.

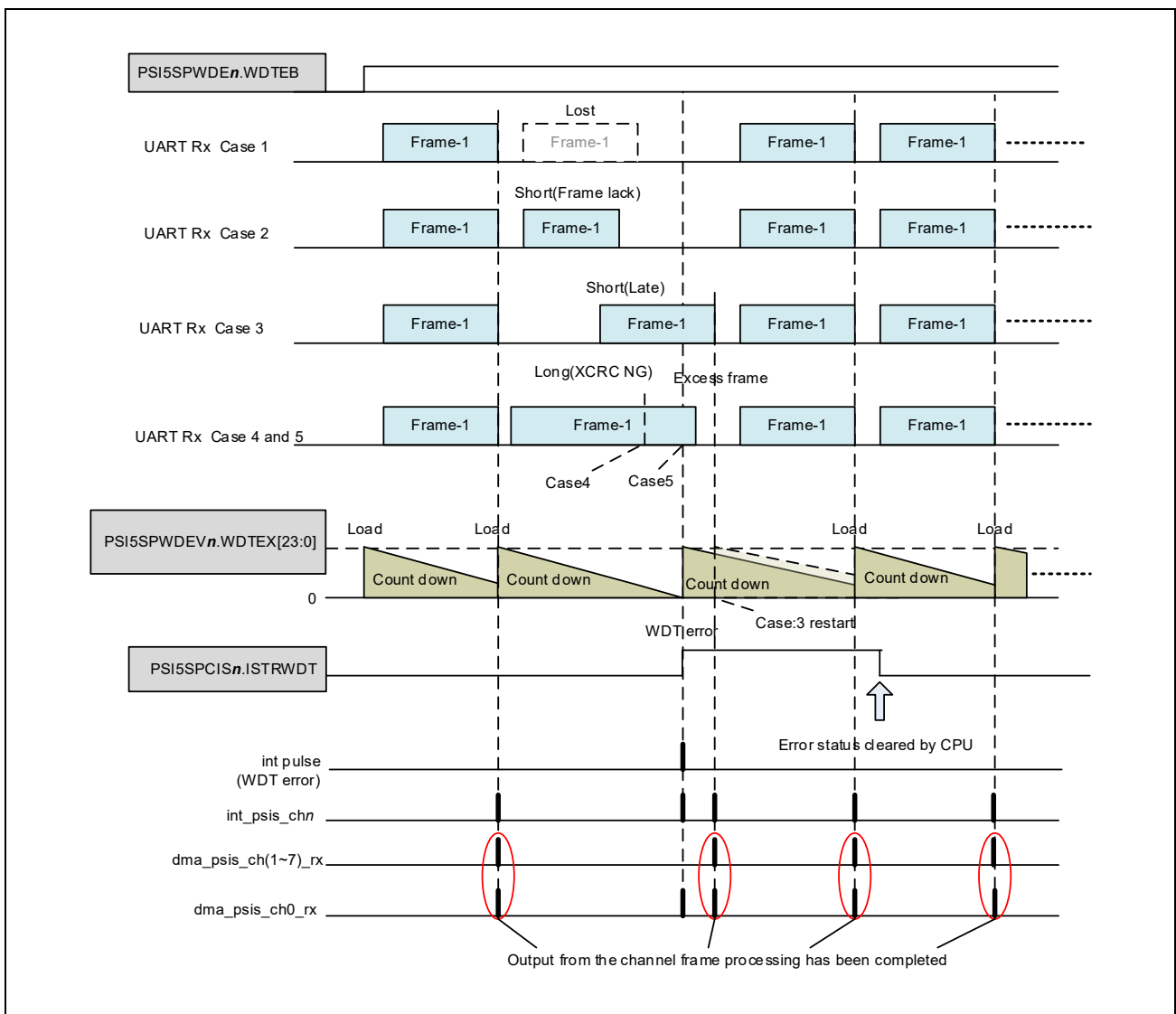


Figure 25.78 WDT Operation Timing (Asynchronous Mode) (WDT Error)

Below shows the MailBox triplet when a WDT error occurs in synchronous mode:

Table 25.99 MB Data Triplet - WDT Error (Asynchronous Mode)

Case of UART Rx	Case1	Case2	Case3	Case4	Case5
Store location	ch0/frm2	ch0/frm2	ch0/frm2	ch0/frm2	ch0/frm2
DCI	Ch0	Ch0	Ch0	Ch0	Ch0
CHID	Channel where WDT error detected	Channel where WDT error detected	Channel where WDT error detected	CHID in the header of the reception frame	Channel where WDT error detected
FID	0b000	FmID in the header of reception frame	FmID in the header of reception frame	FmID in the header of reception frame	FmID in the header of reception frame
MBORERR	MailBox overrun error	MailBox overrun error	MailBox overrun error	MailBox overrun error	MailBox overrun error
WDTERR	1	1	1	1	1
UTFRERR	0	UART framing error	UART framing error	UART framing error	UART framing error
UTPTERR	0	UART parity error	UART parity error	UART parity error	UART parity error
HEADERR	0	Header error	Header error	Header error	Header error
HEADST	0	HEADST in the header of reception frame	HEADST in the header of reception frame	HEADST in the header of reception frame	HEADST in the header of reception frame
CRCERR	0	0	0	CRC error	CRC error
CRC	000 _B	000 _B	000 _B	CRC of reception frame data	CRC of reception frame data
XCRCERR	0	0	0	1	1
XCRC	00_0000 _B	00_0000 _B	00_0000 _B	XCRC of reception frame data	XCRC of reception frame data
Payload data	000_0000 _H	Incomplete frame data *1,*2,*3	Incomplete frame data *1,*2,*3	Payload data of received frame data	Incomplete frame data *1,*2,*3
Time stamp	At WDT error detection (Ch0)	At WDT error detection (Ch0)	At WDT error detection (Ch0)	At header reception (Ch0)	At WDT error detection (Ch0)

Note 1. Not received (missing or incomplete) packet bits are "0".

Note 2. Incomplete data is stored in packets 2 to 4 and the lower 4 bits of packet 5.

Note 3. The incomplete data format is the same as synchronous mode.

The following figure shows the arbitration of the write request to Ch0 and the operation of the interrupt and DMA request.

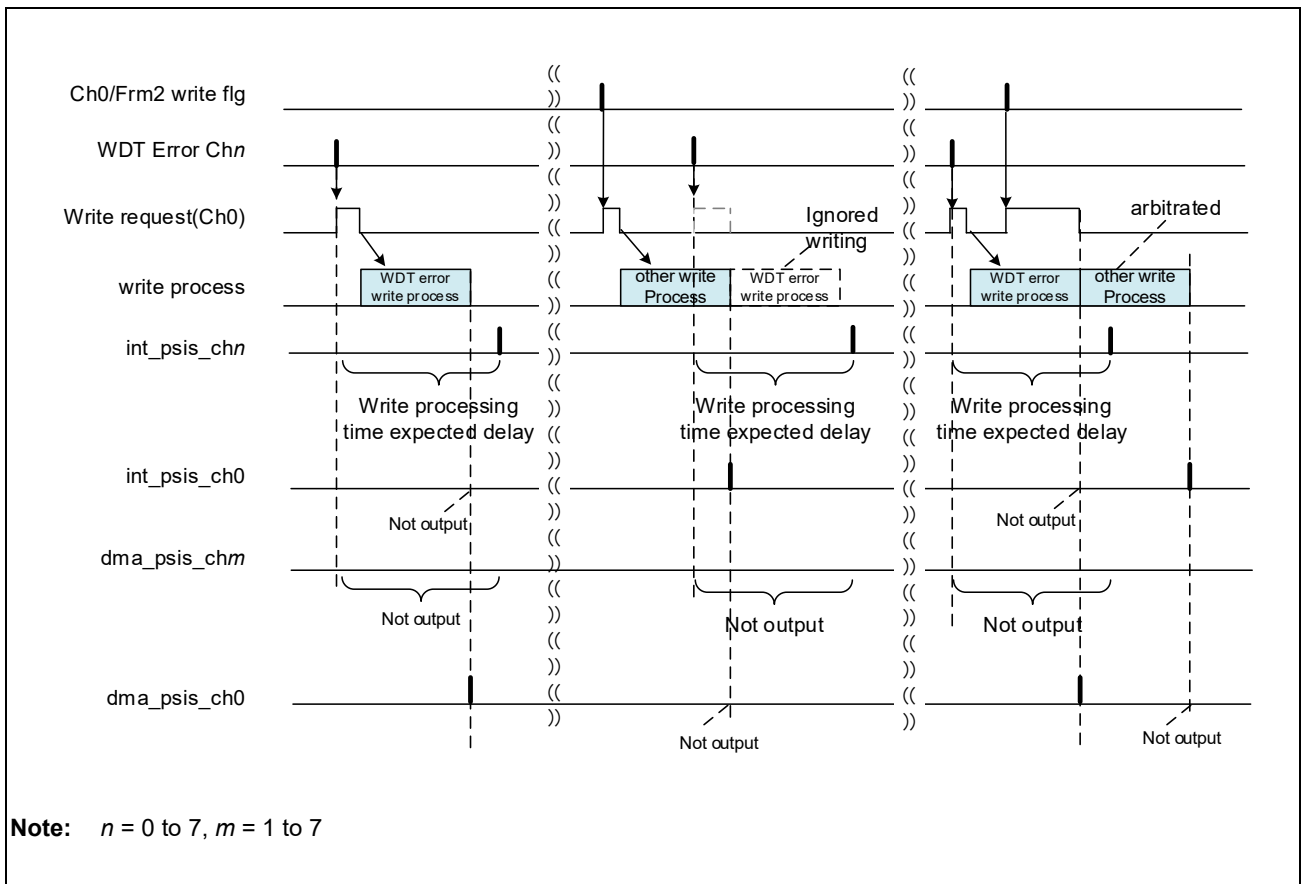


Figure 25.79 Write Request Arbitrated and Interrupt/DMA Output (Asynchronous Mode)

The interrupt request by WDT error outputs from detection Ch after writing the MailBox of Ch0.

The DMA request by WDT error outputs from Ch0 after writing the MailBox of Ch0.

When a write request by WDT error during the write processing to Ch0/frm2, interrupt request outputs and the MB data triplet is discarded.

When a write request by another error (not WDT error) during the write processing to Ch0/frm2, the interrupt request outputs and the MB data triplet write after the write processing to Ch0/frm2.

NOTE

When UART Parity error or UART Framing error or XCRC error is occurred, the received data is stored in Frame2 of Ch0. In that case, there is possible to detect the WDT error of reception Ch, because the reception is not stored in expected Ch.

25.6.2.9 Abnormal Transmission (Transmission Error)

If ECU-to-sensor data is set for a channel while ECU-to-sensor data is being transmitted*1 in the same channel, an ECU-to-sensor data overwrite error is determined to have occurred.

Note 1. Period from setting the value of transmission command to storing the last data in the Tx shifter.

If an overwrite error is determined, the data written afterward is not transmitted.

Data that is already being transmitted is not affected by the overwrite error.

If an overwrite error occurs, an interrupt status signal (PSI5SPCIS1.ISTDDSOW) is asserted. The interrupt status signal is cleared by writing 1 to the corresponding bit (PSI5SPCISC1.ISTCDDSOW) in the interrupt status clear register.

The following figure shows the operation that is performed when an ECU-to-sensor data overwrite error occurs.

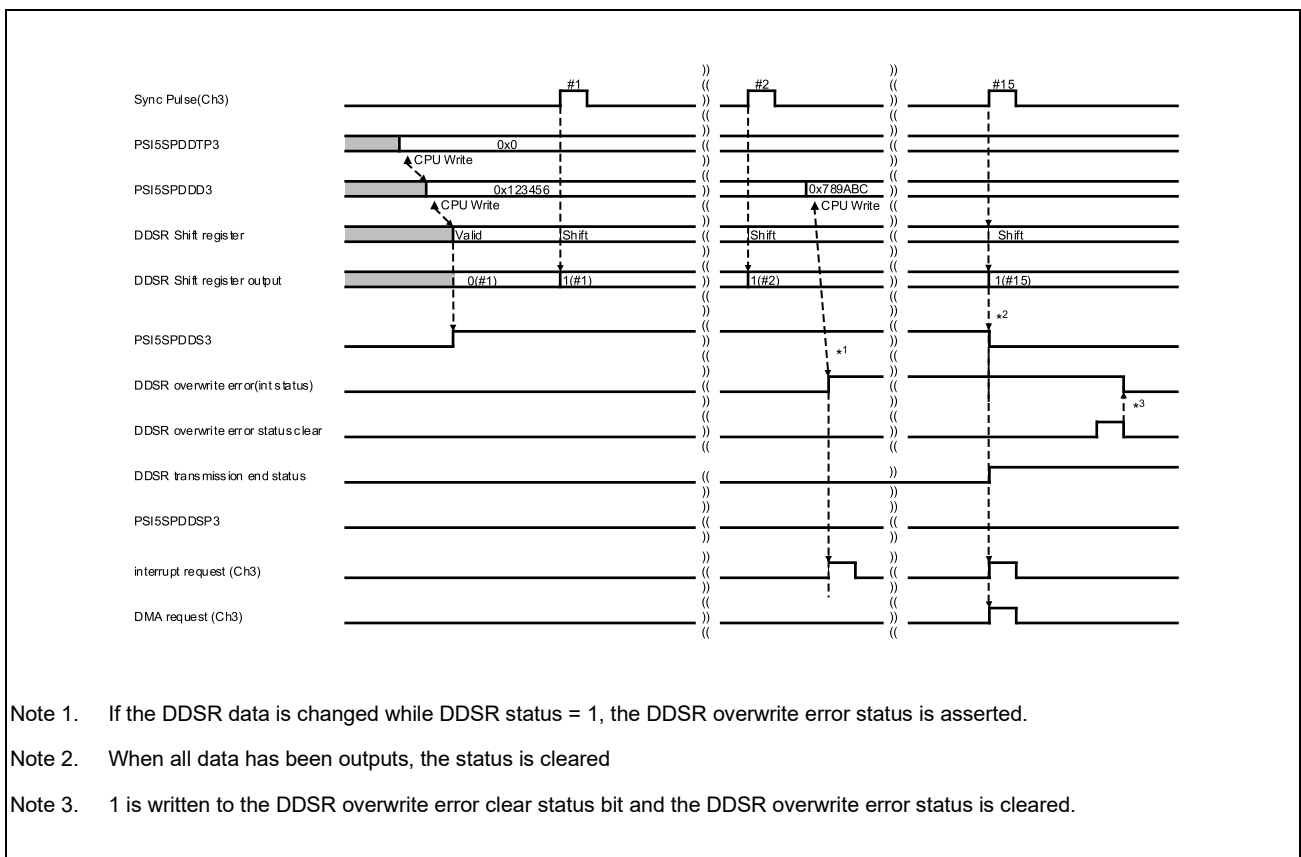


Figure 25.80 ECU-to-Sensor Data Transmission (Overwrite Error)

The following describes the operation that is performed when updating the ECU-to-sensor data conflicts with the start of the last data transmission triggered by a synchronization pulse.

When updating the ECU-to-sensor data occurs within 1 PLCK cycle from the start of the last data transmission triggered by a synchronization pulse, occurrence of an overwrite error is assumed.

When updating the ECU-to-sensor data occurs 2 PCLK cycles from the start of the last data transmission triggered by a synchronization pulse, occurrence of an overwrite error is not assumed.

In configuration mode, even if updating the ECU-to-sensor data occurs, PSI5-S does not start the transmission and PSI5-S does not detect an overwrite error.

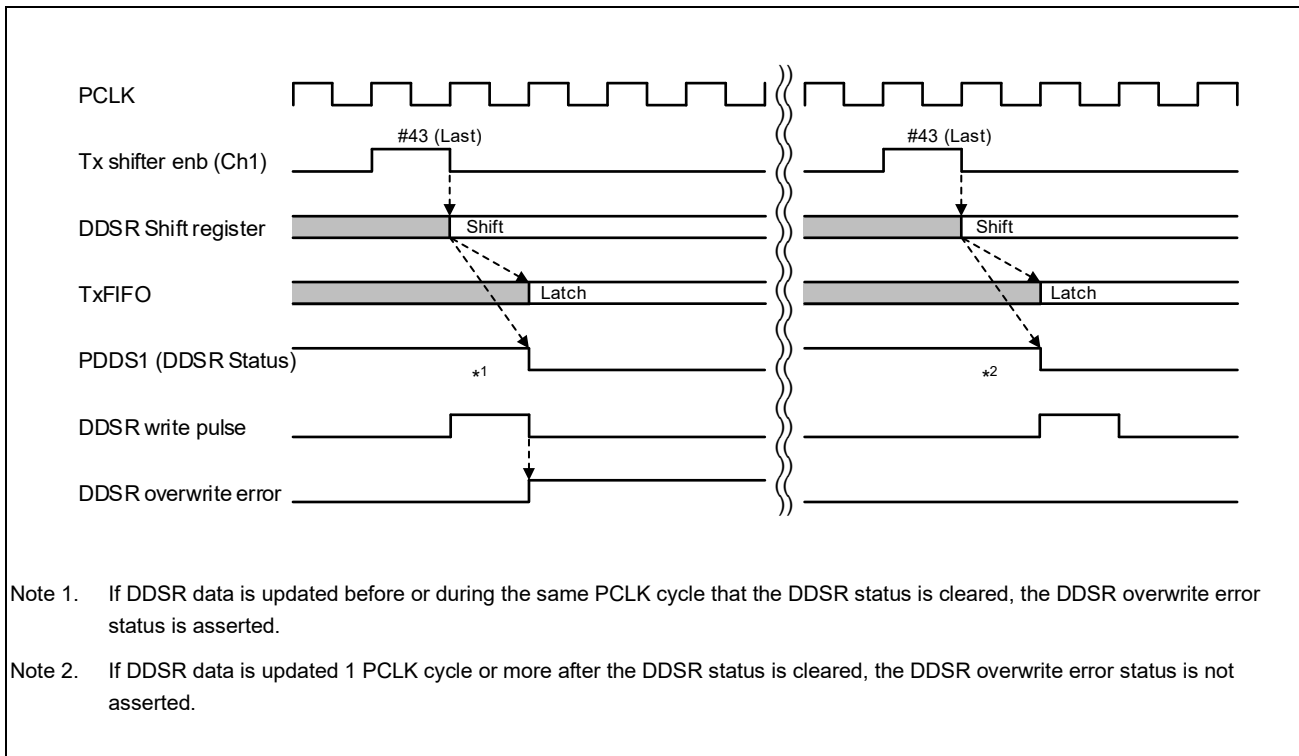


Figure 25.81 ECU-to-Sensor Data Transmit (Overwrite Error Operation is Conflicted)

25.6.2.10 Interrupt Output

PSI5-S can output eight interrupt signals.

Interrupts can be output only when interrupt output is enabled. If an interrupt factor occurs when interrupt output is disabled, no interrupt is output, but the interrupt status is indicated.

Table 25.100 below lists the factors of interrupts that are output in PSI5S mode.

Table 25.100 Interrupt Output Timings

Output Signal	Interrupt Output Timing
int_psis_ch0 to 7	When MB data is received and stored to SFR.
int_psis_ch0 to 7	When WDT operates, the specified time elapses, and the value matches the compare value.
int_psis_ch0 to 7	When a UART frame is received after the start of the idle period.
int_psis_ch0 to 7	When a UART frame is received and the next UART frame is received before the start of the idle period.
int_psis_ch0	When the last command data is stored in the Tx shifter.
int_psis_ch1 to 7	When the last DDSR data is stored in the Tx shifter.
int_psis_ch1 to 7	When DDSR data is written while another DDSR data is being transmitted.

The following describes the interrupt statuses and how to clear them.

To clear an interrupt status, write 1 to the bit corresponding to the interrupt status in the interrupt status clear register.

Table 25.101 Interrupt Statuses

Equipped Ch	Interrupt Status
Ch0	XCRC error
Ch0 to Ch7	CRC error
Ch0 to Ch7	Parity error
Ch0 to Ch7	Transceiver status error
Ch0	UART Rx parity error
Ch0	UART Rx framing error
Ch0 to Ch7	WDT error
Ch0 to Ch7	PSI5 frame reception overrun error
Ch0 to Ch7	PSI5 frame lack error
Ch0 to Ch7	PSI5 frame excess error
Ch0 to Ch7	PSI5 frame reception finish
Ch1 to Ch7	DDSR overwrite error
Ch1 to Ch7	DDSR transmit finish
Ch0	Command data transmit finish

The following describes the operation in which an interrupt status occurs and is cleared.

The following figure shows interrupt output and DMA request output operations in synchronous mode:

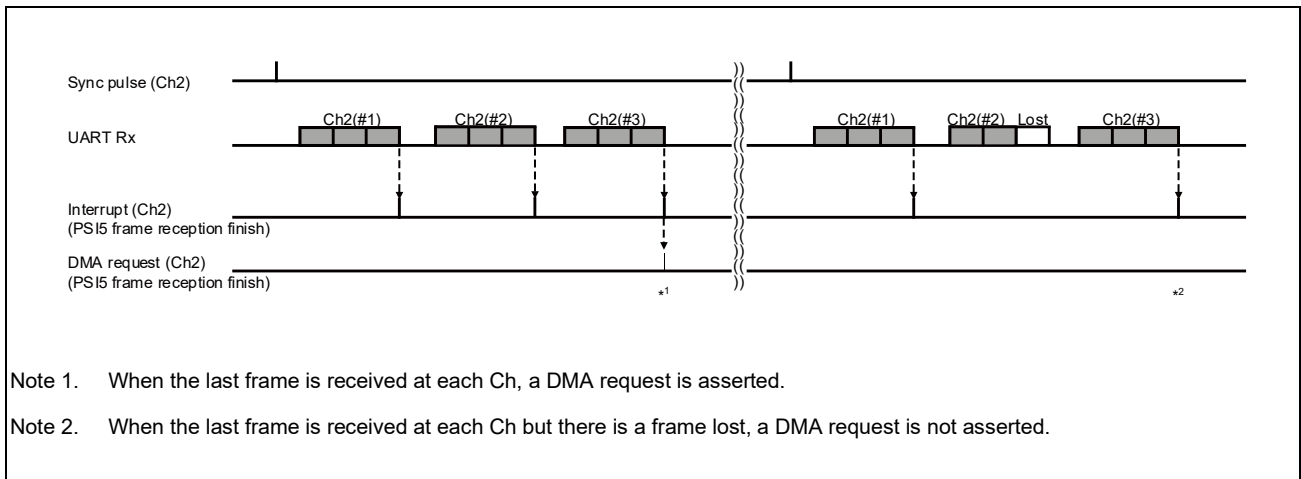


Figure 25.82 Interrupt and DMA Request Timing (Synchronous Mode)

The following figure shows interrupt output and DMA request output operations in asynchronous mode:

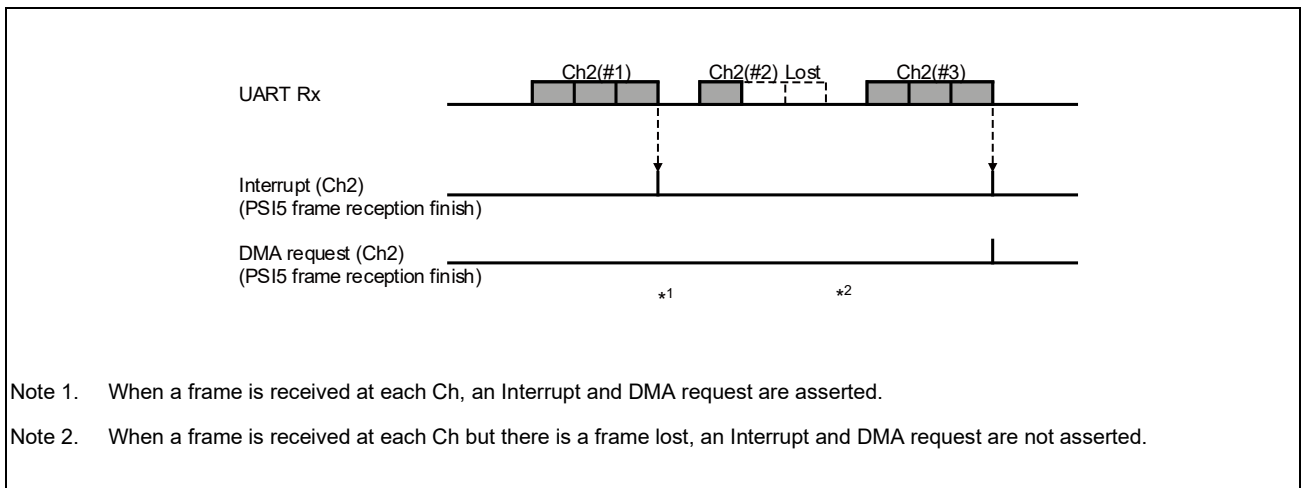


Figure 25.83 Interrupt and DMA Request Timing (Asynchronous Mode)

If the occurrence of an error status and the clearing of an interrupt status occur at the same time, the occurrence of the error has priority over the occurrence of the interrupt status.

The following figure shows the operation that is performed under conditions for setting and clearing interrupts.

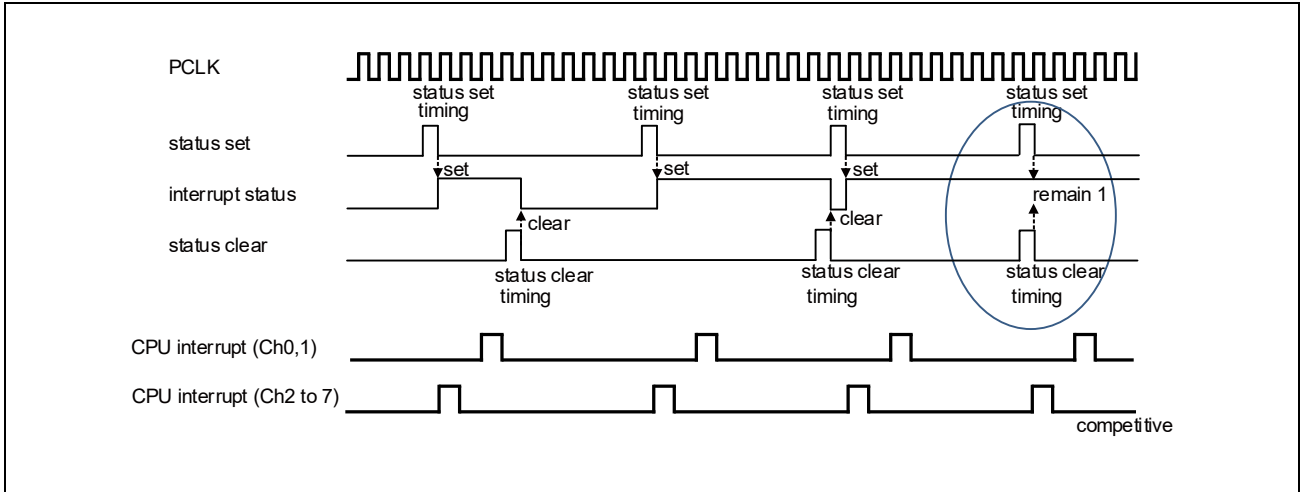


Figure 25.84 Interrupt Operation (Conflict)

If an interrupt setting condition is met while an interrupt status bit is set, an interrupt is output.

The following figure shows the operation described above.

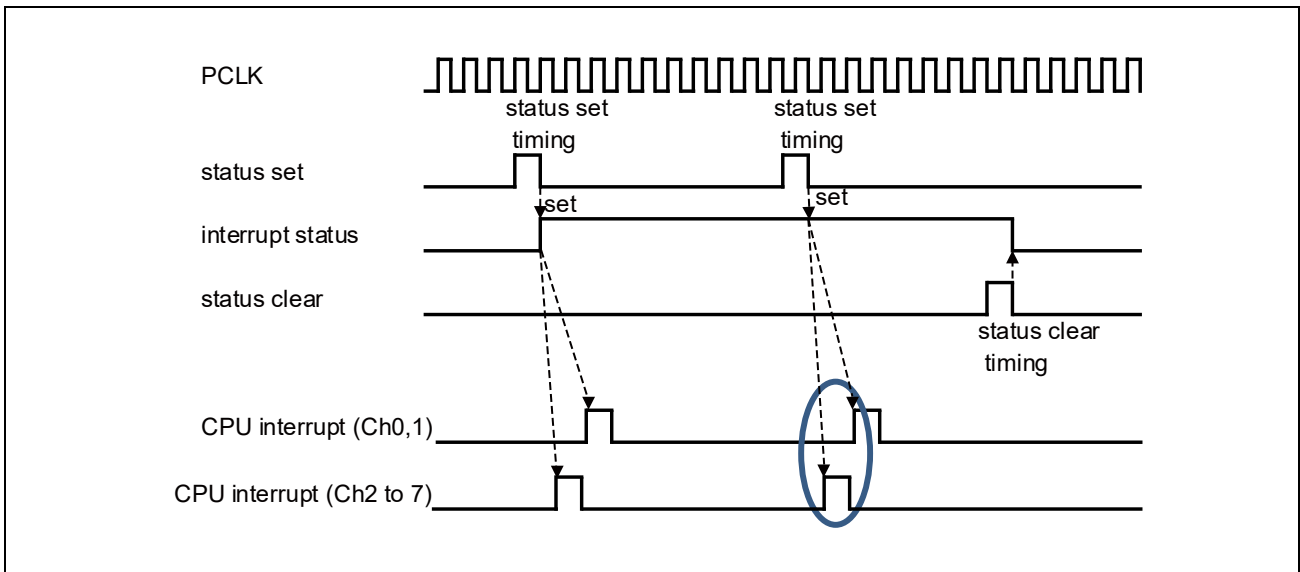


Figure 25.85 Interrupt Operation (Error Setting Occurs When Interrupt Status Is Set)

NOTE

A interrupt may be outputted in a period of maximum 14 PCLK even after PUOS.ACSTS becomes 0.
Then the interrupt should be ignored.

25.6.2.11 DMA Request Output

PSI5-S can output 15 DMA requests (eight receive requests and seven transmit requests).

DMA requests are output only when DMA request output is enabled. DMA requests are not output when DMA request output is disabled.

Table 25.102 below lists the factors of DMA requests that are output in PSI5S mode.

Table 25.102 Timings of DMA Transfer Request Output

Operation Mode	Equipped Channel	Rx/Tx	DMA Request Output Factor
PSI5S mode	Ch0 to Ch7	Rx	<Sync> When last the PSI5 frame is stored in SFR.
			<Async> When a PSI5 frame is stored in SFR.
	Ch0 to Ch7	Rx	When a WDT error occurs.
	Ch1 to Ch7	Tx	When the last DDSR data is stored in the Tx shifter

Figure 25.74, **Figure 25.78**, **Figure 25.82**, and **Figure 25.83** show the DMA request output operations for reception.

Figure 25.66 and **Figure 25.67** show the DMA request output operations for transmission.

25.6.2.12 Transition to Configuration Mode

The operation of PSI5-S to return to the configuration mode is described in **Section 25.4, Operation Modes**.

Figure 25.86 below describes the transition to the configuration mode during reception and transmission.

When CPU writes PSI5SPUOEB.OPEN to 0 during reception, PSI5-S waits for 16 PCLK cycles and then transitions to configuration mode if the PSI5-S is not sending a frame.

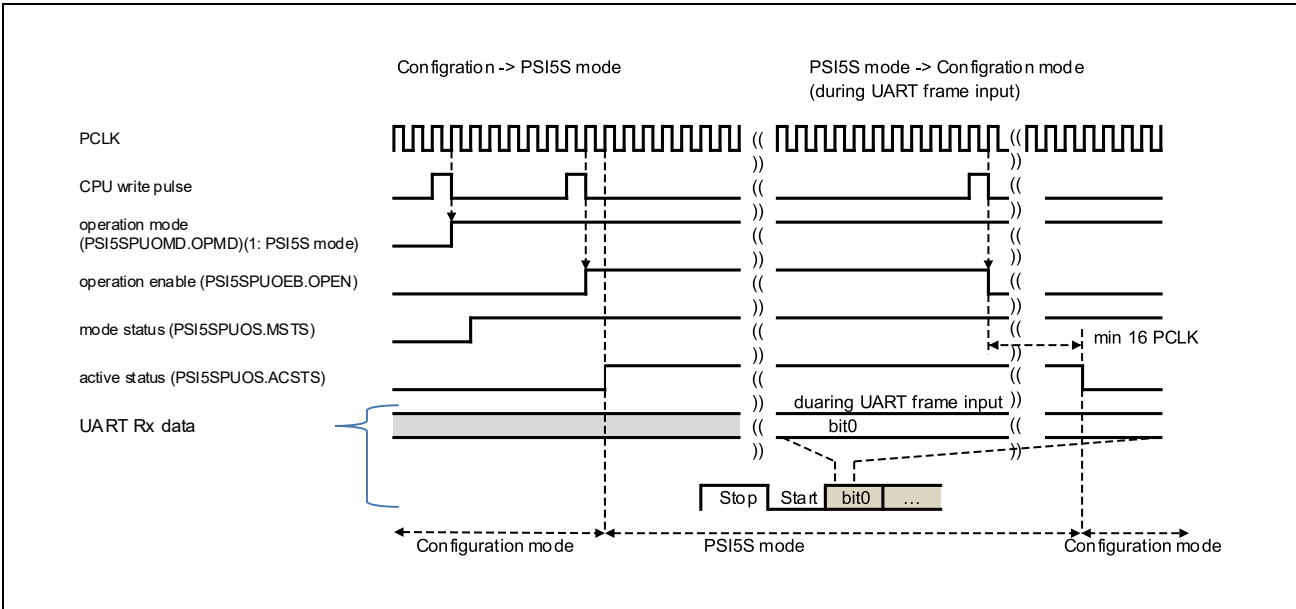


Figure 25.86 Move from PSI5S Mode to Configuration Mode (Rx)

When CPU writes PSI5SPUOEB.OPEN to 0 during transmission, PSI5-S waits for 16 PCLK cycles and then transitions to configuration mode if the PSI5-S isn't sending a frame. If PSI5-S sends a frame, it transitions to configuration mode after it completes to send the frame.

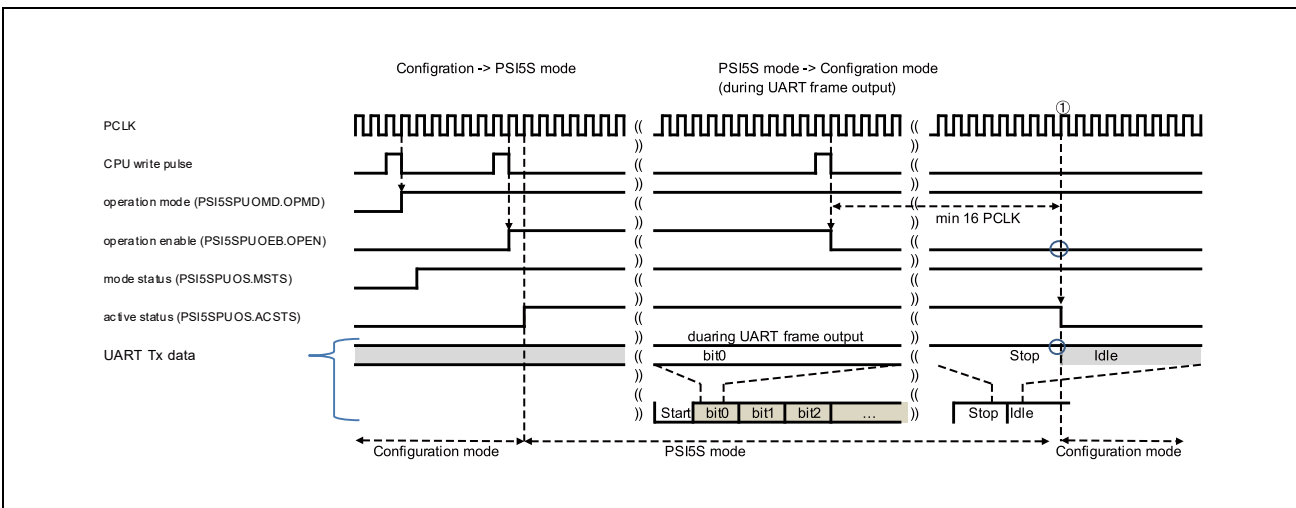


Figure 25.87 Move from PSI5S Mode to Configuration Mode (Tx)

NOTE

In a case that the command to transceiver has been accepted in UART transmission module in PSI5-S when CPU writes PSI5SPUOEB.OPEN to 0, PSI5-S wait for completion of the UART transmission. The maximum period of the mode transition time is a period of 9 UART frames.

25.6.3 Operations in UART Mode

When the operation mode is UART mode (PSI5SPUOMD.OPMD = 0) and the operation is enabled (PSI5SPUOEB.OPEN = 1), PSI5-S can perform 1-byte UART communication with external devices.

Baud rate settings (prescaler division ratio, clock division ratio, and oversample number) are described in **Section 25.5.1, Common Setting Procedure**.

For the UART specifications, see **Section 25.2.1, Functional Overview**.

25.6.3.1 Normal Reception

PSI5-S receives UART frames. Each time PSI5-S receives one UART frame, PSI5-S performs serial-to-parallel (S/P) conversion and determines whether there are any reception errors.

If no reception errors are found, PSI5-S determines that the reception is normal, and then updates the receive data register (PSI5SUCRD) and normal reception end status (PSI5SUCRS.UTRFIN).

Figure 25.88 shows an example of UART reception.

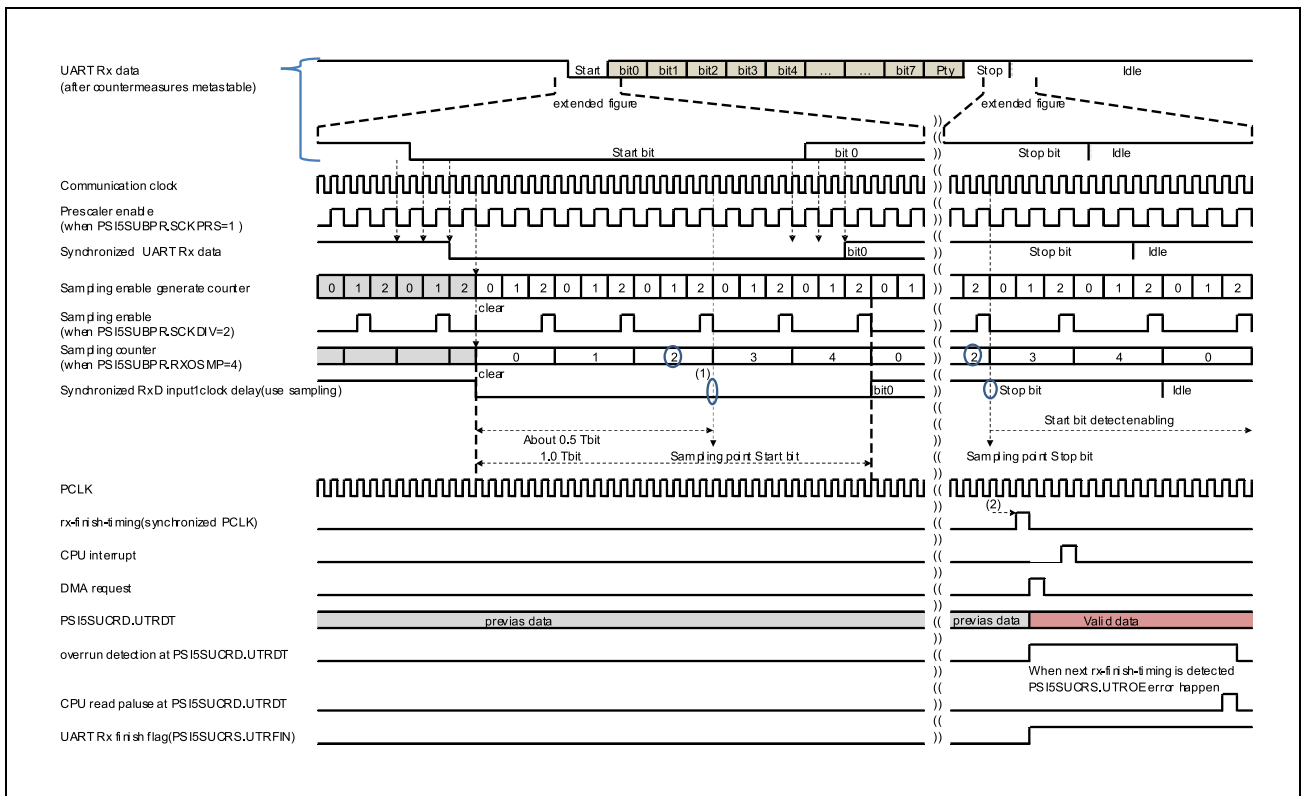


Figure 25.88 UART Rx (No Error)

- (1) When the sampling counter value is “PSI5SPUBPR.RXOSMP/2”, PSI5-S samples receive data. (Example: When PSI5SPUBPR.RXOSMP = 4, PSI5-S fetches receive data when the sampling counter value is 2.)
- (2) The asynchronous transfer processing based on PCLK and the communication clock causes a delay of several clock cycles.

25.6.3.2 Normal Transmission

When the transmit data register (PSI5SUCID) is updated, PSI5-S transmits UART frames.

When transmission ends, PSI5-S updates the transmission status (PSI5SUCTS, PSI5SUCTM).

Figure 25.89 shows an example of UART transmission.

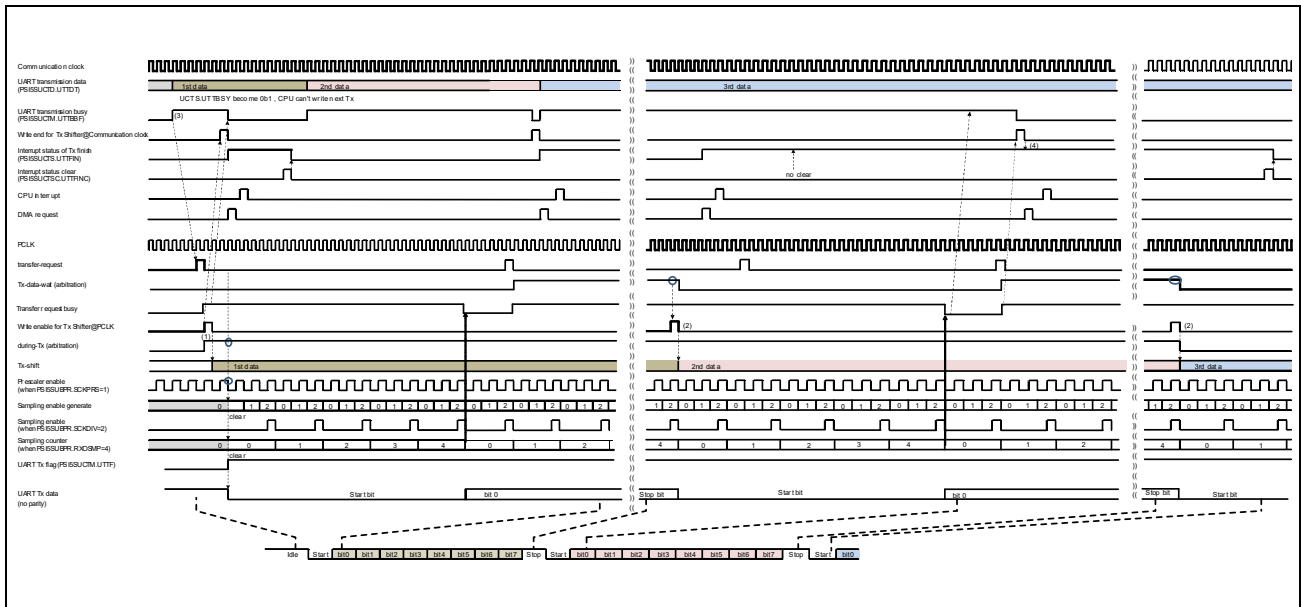


Figure 25.89 UART Tx (No Error)

- (1) When the value of transfer-request is 1 and the during-Tx signal is high level, the signal to enable writing to the transmit data shift register (Tx Shifter) is output.
- (2) When the Tx-data-wait signal is high level at the start of stop bit output, the signal to enable writing to the Tx Shifter register is output.
- (3) Even if the Tx Shifter register is not being used, PSI5-S is busy until asynchronous transfer ends.
- (4) Even when the PSI5SUCTS.UTTFIN signal is high level, interrupts and DMA requests are generated.

Items (1) and (2) above describe the conditions for storing data in the Tx shifter.

25.6.3.3 Abnormal Reception (Reception Error)

After UART reception ends, parity, framing, and overrun errors are detected if they exist.

If an overrun error occurs, the data which is read from the mailbox is unpredictable.

If an error occurs, an interrupt status (see **Table 25.103**) occurs. The interrupt status signal is cleared by writing 1 to the corresponding bit (see **Table 25.103**) in the interrupt status clear register.

Table 25.103 below describes reception errors that can occur during operations in UART mode.

Table 25.103 UART Communication Rx Errors

Error	Status	Status Clear	Detection Condition
Parity error	PSI5UCRS.UTRPE	PSI5UCRSC.UTRPECL	When UART stop bit is received, the received parity value does not match.
Framing error	PSI5UCRS.UTRFE	PSI5UCRSC.UTRFECL	When UART stop bit is received, the received stop bit value is 0.
Overrun error	PSI5UCRS.UTROE	PSI5UCRSC.UTROECL	When UART stop bit is received, a new data is received before the previous data is read.*1

Note 1. When the PSI5-S returns to UART mode after a transition to configuration mode, an overrun error is not detected even if a new data is received in a condition that a previous received data has not been read.

PSI5-S controls the Rx-finish timing and various errors when the stop bit is sampled (midway between stop bits).

Figure 25.90 and **Figure 25.91** below show examples of operation in which an Rx overrun, framing, and parity error occur when reception ends.

If both parity and framing errors occur, occurrence of a framing error is assumed.

When no errors occur, the Rx-finish status signal is set to high level.

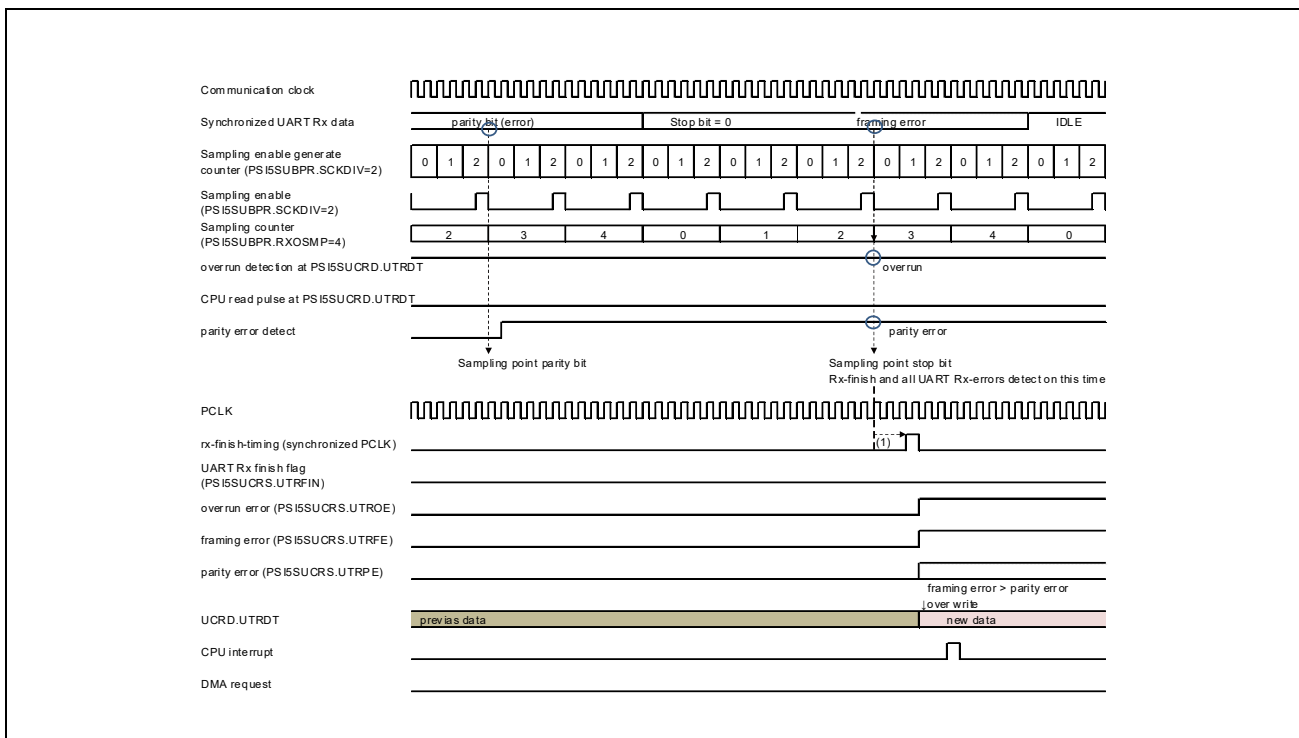


Figure 25.90 UART Rx (Received Error)

- (1) The asynchronous transfer processing based on PCLK and the communication clock causes a delay of several clock cycles.

Figure 25.91 below describes an operation in which an overrun error is detected, as well as an example of an operation in which overrun error detection conflicts with receive data reading and an overrun error is not detected.

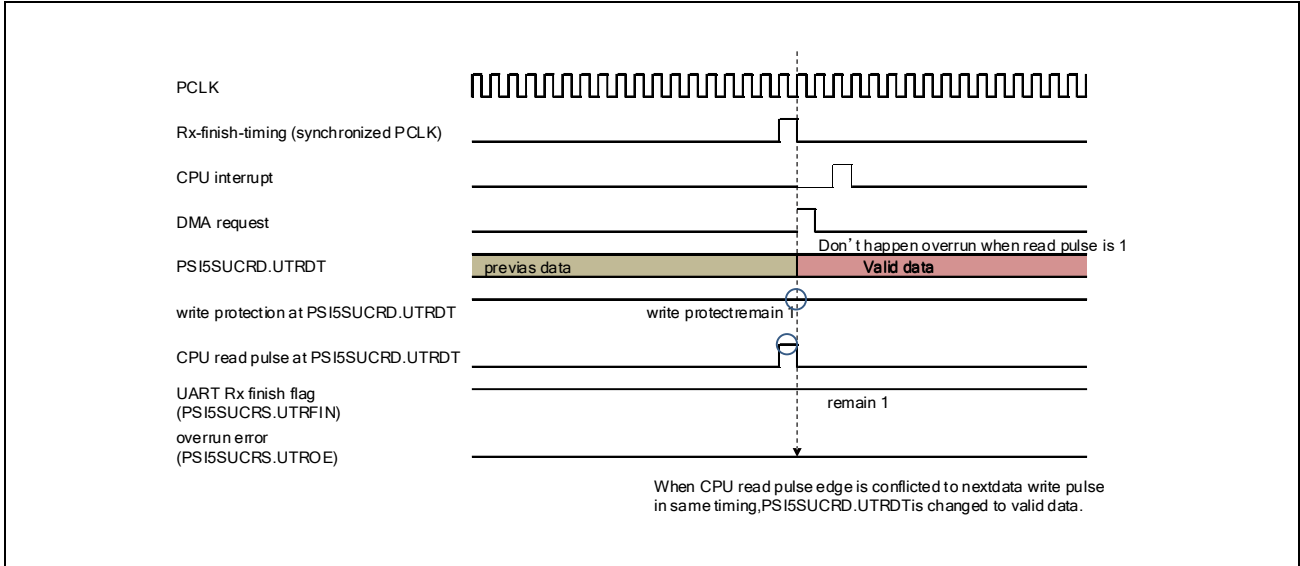


Figure 25.91 UART Rx (Overrun Error Is Not Detected)

The following figure shows an example of operation in which overrun error detection conflicts with receive data reading and an overrun error is detected.

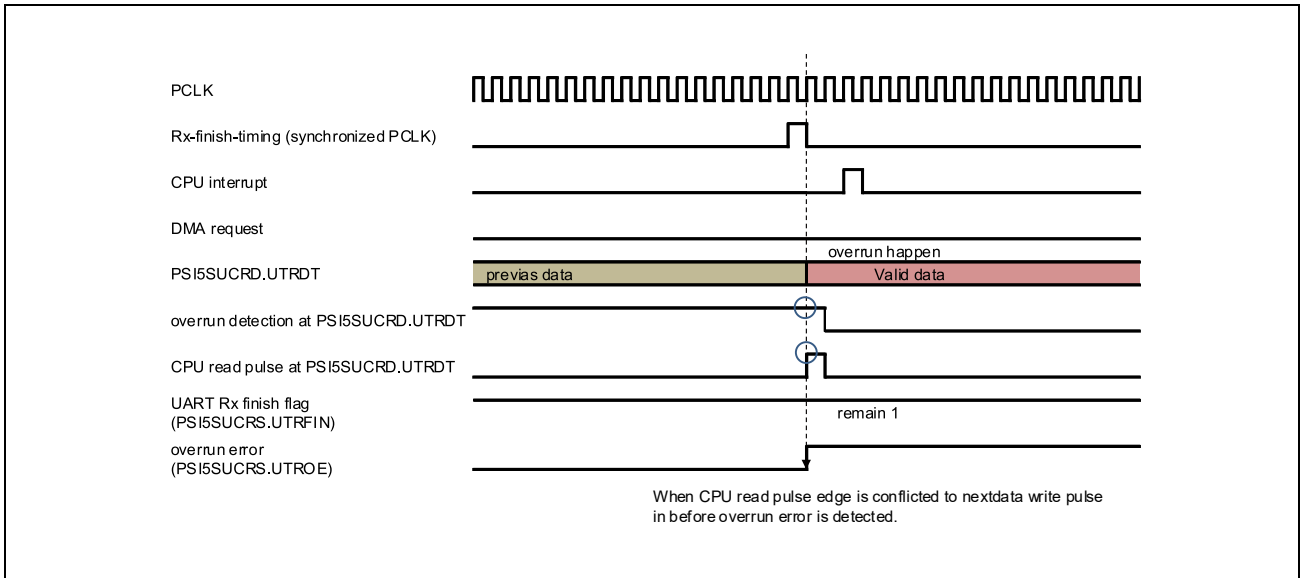


Figure 25.92 UART Rx (Overrun Error Is Detected)

25.6.3.4 Abnormal Transmission (Transmission Error)

This section describes the transmission errors that can occur in UART mode.

A transmission error is detected if new transmit data is set while PSI5-S is busy with UART transmission.

When occurrence of an overwrite error is determined, the data written subsequently is not transmitted.

Already set data is not affected by the overwrite error, and the data is transmitted.

The interrupt status is caused by the occurrence of an overwrite error. The interrupt status can be cleared by writing 1 to the interrupt status clear register.

Table 25.104 UART Communication Tx Error

Error	Status	Status Clear	Detection Condition
Overwrite error	PSI5SUCTS.UTTOWE	PSI5SUCTSC.UTTOWECL	CPU writes (PSI5SUCTD.UTD[7:0]), while PSI5SUCTS.UTTBBF is asserted.

The following figure shows an example of operation in which transmission errors occur.

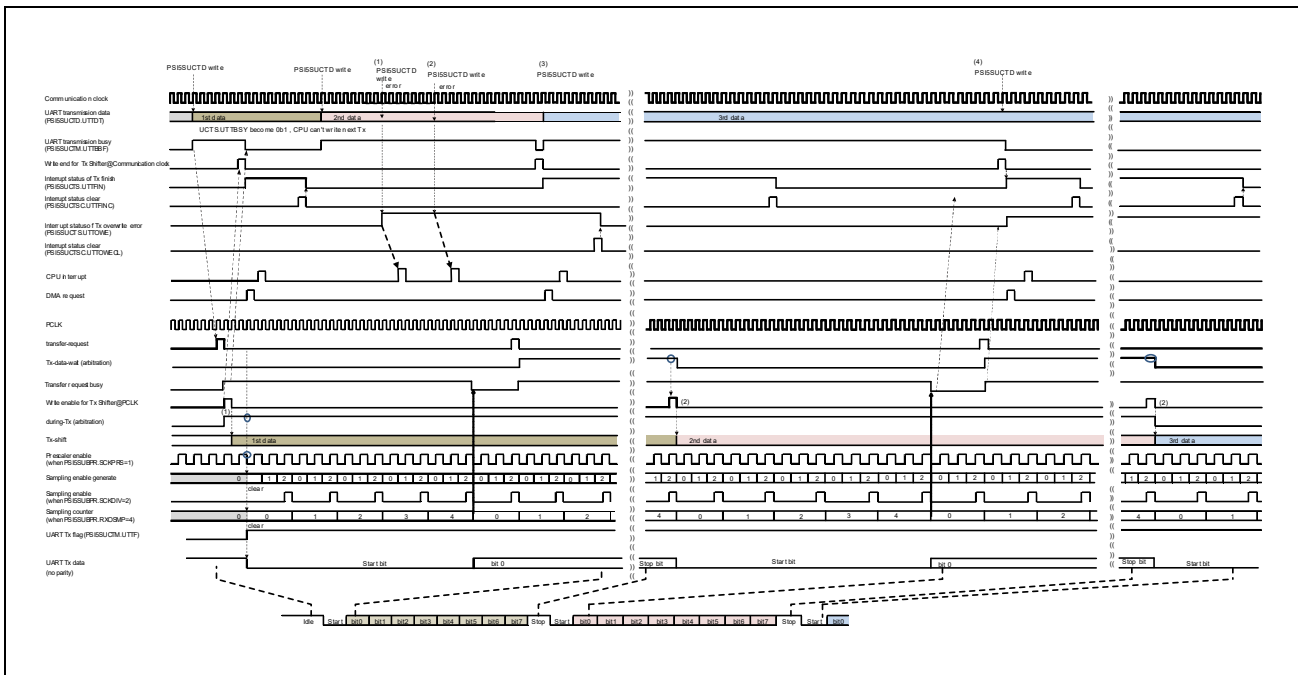


Figure 25.93 UART Tx (Overwrite Error)

- (1) If data is written to the PSI5SUCTD register with the PSI5SUCTS.UTTBSY signal set to high level, the PSI5SUCTM.UTTOWE signal is set to high level (indicating an overwrite error), and an interrupt is output.
- (2) If data is written again to the PSI5SUCTD register while the PSI5SUCTS.UTTOWE signal is high level (indicating an overwrite error), an interrupt is outputs again.
- (3) Even in the overwrite error status (PSI5SUCTS.UTTOWE = high level), the PSI5SUCTS.UTTFIN signal is set to high level and interrupts and DMA requests are output.
- (4) This timing is a boundary for the occurrence of an overwrite error (PSI5SUCTS.UTTOWE = high level). The overwrite error occurs together with transmission end (PSI5SUCTS.UTTFIN = 1). In this case, both status signals are set to high level, and interrupts and DMA requests are output.

In configuration mode, even if data is written to the PSI5SUCTD register, the PSI5-S does not start the transmission and the PSI5-S does not detect overwrite error.

25.6.3.5 Interrupt Output

PSI5-S can output eight interrupt signals.

In UART mode, interrupts are output to channels 0 and 1.

Interrupts can be output only when interrupt output is enabled. If an interrupt factor occurs when interrupt output is disabled, no interrupt is output, but the interrupt status is indicated.

Table 25.105 below lists the factors of interrupts that are output in UART mode:

Table 25.105 Interrupt Output Timings

Output Signal	Interrupt Output Timing
int_psis_ch0	When UART data is received
int_psis_ch1	When UART data is stored in the Tx shifter.
int_psis_ch1	When UART data is written before the Tx shifter data is read.

Table 25.106 below describes the interrupt statuses and clearing of interrupt statuses.

To clear an interrupt status, write 1 to the bit corresponding to the interrupt status in the interrupt status clear register.

Table 25.106 Interrupt Status

Equipped Ch	Interrupt Status
Ch0	UART Rx parity error
Ch0	UART Rx framing error
Ch0	UART Rx overrun error
Ch0	UART Rx finish
Ch1	UART Tx overwrite error
Ch1	UART Tx finish

Figure 25.90, **Figure 25.91**, **Figure 25.92** and **Figure 25.93** show interrupt output operations.

25.6.3.6 DMA Request Output

PSI5-S can output 15 DMA requests (eight receive requests and seven transmit requests).

In UART mode, DMA requests are output to channel 7 for reception and transmission.

DMA requests are output only when DMA request output is enabled. DMA requests are not output when DMA request output is disabled.

Table 25.107 below lists the factors of DMA requests that are output in UART mode.

Table 25.107 Timings of DMA Transfer Request Output

Operation Mode	Equipped Ch	Rx/Tx	DMA Request Output Factor
UART mode	Ch7	Rx	When UART data is received (no error)
	Ch7	Tx	When UART data is stored in the Tx shifter

Figure 25.88 and **Figure 25.89** show the DMA request output operations.

25.6.3.7 Transition to Configuration Mode

The operation of PSI5-S to return to the configuration mode is described in **Section 25.4, Operation Modes**.

Figure 25.94 below describes the transition to the configuration mode during reception and transmission.

When CPU writes PSI5SPUOEB.OPEN to 0 during reception, PSI5-S waits for 16 PCLK cycles and then transitions to configuration mode if the PSI5-S isn't sending a frame.

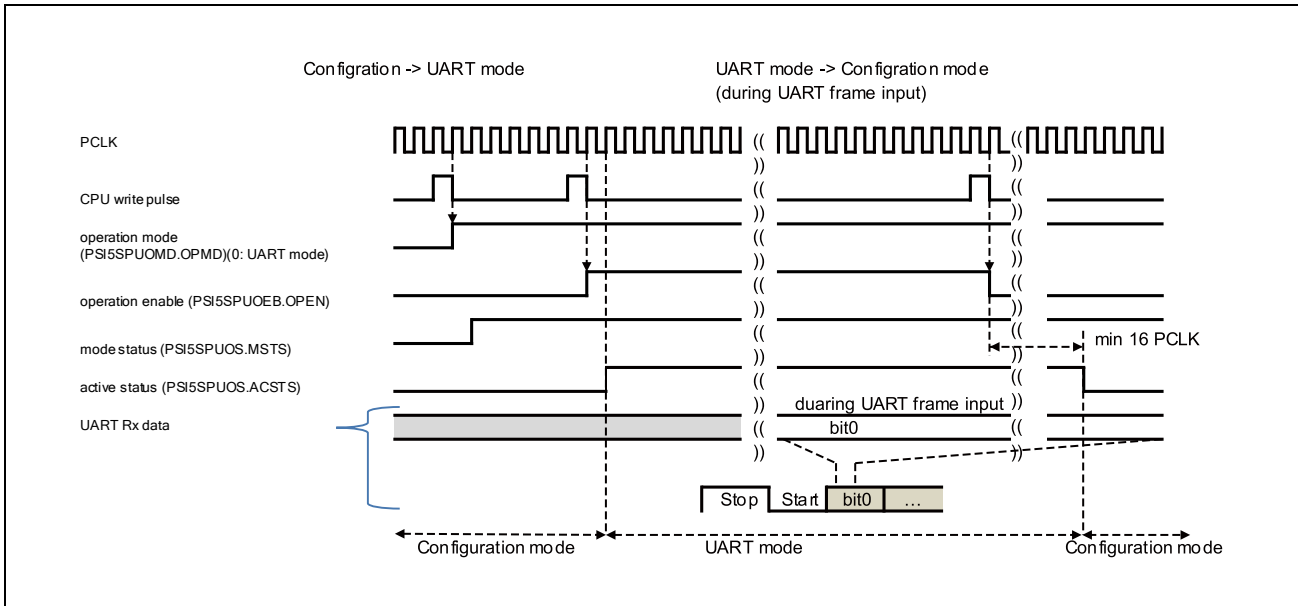


Figure 25.94 Move from UART Mode to Configuration Mode (Rx)

When CPU writes PSI5SPUOEB.OPEN to 0 during transmission, PSI5-S waits for 16 PCLK cycles and then transitions to configuration mode if the PSI5-S isn't sending a frame. If PSI5-S is sending a frame, it will transition to configuration mode after it completes to send the frame.

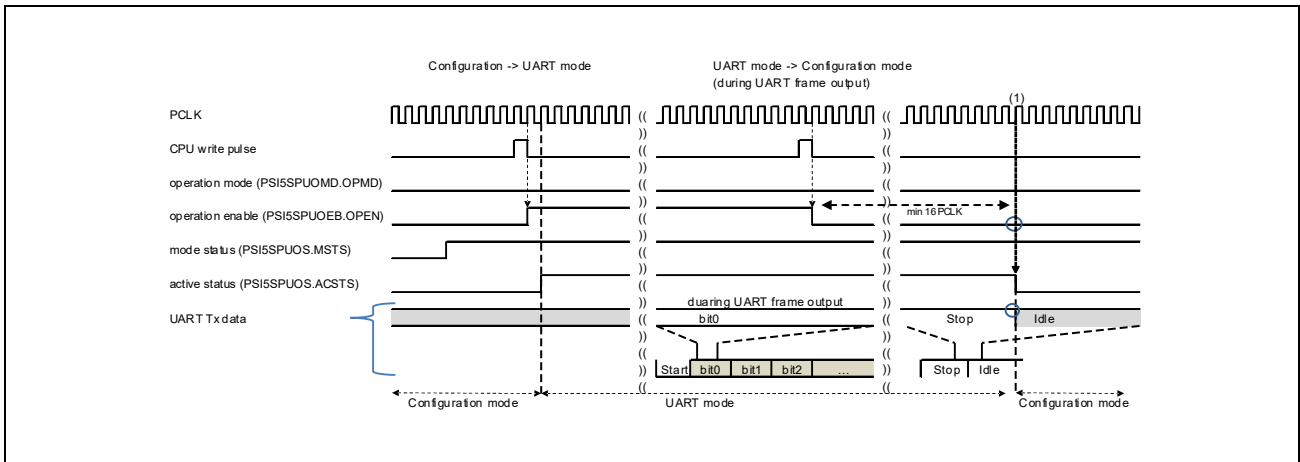


Figure 25.95 Move from UART Mode to Configuration Mode (Tx)

NOTE

In a case that the command to transceiver has been accepted in UART transmission module in PSI5-S when CPU writes PSI5SPUOEB.OPEN to 0, PSI5-S wait for completion of the UART transmission. The maximum period of the mode transition time is a period of 2 UART frames.

25.6.4 Clearing Status Signals

Each status signal can be cleared by writing 1 to the corresponding bit in the status clear register. Note, however, that the following statuses are cleared automatically when their clearing conditions are met:

PSI5SPUOS.SWSTS: Cleared when the software reset period has passed (**Figure 25.26**)

PSI5SPUOS.ACSTS: Cleared when PSI5-S returns to the configuration mode

PSI5SPTFS.TXSTS: Cleared when the last byte of data is stored in the Tx shifter (**Figure 25.63**)

PSI5SUCTM.UTTF: Cleared when stop bits are transferred in UART mode.

PSI5SUCTM.UTTBBF: Cleared when transmit data is loaded into the transmit buffer in UART mode (**Figure 25.89**)

PSI5SPDDS(1-7).DDSRSTS: Cleared when the last ECU-to-sensor data is stored in the Tx shifter in PSI5S mode (**Figure 25.66**) or when a DDSR transmission stop request is generated (**Figure 25.68**)

If setting and clearing conditions are both met at the same time, setting conditions have priority over clearing conditions.

25.7 Note

25.7.1 Restrictions

- Baud rate tolerance must be within 1.5%.

The minimum interval is 60 PCLK cycles from transition to configuration mode until set operation to enable (PSI5SPUOEB.OPEN = 1).

25.7.2 Notes

When the UART receive data pin is “Low” level at the transition to UART mode, the PSI5-S may detect a UART parity error or a UART framing error.

Section 26 High Speed Serial Peripheral Interface (HS-SPI)

This section contains a generic description of the High Speed Serial Peripheral Interface (HS-SPI).

The first part of this section describes the specific properties of this product, such as the number of units and register base addresses. The remainder of the section describes the functions and registers of HS-SPI.

26.1 Features of HS-SPI

26.1.1 Units and Channels

This microcontroller has the following number of HS-SPI channels.

Table 26.1 Number of Channels (for E2x-FCC1)

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Channels	1	1
Name	HSPIn (n = 0)	HSPIn (n = 0)

Table 26.2 Number of Channels (for E2M)

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Channels	1	1
Name	HSPIn (n = 0)	HSPIn (n = 0)

Table 26.3 Index

Index	Meaning
n	Throughout this section, the individual HS-SPI units are identified by the index "n": For example, HSPInCKEN is the HSPIn Clock Control Register.

26.1.2 Register Base Address

HS-SPI base addresses are listed in the following table.

In general, HS-SPI register addresses are given as offsets from the base addresses.

Table 26.4 Register Base Address

Base Address Name	Base Address	Bus Group
<HSPI0_base>	FF04 0000 _H	Peripheral Group 9

26.1.3 Clock Supply

Clock supply by and to HS-SPI is listed in the following table.

Table 26.5 Clock Supply

Unit Name	Unit Clock Name	Clock Supply Name
HSPIn	PCLK	CLK_HBUS
HSPIn	ACLK	CLK_HBUS
HSPIn	TXCLK	CLK_HSB

26.1.4 Interrupt Requests

The HS-SPI interrupt requests are listed in the following table.

Table 26.6 Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
INTHSPI0CSTART	INT_HSPI_CSTART	Communication start interrupt (slave only)	481	—	—
INTHSPI0CEND	INT_HSPI_CEND	Communication complete interrupt	482	—	—
INTHSPI0ERR	INT_HSPI_ERR	Communication error interrupt	483	—	—
INTHSPI0BERR	INT_HSPI_BERR	H-Bus error interrupt	484	—	—

26.1.5 Reset Sources

HS-SPI reset sources are shown below.

HS-SPI is initialized by the following reset sources.

Table 26.7 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
HSPIn	All registers	✓	✓	✓	✓	—	—

26.1.6 External Input/Output Signals

HSPIn external input/output signals are listed below.

Table 26.8 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
HSPi0		
SPCK	Transmission clock	HSSPI0_CLKP/ HSSPI0_CLKN
SSL0	Slave selection input signal	HSSPI0_SSLI
SSL0	Slave selection output signal	HSSPI0_SSL
MOSI/MISO	HSSPI transmission data	HSSPI0_TXDP/ HSSPI0_TXDN
MOSI/MISO	HSSPI reception data	HSSPI0_RXDP/ HSSPI0_RXDN

26.2 Overview

The HS-SPI supports full- and half-duplex synchronous serial communication.

26.2.1 Functional Overview

Table 26.9 Functional Overview of HS-SPI

Item	Description												
Communication method	<ul style="list-style-type: none"> • Either master or slave can be selected (multi-master is not supported) • Communication mode can be set to: Master: transmission or transmission/reception Slave: transmission, reception, or transmission/reception Master reception is not supported • SPCK polarity can be selected • The SPCK phase can be selected 												
Data format	<ul style="list-style-type: none"> • Either MSB or LSB first can be selected • Either inclusion or non-inclusion of parity can be selected (the frame includes 1 parity bit when inclusion is selected); and either odd or even parity can be selected • The number of data bits per frames for transfer can be selected (8, 16, 24, 32, 40, 48, 56, or 64 bits) • Up to 64K frames are transferable per communication round 												
Bit rate	<ul style="list-style-type: none"> • Maximum bit rate for each communication mode [TXCLK at 80 MHz] <table border="1"> <thead> <tr> <th></th> <th>Transmission</th> <th>Reception</th> <th>Transmission/Reception</th> </tr> </thead> <tbody> <tr> <td>Master mode</td> <td>20 Mbps</td> <td>—</td> <td>20 Mbps</td> </tr> <tr> <td>Slave mode</td> <td>20 Mbps</td> <td>20 Mbps</td> <td>20 Mbps</td> </tr> </tbody> </table> <ul style="list-style-type: none"> • The communication clock in master mode can be selected (TXCLK frequency divided by 4, 8 or 16) 		Transmission	Reception	Transmission/Reception	Master mode	20 Mbps	—	20 Mbps	Slave mode	20 Mbps	20 Mbps	20 Mbps
	Transmission	Reception	Transmission/Reception										
Master mode	20 Mbps	—	20 Mbps										
Slave mode	20 Mbps	20 Mbps	20 Mbps										
SSL control	<ul style="list-style-type: none"> • 1 SSL pin (SSL0) per channel • When slave mode is set: pin SSL0 is for input • Delay from SSL assertion to SPCK operation start (SPCK delay) can be set Setting range: 0.5 to 4 cycles of SPCK (in increments of 0.5 per SPCK cycle) • Delay from SPCK operation stop to SSL negation (SSL negation delay) can be set Setting range: 0.5 to 4 cycles of SPCK (in increments of 0.5 per SPCK cycle) • Delay from SSL negation to SSL assertion of the next frame (next frame delay) can be set Setting range: 0.5 to 4 cycles of SPCK (in increments of 0.5 per SPCK cycle) Setting of delay of the next frame is ignored during burst transfer • SSL polarity can be selected • In master mode, burst transfer can be selected (Burst transfer is also enabled during slave transfer depending on the SSL signal of the other master device) 												

Table 26.10 HS-SPI Features

Item	Description
Buffer structure	<ul style="list-style-type: none"> • Transmission and reception: Each 128 bytes
Interrupts	<ul style="list-style-type: none"> • Communication start interrupt (slave only) • Communication complete interrupt • Communication error interrupt (reception parity, transmission underrun, reception overflow) • H-Bus error interrupt
Interrupt sources	<ul style="list-style-type: none"> • Communication start (slave only) • Communication complete • Virtual port reception complete (slave only) • Reception parity error • Transmission underrun error (slave only) • Reception overflow error • H-Bus error
Error detection	<ul style="list-style-type: none"> • Reception parity error • Transmission underrun error (slave only) • Reception overflow error • H-Bus error
Other	<ul style="list-style-type: none"> • A DMAC is incorporated for transfer of communication data • Register write protection • Loop-back testing • Virtual port function (for slave reception)

26.2.2 Block Diagram

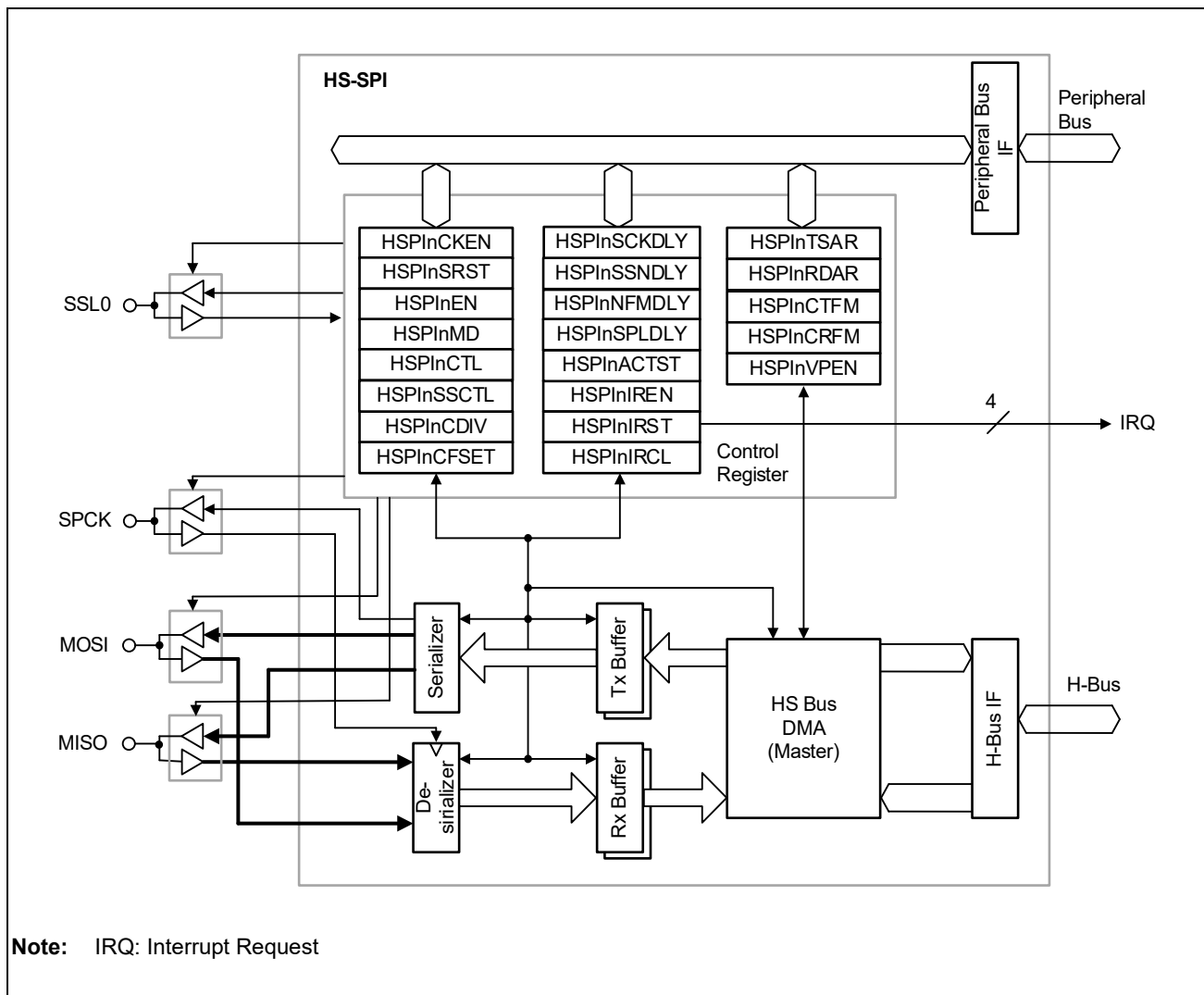


Figure 26.1 Block Diagram of HS-SPI

- The IP configures the control registers from the peripheral bus via the peripheral bus IF.
- The control registers control the DMA master, transmission buffer, serializer, de-serializer, and reception buffer.
- Data for transmission is DMA-transferred via high speed bus from external RAM to the HS-SPI and is stored in the transmission buffer. When data is in the transmission buffer, the data is transmitted to the serializer in sequence, converted into serial data, and output from the serial output pin.
- Received data is input serially, parallelized by the de-serializer, and stored in the reception buffer. When the reception buffer becomes full, the data is DMA-transferred from the HS-SPI to the external RAM via high speed bus. HSPInCFSET[16:0], HSPInMD.FLEN[6:0], HSPInTSAR[31:0], and HSPInRDAR[31:0] bit values determine the number of the frames that fill the transmission and reception buffers.
- In master mode, the HS-SPI transmitter generates the clock and transmits it to the slave.
- In slave mode, the HS-SPI uses a communication clock received from the master for the timing of communication.
- In master mode, the HS-SPI can connect one slave, and communications with the slave to be connected to SSL0.
- In slave mode, the HS-SPI can communicate with the master to which it is connected.

26.3 Registers

26.3.1 List of Registers

Table 26.11 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
HSPIn	Clock Control Register	HSPInCKEN	<HSPIn_base> + 00 _H	8, 16, 32	—	
	Software Reset Register	HSPInSRST	<HSPIn_base> + 04 _H	8, 16, 32	—	
	Communication Enabling Register	HSPInEN	<HSPIn_base> + 08 _H	8, 16, 32	—	
	Communication Mode Register	HSPInMD	<HSPIn_base> + 0C _H	8, 16, 32	—	
	Communication Control Register	HSPInCTL	<HSPIn_base> + 10 _H	8, 16, 32	—	
	SSL Polarity Selection Register	HSPInSSCTL	<HSPIn_base> + 14 _H	8, 16, 32	—	
	SPCK Delay Selection Register	HSPInSCKDLY	<HSPIn_base> + 18 _H	8, 16, 32	—	
	SSL Negation Delay Selection Register	HSPInSSNDLY	<HSPIn_base> + 1C _H	8, 16, 32	—	
	Next Frame Delay Selection Register	HSPInNFMPLY	<HSPIn_base> + 20 _H	8, 16, 32	—	
	Sampling Delay Selection Register	HSPInSPLDLY	<HSPIn_base> + 24 _H	8, 16, 32	—	
	Communication Clock Division Selection Register	HSPInCDIV	<HSPIn_base> + 28 _H	8, 16, 32	—	
	Transmission Data Transfer Source Address Setting Register	HSPInTSAR	<HSPIn_base> + 2C _H	8, 16, 32	—	
	Reception Data Transfer Destination Address Setting Register	HSPInRDAR	<HSPIn_base> + 30 _H	8, 16, 32	—	
	Communication Frame Count Setting Register	HSPInCFSET	<HSPIn_base> + 34 _H	8, 16, 32	—	
	Interrupt Enabling Register	HSPInIREN	<HSPIn_base> + 38 _H	8, 16, 32	—	
	Interrupt Status Register	HSPInIRST	<HSPIn_base> + 3C _H	8, 16, 32	—	
	Interrupt Status Clearing Register	HSPInIRCL	<HSPIn_base> + 40 _H	8, 16, 32	—	
	Operation Status Register	HSPInACTST	<HSPIn_base> + 44 _H	8, 16, 32	—	
	Current Transmission Frame Count Register	HSPInCTFM	<HSPIn_base> + 48 _H	8, 16, 32	—	
	Current Reception Frame Count Register	HSPInCRFM	<HSPIn_base> + 4C _H	8, 16, 32	—	
	Virtual Port Output Enabling Register	HSPInVPEN	<HSPIn_base> + 50 _H	8, 16, 32	—	
	Reserved	—	—	<HSPIn_base> + 5C _H to <HSPIn_base> + 68 _H	—	—
	Test Mode Register	HSPInTESTMD	—	<HSPIn_base> + 6C _H	8, 16, 32	—
Reserved	—	—	<HSPIn_base> + 70 _H to <HSPIn_base> + 7C _H	—	—	

Note: When a reserved area is accessed, read value is undefined. The write value should be 0.

26.3.2 HSPInCKEN — Clock Control Register

- Function: Starts and stops the HS-SPI clock
- Protection: Within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSPInCKEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 26.12 HSPInCKEN Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
0	HSPInCKEN	Clock Operation Enabling 0: ACLK and TXCLK are stopped. 1: ACLK and TXCLK are running.

Rewrite the register while communication is disabled (HSPInEN.HSPInEN bit is 0 and HSPInACTST.HSPInACTF bit is 0).

HSPInCKEN Bit (Clock Operation Enabling)

This bit operates and stops clocks.

The target clocks are ACLK and TXCLK, not PCLK and SPCK.

26.3.3 HSPInSRST — Software Reset Register

- Function: Applies a software reset on HS-SPI
- Protection: Within scope
- Software reset: Off target

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSPInSRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 26.13 HSPInSRST Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
0	HSPInSRST	Software Reset 0: Writing is ignored. 1: Software reset is applied.

Apply a software reset while communication is disabled (the HSPInEN.HSPInEN bit is 0 and the HSPInACTST.HSPInACTF bit is 0).

When accessing other registers of the HS-SPI immediately after a software reset, read the register once before accessing other registers.

HSPInSRST Bit (Software reset)

This bit initializes all registers of the HS-SPI except this register during a software reset.

When read, this bit is always read as 0.

26.3.4 HSPInEN — Communication Enabling Register

- Function: Enables and disables communication
- Protection: Not within scope
- Software reset: Within scope (register cleared to return it to initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	HSPInS TATR _G	—	—	—	—	—	—	—	HSPInE N
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 26.14 HSPInSRST Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
8	HSPInSTATRG	Communication Start Trigger 0: Writing is ignored. 1: Upon completion of communication, start communication with the same settings.
7 to 1	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
0	HSPInEN	Communication Enabling 0: Communication disabled 1: Communication enabled

Start communication by using the communication start trigger bit (HSPInSTATRG) while communication is disabled (the HSPInEN bit is 1 and the HSPInACTST.HSPInACTF bit is 0).

HSPInSTATRG Bit (Communication Start Trigger)

This bit starts communication.

Writing 1 to the bit starts communication with the same setting as the previous transfer.

When the virtual port output is enabled (the HSPInVPEN.HSPInVPEN bit is 1), communication cannot be started using this bit.

HSPInEN Bit (Communication Enabling)

This bit enables or disables communication.

Writing 1 to the bit starts communication.

Writing 0 to the bit during communication stops the communication. If 0 is written to the bit during communication, follow the procedure described in **Section 26.5.6, Suspend Communication Procedure**.

Registers targeted for protection cannot be accessed while communication is enabled.

The state of the HS-SPI is defined as follows by this bit and the HSPInACTST.HSPInACTF bit.

Table 26.15 Definition of the HS-SPI State

HSPInEN	HSPInACTST.ACTF	State of HS-SPI
0	0	Communication prohibited
0	1	Communication suspended
1	0	Communication stopped
1	1	Communication in progress

26.3.5 HSPInMD — Communication Mode Register

- Function: Selects communication mode (Master/slave, duplex, and frame length)
- Protection: Within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0008 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	HSPInFLEN[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	HSPInDPMD [1:0]		—	—	—	HSPInMSTR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W

Table 26.16 HSPInSRST Register Contents

Bit Position	Bit Name	Function
31 to 23	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
22 to 16	HSPInFLEN[6:0]	Frame Length Selection 08 _H : 8 bits 10 _H : 16 bits 18 _H : 24 bits 20 _H : 32 bits 28 _H : 40 bits 30 _H : 48 bits 38 _H : 56 bits 40 _H : 64 bits Settings other than above are prohibited.
15 to 6	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
5, 4	HSPInDPMD[1:0]	Duplex Mode Selection 00: Transmission/reception 01: Transmission 10: Reception (setting prohibited in master mode) 11: Setting prohibited
3 to 1	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
0	HSPInMSTR	Master/Slave Mode Selection 0: Slave mode 1: Master mode

Rewrite the register while communication is disabled (HSPInEN.HSPInEN bit is 0 and HSPInACTST.HSPInACTF bit is 0).

HSPInFLEN[6:0] Bits (Frame Length Selection)

These bits select transfer bit length of a frame. Bit values 8, 16, 24, 32, 40, 48, 56, or 64 can be selected.

HSPInDPMD[1:0] Bits (Duplex Mode Selection)

These bits select the communication direction (transmission, reception, or transmission/reception).
The combination of these bits and the HSPInMSTR bit determines the communication method.

HSPInMSTR Bit (Master/Slave Mode Selection)

This bit selects master or slave mode.
The combination of this bit and the HSPInDPMD[1:0] bits determines the communication method.

26.3.6 HSPInCTL — Communication Control Register

- Function: Selects data format (MSB/LSB first, parity, clock polarity, and clock phase), communication target, and burst operation
- Protection: Within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0010_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	HSPInLSBEN	—	—	HSPInPRSL[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	HSPInBRST	—	—	HSPInCPOL	HSPInC PHA	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R	R	R	R

Table 26.17 HSPInCTL Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 21	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
20	HSPInLSBEN	LSB First Selection 0: MSB first 1: LSB first
19, 18	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
17, 16	HSPInPRSL[1:0]	Parity Selection 00: Transmit data parity bit is not added; parity check is not performed on received data. 01: Communication is performed with even parity 10: Setting prohibited 11: Communications is performed with odd parity
15 to 9	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
8	HSPInBRST	Burst Operation Selection 0: SSL is negated for every frame 1: SSL is negated after completion of transfer of all frames (burst transfer)
7, 6	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
5	HSPInCPOL	Clock Polarity Selection 0: SPCK is Low when idle 1: SPCK is High when idle

Table 26.17 HSPInCTL Register Contents (2/2)

Bit Position	Bit Name	Function
4	HSPInCPHA	Clock Phase Selection 0: Reception sampling on odd-numbered edges; bits shifted out for transmission on even-numbered edges. 1: Bits shifted out for transmission on odd-numbered edges; reception sampling on even-numbered edges
3 to 0	—	Reserved When read, the value after reset is read. When writing, write the value after reset.

Rewrite the register while communication is disabled (the HSPInEN.HSPInEN bit is 0 and the HSPInACTST.HSPInACTF bit is 0).

HSPInLSBEN Bit (LSB First Selection)

This bit selects MSB or LSB first for communications data.

For details, see **Section 26.4.3.3, MSB First and LSB First**.

HSPInPRSL[1:0] Bits (Parity Selection)

These bits select no parity bit, even parity bit, or odd parity bit for communication data.

For details, see **Section 26.4.4, Parity Bit**.

HSPInBRST Bit (Burst Operation Selection)

This bit selects the control method for the SSL signal in master mode.

Setting the bit to 1 enables burst transfer.

For details, see **Section 26.4.11, Burst Operation**.

In slave mode, burst transfer is possible according to the operation of the SSL signal in the master device regardless of the setting of the bit.

HSPInCPOL Bit (Clock Polarity Selection)

This bit selects polarity of communication clock (SPCK).

For details, see **Section 26.4.3.2, Relationship between the Serial Clock and Serial Data**.

HSPInCPHA Bit (Clock Phase Selection)

This bit selects phase of the communication clock (SPCK).

For details, see **Section 26.4.3.2, Relationship between the Serial Clock and Serial Data**.

26.3.7 HSPInSSCTL — SSL Polarity Selection Register

- Function: Selects SSL signal polarity
- Protection: Within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSPInSS0P
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 26.18 HSPInSSCTL Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
0	HSPInSS0P	SSL0 Signal Polarity Selection 0: The SSL0 signal is active low. 1: The SSL0 signal is active high.

Rewrite the register while communications are disabled (the HSPInEN.HSPInEN bit is 0 and the HSPInACTST.HSPInACTF bit is 0).

HSPInSS0P Bit (SSL0 Signal Polarity Selection)

This bit selects the activity level of the SSL0 signal.

In slave mode, it determines the activity level of the slave selection input SSL0 signal.

26.3.8 HSPInSCKDLY — SPCK Delay Selection Register

- Function: Selects length of delay from SSL assertion to SPCK operation start
- Protection: Within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	HSPInSCKDLY[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 26.19 HSPInSCKDLY Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
2 to 0	HSPInSCKDLY[2:0]	SPCK Delay Selection 000: 0.5 cycles of SPCK 001: 1.0 cycle of SPCK 010: 1.5 cycles of SPCK 011: 2.0 cycles of SPCK 100: 2.5 cycles of SPCK 101: 3.0 cycles of SPCK 110: 3.5 cycles of SPCK 111: 4.0 cycles of SPCK

Rewrite the register while communications are disabled (HSPInEN.HSPInEN bit is 0 and HSPInACTST.HSPInACTF bit is 0).

HSPInSCKDLY[2:0] Bits (SPCK Delay Selection)

These bits select length of delay from SSL assertion to SPCK operation start in master mode. In slave mode, the setting of these bits are invalid.

26.3.9 HSPInSSNDLY — SSL Negation Delay Selection Register

- Function: Selects length of delay from SPCK operation stop to SSL negation
- Protection: Within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	HSPInSSNDLY[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 26.20 HSPInSSNDLY Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
2 to 0	HSPInSSNDLY[2:0]	SSL Negation Delay Selection 000: 0.5 cycles of SPCK 001: 1.0 cycle of SPCK 010: 1.5 cycles of SPCK 011: 2.0 cycles of SPCK 100: 2.5 cycles of SPCK 101: 3.0 cycles of SPCK 110: 3.5 cycles of SPCK 111: 4.0 cycles of SPCK

Rewrite the register while communication is disabled (HSPInEN.HSPInEN bit is 0 and HSPInACTST.HSPInACTF bit is 0).

HSPInSSNDLY[2:0] Bits (SSL Negation Delay Selection)

These bits select length of delay from SPCK operation stop to SSL negation in master mode. In slave mode, the setting of the bits is invalid.

26.3.10 HSPInNFMDLY — Next Frame Delay Selection Register

- Function: Selects length of delay from SSL negation to SSL assertion of the next frame.
- Protection: Within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	HSPInNFMDLY[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 26.21 HSPInNFMDLY Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
2 to 0	HSPInNFMDLY[2:0]	Next Frame Delay Selection 000: 0.5 cycles of SPCK 001: 1.0 cycle of SPCK 010: 1.5 cycles of SPCK 011: 2.0 cycles of SPCK 100: 2.5 cycles of SPCK 101: 3.0 cycles of SPCK 110: 3.5 cycles of SPCK 111: 4.0 cycles of SPCK

Rewrite the register while communication is disabled (HSPInEN.HSPInEN bit is 0 and HSPInACTST.HSPInACTF bit is 0).

HSPInNFMDLY[2:0] Bits (Next Frame Delay Selection)

These bits select length of delay from SSL negation to SSL assertion of next frame in master mode.

Setting of the bits is invalid when burst transfer is selected.

In slave mode, the setting of the bits is invalid.

26.3.11 HSPInSPLDLY — Sampling Delay Selection Register

- Function: Selects sampling timing of reception data
- Protection: Within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	HSPInS PLDLY E	—	—	—	—	—	—	HSPInSPLDLY [1:0]	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W

Table 26.22 HSPInSPLDLY Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
8	HSPInSPLDLYE	Sampling Timing Delay Enabling 0: Delay of sampling timing is disabled 1: Delay of sampling timing is enabled
7 to 2	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
1,0	HSPInSPLDLY[1:0]	Selects Sampling Timing Delay 00: Delay of 1 cycle of TXCLK 01: Delay of 2 cycles of TXCLK 10: Setting prohibited. 11: Setting prohibited.

Rewrite the register while communications are disabled (HSPInEN.HSPInEN bit is 0 and HSPInACTST.HSPInACTF bit is 0).

HSPInSPLDLYE Bit (Sampling Timing Delay Enabling)

This bit enables or disables delay of sampling timing of received data in master mode.

In slave mode, the setting of the bits is ignored.

Setting of the bit during loop-back testing is ignored.

HSPInSPLDLY[1:0] Bits (Sampling Timing Delay Selection)

These bits select sampling timing of received data in master mode.

When the HSPInSPLDLYE bit is 1, the setting of the bits is effective.

In slave mode, the setting of the bits is ignored.

Setting of the bits during loop-back testing is ignored.

26.3.12 HSPInCDIV — Communication Clock Division Selection Register

- Function: Selects value of the divided frequency of the communication clock
- Protection: Within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	HSPInCDIV[2:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 26.23 HSPInCDIV Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
2 to 0	HSPInCDIV[2:0]	Communication Clock Frequency Division Selection 000: TXCLK/2 001: TXCLK/4 011: TXCLK/8 111: TXCLK/16 Other settings prohibited.

Rewrite the register while communication is disabled (HSPInEN.HSPInEN bit is 0 and HSPInACTST.HSPInACTF bit is 0).

HSPInCDIV[2:0] Bits (Communication Clock Divided Frequency Selection)

These bits select a value for the divided frequency of the communication clock (SPCK) in master mode or during loop-back testing in slave mode. In slave mode except during loop-back testing, the setting of the bits is ignored.

CAUTION

This setting must be a value that does not exceed the maximum bit rate.
The maximum bit rate is 20 Mbps.

26.3.13 HSPInTSAR — Transmission Data Transfer Source Address Setting Register

- Function: Sets data transfer source addresses for transmission
- Protection: Within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	HSPInTSAR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HSPInTSAR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.24 HSPInTSAR Register Contents

Bit Position	Bit Name	Function
31 to 0	HSPInTSAR[31:0]	Transmission Data Transfer Source Address Setting Address of the transfer source of transmission data

Rewrite the register while communication is disabled (the HSPInEN.HSPInEN bit is 0 and the HSPInACTST.HSPInACTF bit is 0).

HSPInTSAR[31:0] Bits (Transmission Data Transfer Source Address Setting)

These bits set the address of the transfer source of data being transmitted.

Any value can be set*¹. The addresses read from the external memory will be set as follows according to the specified frame length.

- Note 1.** Data transfer source addresses are set in multiples of 8 other than in the Cluster RAM area (The 3 lower-order bit is fixed to 0).
The HS-SPI does not support unaligned transfers of H-Bus other than in the Cluster RAM area.

Table 26.25 Relationship between the Frame Length and the Transmission Data Transfer Source Address

Frame Length	HSPInMD.HSPInFLEN[6:0]	Transmission Data Transfer Source Address
8 bits	08 _H	HSPInTSAR[31:0]
16 bits	10 _H	{HSPInTSAR[31:1], 0 _B } (The 1 lower-order bit is fixed to 0)
24 or 32 bits	18 _H , 20 _H	{HSPInTSAR[31:2], 00 _B } (The 2 lower-order bit is fixed to 0)
40, 48, 56, or 64 bits	28 _H , 30 _H , 38 _H , 40 _H	{HSPInTSAR[31:3], 000 _B } (The 3 lower-order bit is fixed to 0)

The value read from the register is the set value regardless of the address for DMA transfer.

26.3.14 HSPInRDAR — Reception Data Transfer Destination Address Setting Register

- Function: Sets transfer destination address for received data
- Protection: Within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	HSPInRDAR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HSPInRDAR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.26 HSPInRDAR Register Contents

Bit Position	Bit Name	Function
31 to 0	HSPInRDAR[31:0]	Reception Data Transfer Destination Address Setting Assigns destination address for received data

Rewrite the register while communication is disabled (the HSPInEN.HSPInEN bit is 0 and the HSPInACTST.HSPInACTF bit is 0).

HSPInRDAR[31:0] Bits (Reception Data Transfer Destination Address Setting)

These bits set the destination address for received data.

Any value can be set*1. The addresses read from the external memory are set as follows according to the specified frame length.

Note 1. Reception data transfer destination addresses are set in multiples of 8 other than in the Cluster RAM area (The 3 lower-order bit is fixed to 0).

The HS-SPI does not support unaligned transfers of H-Bus other than in the Cluster RAM area.

Table 26.27 Relationship between the Frame Length and the Reception Data Transfer Destination Address

Frame Length	HSPInMD.HSPInFLEN[6:0]	Reception Data Transfer Destination Address
8 bits	08 _H	HSPInRDAR[31:0]
16 bits	10 _H	{HSPInRDAR[31:1], 0 _B } (The 1 lower-order bit is fixed to 0)
24 or 32 bits	18 _H , 20 _H	{HSPInRDAR[31:2], 00 _B } (The 2 lower-order bit is fixed to 0)
40, 48, 56, or 64 bits	28 _H , 30 _H , 38 _H , 40 _H	{HSPInRDAR[31:3], 000 _B } (The 3 lower-order bit is fixed to 0)

The value read from the register is the set value regardless of the address for DMA transfer.

26.3.15 HSPInCFSET — Communication Frame Count Setting Register

- Function: Sets number of communication frames
- Protection: Within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSPInCFSET[16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HSPInCFSET[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.28 HSPInCFSET Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
16 to 0	HSPInCFSET[16:0]	Communication Frame Count Setting Sets the number of communication frames for transmission and reception

Rewrite the register while communication is disabled (the HSPInEN.HSPInEN bit is 0 and the HSPInACTST.HSPInACTF bit is 0).

HSPInCFSET[16:0] Bits (Communication Frame Count Setting)

These bits set the number of communication frames for transmission and reception.

Setting range: up to 64-K frame (00001_H to 10000_H)

Settings outside the range are prohibited.

In slave reception, while the virtual port output is enabled (the HSPInVPEN.HSPInVPEN bit is 1), setting of the register is invalid.

(The set value is ignored.)

26.3.16 HSPInIREN — Interrupt Enabling Register

- Function: Enables and disables output of interrupt requests
- Protection: Within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	HSPInVPE	—	—	HSPInBEE	HSPInPEE	HSPInROVFEE	HSPInTUDREE	HSPInCSTAE	HSPInCENDE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.29 HSPInIREN Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
8	HSPInVPE	Virtual Port Reception Complete Interrupt Enabling 0: Disables output of a communication complete interrupt request once virtual port reception completed 1: Enables output of a communication complete interrupt request once virtual port reception completed
7 to 6	—	When read, the value after reset is returned. When writing, write the value after reset.
5	HSPInBEE	H-Bus Error Interrupt Enabling 0: Disables output of H-Bus error interrupt requests 1: Enables output of H-Bus error interrupt requests
4	HSPInPEE	Reception Parity Error Interrupt Enabling 0: Disables output of a communication error interrupt request when a reception parity error occurs 1: Enables output of a communication error interrupt request when a reception parity error occurs
3	HSPInROVFEE	Reception Overflow Error Interrupt Enabling 0: Disables output of a communication error interrupt request when a reception overflow error occurs 1: Enables output of a communication error interrupt request when a reception overflow error occurs
2	HSPInTUDREE	Transmission Underrun Error Interrupt Enabling 0: Disables output of a communication error interrupt request when a communication underrun error occurs 1: Enables output of a communication error interrupt request when a communication underrun error occurs
1	HSPInCSTAE	Communication Start Interrupt Enabling 0: Disables output of a communication start interrupt request 1: Enables output of a communication start interrupt request

Table 26.29 HSPInIREN Register Contents (2/2)

Bit Position	Bit Name	Function
0	HSPInCENDE	Completed Communication Interrupt Enabling 0: Disables output of a communication complete interrupt requests once a communication is completed 1: Enables output of a communication complete interrupt requests once a communication is completed

Rewrite the register while communications are disabled (HSPInEN.HSPInEN bit is 0 and HSPInACTST.HSPInACTF bit is 0).

HSPInVPE Bit (Virtual Port Reception Complete Interrupt Enabling)

This bit enables or disables output of a communication complete request once virtual port reception completed.

HSPInBEE Bit (H-Bus Error Interrupt Enabling)

This bit enables or disables output of the H-Bus error interrupt request.

HSPInPEE Bit (Reception Parity Error Interrupt Enabling)

This bit enables or disables output of a communication error interrupt request when a reception parity error occurs.

HSPInROVFE Bit (Reception Overflow Error Interrupt Enabling)

This bit enables or disables output of a communication error interrupt request when a reception overflow error occurs.

HSPInTUDREE Bit (Transmission Underrun Error Interrupt Enabling)

This bit enables or disables output of a communication error interrupt request when a transmission underrun error occurs.

Transmission underrun errors only occur in slave mode.

HSPInCSTAE Bit (Communication Start Interrupt Enabling)

This bit enables or disables output of a communication start interrupt request.

Communication start interrupt requests are only issued in slave mode.

HSPInCENDE Bit (Completed Communication Interrupt Enabling)

This bit enables or disables output of a completed communication interrupt request once a communication is completed.

26.3.17 HSPInIRST — Interrupt Status Register

- Function: Reports Interrupt status
- Protection: Not within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	HSPInV PF	—	—	HSPInB EF	HSPInP EF	HSPInR OVFEF	HSPInT UDREF	HSPInC STAF	HSPInC ENDF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.30 HSPInIRST Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is read.
8	HSPInVPF	Virtual Port Reception Complete Flag 0: Reception through the virtual port is in progress. 1: Reception through the virtual port has been complete.
7 to 6	—	Reserved When read, the value after reset is read.
5	HSPInBEF	H-Bus Error Flag 0: No H-Bus error has occurred 1: An H-Bus error has occurred
4	HSPInPEF	Reception Parity Error Flag 0: No reception parity error has occurred 1: A reception parity error has occurred
3	HSPInROVFEF	Reception Overflow Error Flag 0: No reception overflow error has occurred 1: A reception overflow error has occurred
2	HSPInTUDREF	Transmission Underrun Error Flag (slave mode only) 0: No transmission underrun error has occurred 1: A transmission underrun error has occurred
1	HSPInCSTAF	Communication Start Flag (slave mode only) 0: Waits for communication to start 1: Start of communication detected
0	HSPInCENDF	Communication Complete Flag 0: Communication in progress 1: Communication completed

When starting communication, clear the respective interrupt status flags to 0 and enable communication (HSPInEN.HSPInEN bit is 1), or start communication by using the communication start trigger (HSPInEN.HSPInSTATRG bit is 1).

HSPInVPF Flag (Virtual Port Reception Completion)

This flag indicates that the received data has been transferred to the virtual port.

When the flag is 1, a communication complete interrupt request (INT_HSPI_CEND) is not output.

[Setting condition]

When all of the conditions below are satisfied:

- When one-frame of data received has been transferred to the virtual port during slave reception.
- When the HSPInEN.HSPInEN bit is 1
- When the HSPInVPEN.HSPInVPEN bit is 1

[Clearing condition]

When the condition below is satisfied:

- When 1 is written to the virtual port reception completed flag clearing bit (HSPInIRCL.HSPInVPC)

HSPInBEF Flag (H-Bus Error)

This flag indicates that there is an H-Bus error response during a DMA transfer.

While the flag is 1, subsequent H-Bus error interrupt requests (INT_HSPI_BERR) are not output.

[Setting condition]

When all of the conditions below are satisfied:

- When an H-Bus error response is received during a DMA transfer
- When the HSPInEN.HSPInEN bit is 1
- When the HSPInVPEN.HSPInVPEN bit is 0

[Clearing condition]

When the condition below is satisfied:

- When 1 is written to the H-Bus error flag clearing bit (HSPInIRCL.HSPInBEC)

HSPInPEF Flag (Reception Parity Error)

This flag indicates that there is a parity error in serial received data.

While the flag is 1, subsequent communication error interrupt requests (INT_HSPI_ERR) are not output.

[Setting condition]

When all of the conditions below are satisfied:

- When parity error data is received in a serial transfer
- When the HSPInEN.HSPInEN bit is 1

[Clearing condition]

When the condition below is satisfied:

- When 1 is written to the reception parity error flag clearing bit (HSPInIRCL. HSPInPEC)

HSPInROVFEF Flag (Reception Overflow Error)

This flag indicates that the reception buffer has overflowed.

While the flag is 1, subsequent communication error interrupt requests (INT_HSPI_ERR) are not output.

[Setting condition]

When all of the conditions below are satisfied:

- When the reception buffer has overflowed.
- When the HSPInEN.HSPInEN bit is 1

- When the HSPInVPEN.HSPInVPEN bit is 0

[Clearing condition]

When the condition below is satisfied:

- When 1 is written to the reception overflow error flag clearing bit (HSPInIRCL.HSPInROVFEC)

HSPInTUDREF Flag (Transmission Underrun Error)

This flag indicates that correct data is not output because preparation of output of data for transmission is not completed within allotted time in slave mode.

While the flag is 1, subsequent communication error interrupt requests (INT_HSPI_ERR) are not output.

[Setting condition]

When all of the conditions below are satisfied:

- When data for transmission is not correctly output in slave mode
- When the HSPInEN.HSPInEN bit is 1
- When the HSPInVPEN.HSPInVPEN bit is 0

[Clearing condition]

When the condition below is satisfied:

- When 1 is written to the transmission underrun error flag clearing bit (HSPInIRCL.HSPInTUDREC)

HSPInCSTAF Flag (Communication Start)

This flag indicates that communication has started in slave mode.

While the flag is 1, communication start interrupt requests (INT_HSPI_CSTART) are not output.

[Setting conditions]

When all of the conditions below are satisfied:

- When an assertion of SSL in the first frame is detected in slave mode
- When the HSPInEN.HSPInEN bit is 1
- When the HSPInVPEN.HSPInVPEN bit is 0

[Clearing condition]

When the condition below is satisfied:

- When 1 is written to the communication start flag clearing bit (HSPInIRCL.HSPInCSTAC)

HSPInCENDF Flag (Communication Completion)

This flag indicates that communication has been completed.

While either one of the H-Bus error flag (HSPInBEF), reception overflow error flag, (HSPInROVFEC), transmission underrun error flag, (HSPInTUDREF), or this flag is 1, a communication complete interrupt request (INT_HSPI_CEND) is not output.

[Setting condition]

When all of the conditions below are satisfied:

- When the number of frames specified in the Communication Frame Count Setting Register (HSPInCFSET) are transferred, the conditions for transfer completion are as follows.
 - Transmission/reception: When DMA transfer of the received data is complete
 - Transmission: When transfer of the data for transmission is complete
 - Reception: When DMA transfer of the data for reception is complete
- When the HSPInBEF flag is 0, the HSPInROVFEC flag is 0 and the HSPInTUDREF flag is 0

- When the HSPInEN.HSPInEN bit is 1
- When the HSPInVPEN.HSPInVPEN bit is 0

[Clearing condition]

When the condition below is satisfied:

- When 1 is written to the communication complete flag clearing bit (HSPInIRCL.HSPInCENDC)

26.3.18 HSPInRCL — Interrupt Status Clearing Register

- Function: Clears the interrupt status flag
- Protection: Not within scope
- Software reset: Not within scope

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	HSPInVPC	—	—	HSPInBEC	HSPInPEC	HSPInROVFEC	HSPInTUDREC	HSPInCSTAC	HSPInCENDC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.31 HSPInRCL Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
8	HSPInVPC	Virtual Port Reception Complete Clearing 0: Writing to this bit is ignored. 1: The virtual port reception complete flag is cleared to 0.
7 to 6	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
5	HSPInBEC	H-Bus Error Flag Clearing 0: Writing to this bit is ignored. 1: The H-Bus error flag is cleared to 0
4	HSPInPEC	Reception Parity Error Flag Clearing 0: Writing to this bit is ignored. 1: The reception parity error flag is cleared to 0
3	HSPInROVFEC	Reception Overflow Error Flag Clearing 0: Writing to this bit is ignored. 1: The reception overflow error flag is cleared to 0
2	HSPInTUDREC	Transmission Underrun Error Flag Clearing 0: Writing to this bit is ignored. 1: The transmission underrun error flag is cleared to 0
1	HSPInCSTAC	Communication Start Flag Clearing 0: Writing to this bit is ignored. 1: The communication start flag is cleared to 0
0	HSPInCENDC	Communication Complete Flag Clearing 0: Writing to this bit is ignored. 1: The communication complete flag is cleared to 0

When read, this register is always read as 0000 0000_H.

HSPInVPC Bit (Virtual Port Reception Complete Flag Clearing)

This bit clears the virtual port reception complete flag (HSPInIRST.HSPInVPF) to 0.

HSPInBEC Bit (H-Bus Error Flag Clearing)

This bit clears the H-Bus error flag (HSPInIRST.HSPInBEF) to 0.

HSPInPEC Bit (Reception Parity Error Flag Clearing)

This bit clears the reception parity error flag (HSPInIRST.HSPInPEF) to 0.

HSPInROVFEC Bit (Reception Overflow Error Flag Clearing)

This bit clears the reception overflow error flag (HSPInIRST.HSPInROVFEF) to 0.

HSPInTUDREC Bit (Transmission Underrun Error Flag Clearing)

This bit clears the transmission underrun error flag (HSPInIRST.HSPInTUDREF) to 0.

HSPInCSTAC Bit (Communication Start Flag Clearing)

This bit clears the communication start flag (HSPInIRST.HSPInCSTAF) to 0.

HSPInCENDC Bit (Communication Complete Flag Clearing)

This bit clears the communication complete flag (HSPInIRST.HSPInCENDF) to 0.

26.3.19 HSPInACTST — Operation Status Register

- Function: Indicates operating status of the HS-SPI
- Protection: Not within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSPInACTF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.32 HSPInACTST Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is read.
0	HSPInACTF	Operation Status Flag 0: Inactive 1: Active

HSPInACTF Flag (Operation Status Flag)

This flag indicates the operating status of the HS-SPI.

[Setting conditions]

- Normal communication: When DMA transfer, serial reception or serial transmission is in progress after start communication by using the HSPInEN.HSPInEN bit or the HSPInEN.HSPInSTATRG bit
- Virtual port communication: When the HSPInEN.HSPInEN bit is 1

[Clearing conditions]

- Normal communication (transmission/reception or reception): When all frames specified in the HSPInCFSET register have been DMA-transferred
- Normal communication (transmission): When all frames specified in the HSPInCFSET register have been serially transmitted
- Normal communication (suspension): When DMA transfer and serial transmission stop after the HSPInEN.HSPInEN bit is set to 0
- Virtual port communication: When the HSPInEN.HSPInEN bit is 0

26.3.20 HSPInCTFM — Current Transmission Frame Count Register

- Function: Indicates number of current communication frames
- Protection: Not within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSPInCTFM[16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HSPInCTFM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.33 HSPInCTFM Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is read.
16 to 0	HSPInCTFM[16:0]	Current Transmission Frame Count The number of current communication frames (00000 _H to 10000 _H)

HSPInCTFM[16:0] Bits (Current Transmission Frame Count)

These bits indicate the number of the current communication frames.

The register indicates the number of frames that have been read from an external memory by DMA transfer.

The register is cleared to 0000 0000_H 1 PCLK + 1 ACLK cycles after communication is disabled (the HSPInEN.HSPInEN bit is 0).

The register is cleared to 0000 0000_H 1 PCLK + 1 ACLK cycles after communication is started by the communication start trigger bit (the HSPInEN.HSPInSTATRG bit is 1).

26.3.21 HSPInCRFM — Current Reception Frame Count Register

- Function: Indicates the number of frames currently being received
- Protection: Not within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSPInC RFM [16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HSPInCRFM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.34 HSPInCRFM Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is read.
16 to 0	HSPInCRFM[16:0]	Current Received Frame Count The current number of frames received (00000 _H to 10000 _H)

HSPInCRFM[16:0] Bits (Current Received Frame Count)

These bits indicate the number of the current received frames.

The register indicates the number of frames that have been written to an external memory by DMA transfer.

The register is cleared to 0000 0000_H 1 PCLK + 1 ACLK cycles after communication is disabled (the HSPInEN.HSPInEN bit is 0).

The register is cleared to 0000 0000_H 1 PCLK + 1 ACLK cycles after communication is started by using the communication start trigger bit (the HSPInEN.HSPInSTATRG bit is 1).

26.3.22 HSPInVPEN — Virtual Port Output Enabling Register

- Function: Enables output of received data to the virtual port
- Protection: Within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSPInVPEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 26.35 HSPInVPEN Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
0	HSPInVPEN	Virtual Port Output Enabling 0: Writes received data to an external memory 1: Outputs the received data to the virtual port (effective only in slave mode)

Rewrite the register while communication is disabled (the HSPInEN.HSPInEN bit is 0 and the HSPInACTST.HSPInACTF bit is 0).

HSPInVPEN Bit (Virtual Port Output Enabling)

This bit enables or disables output of the data received to the virtual port in slave reception mode.

Setting this bit to 1 in slave reception mode outputs the serial received data to the virtual port.

At this time, no received data is written to an external memory.

Set the bit to 0 when writing the received data to an external memory.

This is valid only in slave reception mode while loop-back testing mode is not in use. Writing the bit to 1 in other modes (master mode when loop-back testing mode is not used, slave transmission mode, or slave transmission/reception mode) is also ignored (with no operation of the virtual port, in normal operation).

The bit is ignored when the loop-back function is operating (the operation becomes the same as when the HSPInVPEN bit is 0).

26.3.23 HSPInTESTMD — Test Mode Register

- Function: Selects normal operation or loop-back testing operation
- Protection: Within scope
- Software reset: Within scope (cleared to return it to its initial value)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSPInLPBT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 26.36 HSPInTESTMD Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
0	HSPInLPBT	Loop-back Testing Mode 0: Normal operation 1: Loop-back testing operation

Rewrite the register while communication is disabled (the HSPInEN.HSPInEN bit is 0 and the HSPInACTST.HSPInACTF bit is 0).

HSPInLPBT Bit (Loop-back Testing Mode)

This bit controls loop-back testing mode for receiving data for transmission of the HS-SPI.

Setting this bit to 1 shifts to loop-back testing mode regardless of the value of the HSPInMD.HSPInDPMD[1:0] bits.

A loop-back test can be run in both master and slave modes.

In slave mode, however, the test is operated with an internally generated clock without using the communications clock (SPCK).

The registers that are valid in loop-back testing mode are as follows.

- Clock Control Register (HSPInCKEN)
- Software Reset Register (HSPInSRST)
- Communication Enabling Register (HSPInEN)
- Communication Mode Register (HSPInMD)
- Communication Control Register (HSPInCTL)
- SSL Polarity Selection Register (HSPInSSCTL)
- SPCK Delay Selection Register (HSPInSCKDLY)
- SSL Negation Delay Selection Register (HSPInSSNDLY)

- Next Frame Delay Selection Register (HSPInNFMDLY)
- Communication Clock Division Selection Register (HSPInCDIV)
- Transmission Data Transfer Source Address Setting Register (HSPInTSAR)
- Reception Data Transfer Destination Address Setting Register (HSPInRDAR)
- Communication Frame Count Setting Register (HSPInCFSET)
- Interrupt Enabling Register (HSPInIREN)
- Interrupt Status Register (HSPInIRST)
- Interrupt Status Clearing Register (HSPInIRCL)
- Operation Status Register (HSPInACTST)
- Current Transmission Frame Count Register (HSPInCTFM)
- Current Reception Frame Count Register (HSPInCRFM)

26.4 Detailed Functional Descriptions

26.4.1 Overview of the HS-SPI Operation

The HS-SPI handles full- and half-duplex clock synchronous serial communications.

There are five communication modes: master transmission mode, master transmission/reception mode, slave transmission mode, slave transmission/reception mode, and slave reception mode.

When starting the HS-SPI transfer, the pin settings must be finished in advance.

For details on pin settings, see **Section 2, Pin Function**.

26.4.1.1 Master Transmission/Reception Mode

Operation of the HS-SPI in master transmission/reception mode is as follows. For the setting procedure, see **Section 26.5.2.1, Transmission and Reception Setting Flows**.

1. Set master/slave mode selection bit (HSPInMD.HSPIn.MSTR) to 1.
2. Set the communication enabling bit (HSPInEN.HSPInEN) to 1.
3. Use the DMAC in the HS-SPI to read the data for transmission from the address specified in the Transmission Data Transfer Source Address Setting Register (HSPInTSAR) and store that data in the transmission buffer.
4. Once the data for transmission has been transferred from the transmission buffer to the transmission block (serializer), transmission proceeds.
5. The SSL signal is asserted and the transmission clock is output from the SPCK pin. Data is output from the MOSI pin, synchronized with the SPCK clock.
6. Reception starts in response to the SPCK clock start.
7. The receiving block (de-serializer) stores the frame of data received from the MISO input pin in the reception buffer.
8. The DMAC in the HS-SPI transfers received data from the reception buffer to the address specified in the Reception Data Transfer Destination Address Setting Register (HSPInRDAR).
9. Once the number of frames specified in the Communication Frame Count Setting Register (HSPInCFSET) have been transferred, the SSL0 output signal is negated, and once the number of frames specified in the Communication Frame Count Setting Register (HSPInCFSET) have been transferred, a communication complete interrupt request (INT_HSPI_CEND) is output.

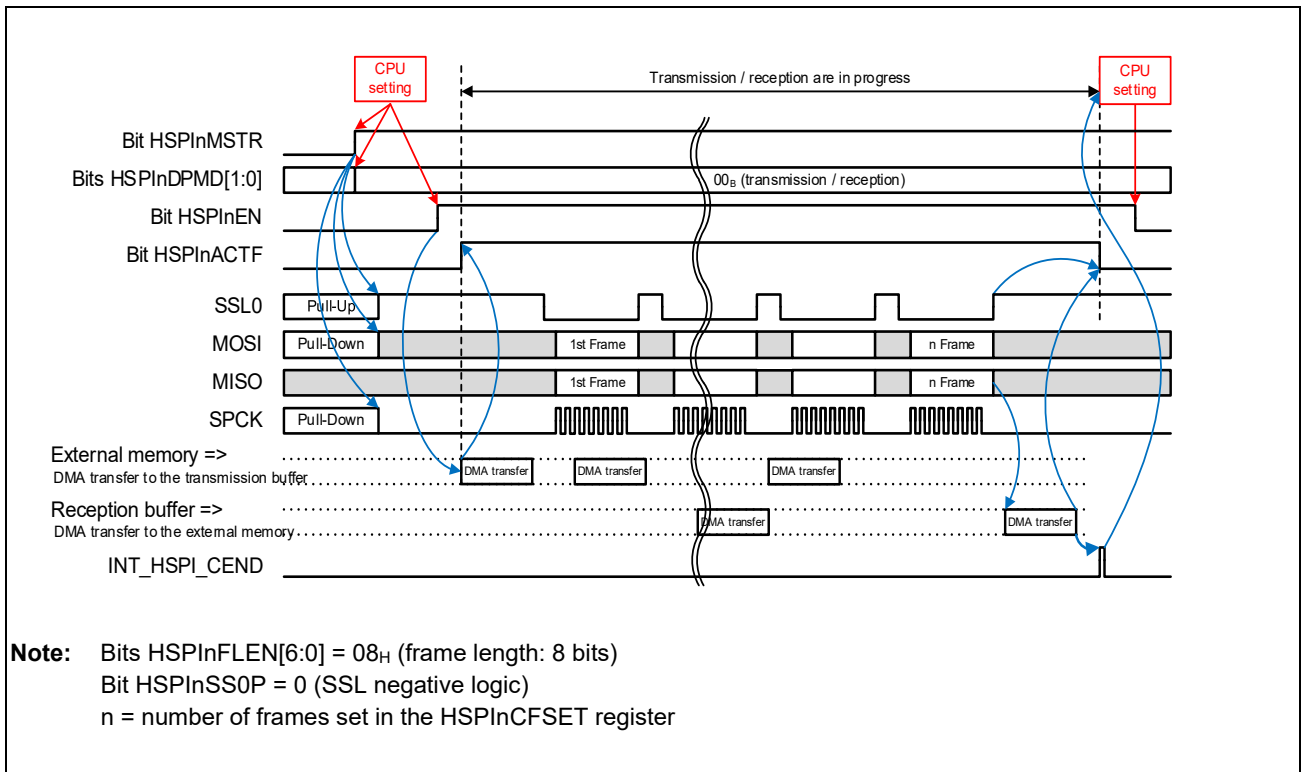


Figure 26.2 Operation of Master Transmission/Reception

26.4.1.2 Master Transmission Operation

Operation of HS-SPI in master transmission mode is as follows. For the setting procedure, see **Section 26.5.2.2, Transmission Setting Flow**.

1. Set the master/slave mode selection bit (HSPInMD.HSPIn.MSTR) to 1.
2. Set the communication enabling bit (HSPInEN.HSPInEN) to 1.
3. Use the DMAC in the HS-SPI to read the data for transmission from the address specified by the Transmission Data Transfer Source Address Setting Register (HSPInTSAR) and store this data in the transmission buffer.
4. Once the data for transmission has been transferred from the transmission buffer to the transmission block (Serializer), transmission proceeds.
5. The SSL signal is asserted and the transmission clock is output from the SPCK pin. Bits are output from the MOSI pin, synchronized with the SPCK clock.
6. Once the number of frames specified in the Communication Frame Count Setting Register (HSPInCFSET) have been transferred, the SSL0 signal is negated and a communication complete interrupt request (INT_HSPI_CEND) is output.

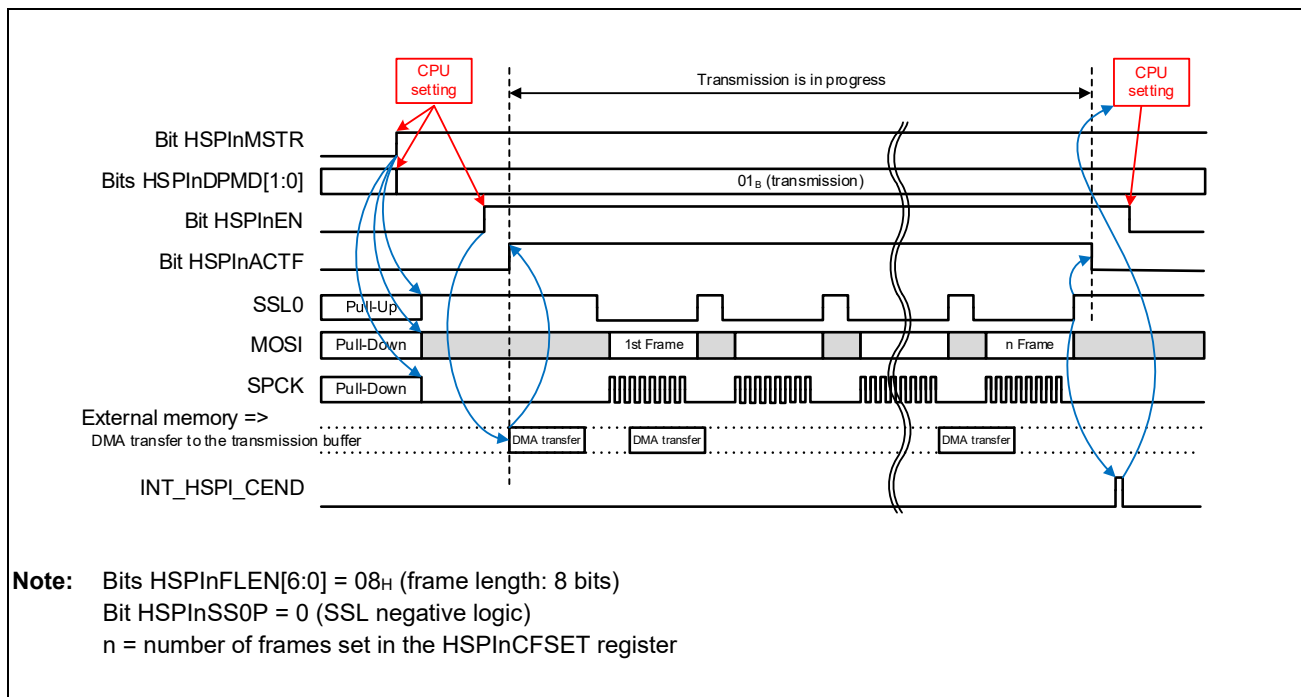


Figure 26.3 Master Transmission Operation Example

26.4.1.3 Slave Transmission/Reception

Operation of the HS-SPI in slave transmission/reception mode is as follows. For the setting procedure, see **Section 26.5.3.1, Transmission and Reception Setting Flows**.

1. Set the communication enabling bit (HSPInEN.HSPInEN) to 1.
2. Use the DMAC in HS-SPI to read data for transmission from the address specified in the Transmission Data Transfer Source Address Setting Register (HSPInTSAR) and store this data in the transmission buffer.
3. Once the data for transmission has been transferred from the transmission buffer to the transmission block (serializer), transmission proceeds.
4. During transmission, data is output from the MISO output pin, synchronized with the SPCK clock (the SPCK input pin) for the period over which SSL0 input pin is asserted.
5. Reception also proceeds in response to the SPCK clock (SPCK input pin) for the period over which the SSL0 input pin is asserted.
6. The receiving block (de-serializer) stores data frames received from the MISO input pin in the reception buffer.
7. The DMAC in the HS-SPI transfers received data from the reception buffer to the address specified in the Reception Data Transfer Destination Address Setting Register (HSPInRDAR).
8. Once the number of frames specified in the Communication Frame Count Setting Register (HSPInCFSET) have been transmitted, and once the number of frames of received data specified in the Communication Frame Count Setting Register (HSPInCFSET) have been DMA-transferred, a communication complete interrupt request (INT_HSPI_CEND) is output.

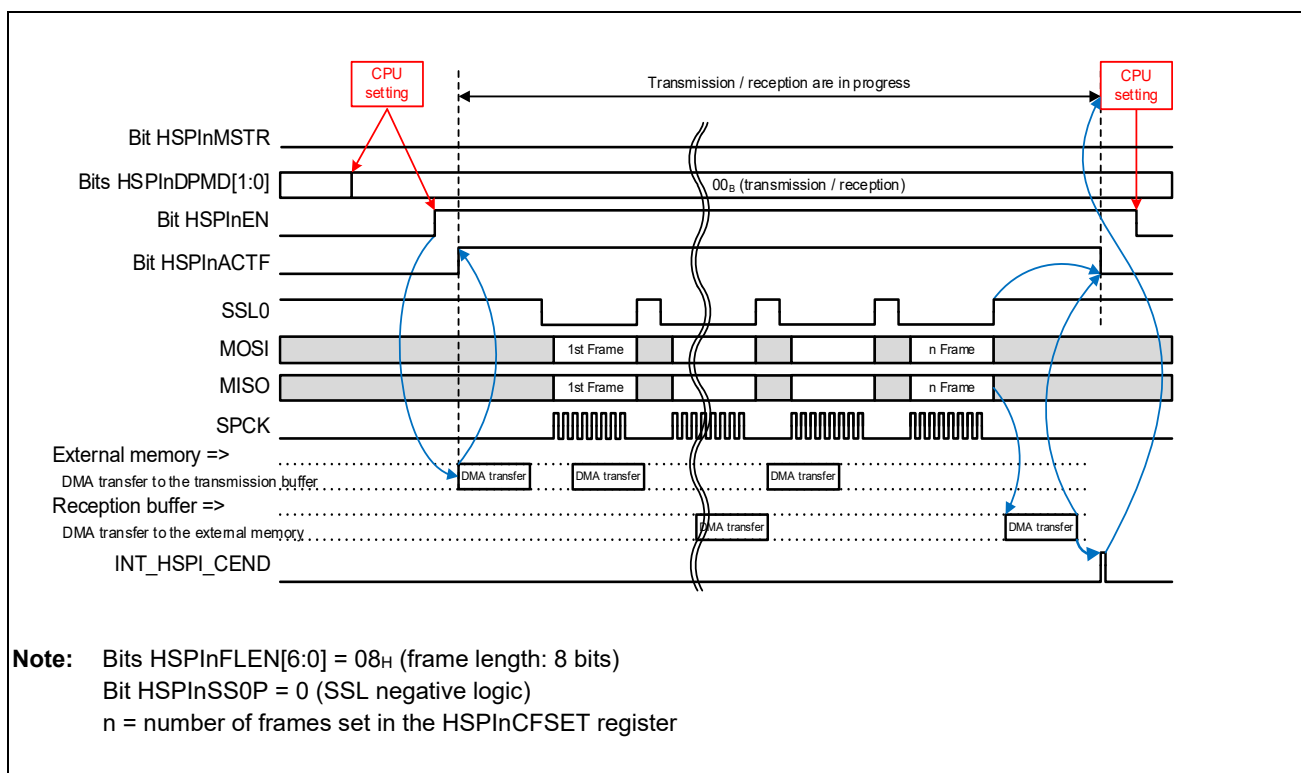


Figure 26.4 Slave Transmission/Reception Operation Example

26.4.1.4 Slave Transmission

Operation of the HS-SPI in slave transmission mode is as follows. For the setting procedure, see **Section 26.5.3.2, Transmission Setting Flow**.

1. Set communication enabling bit (HSPInEN.HSPInEN) to 1.
2. Use the DMAC in HS-SPI to read data for transmission from the address specified in the Transmission Data Transfer Source Address Setting Register (HSPInTSAR) and store this data in the transmission buffer.
3. Once the data for transmission has been transferred from the transmission buffer to the transmission block (serializer), transmission proceeds.
4. During transmission, data is output from the MISO output pin, synchronized with the SPCK clock (the SPCK input pin) for the period over which SSL0 input pin is asserted.
5. Once the number of frames specified in the Communication Frame Count Setting Register (HSPInCFSET) have been transferred, a communication complete interrupt request (INT_HSPI_CEND) is output.

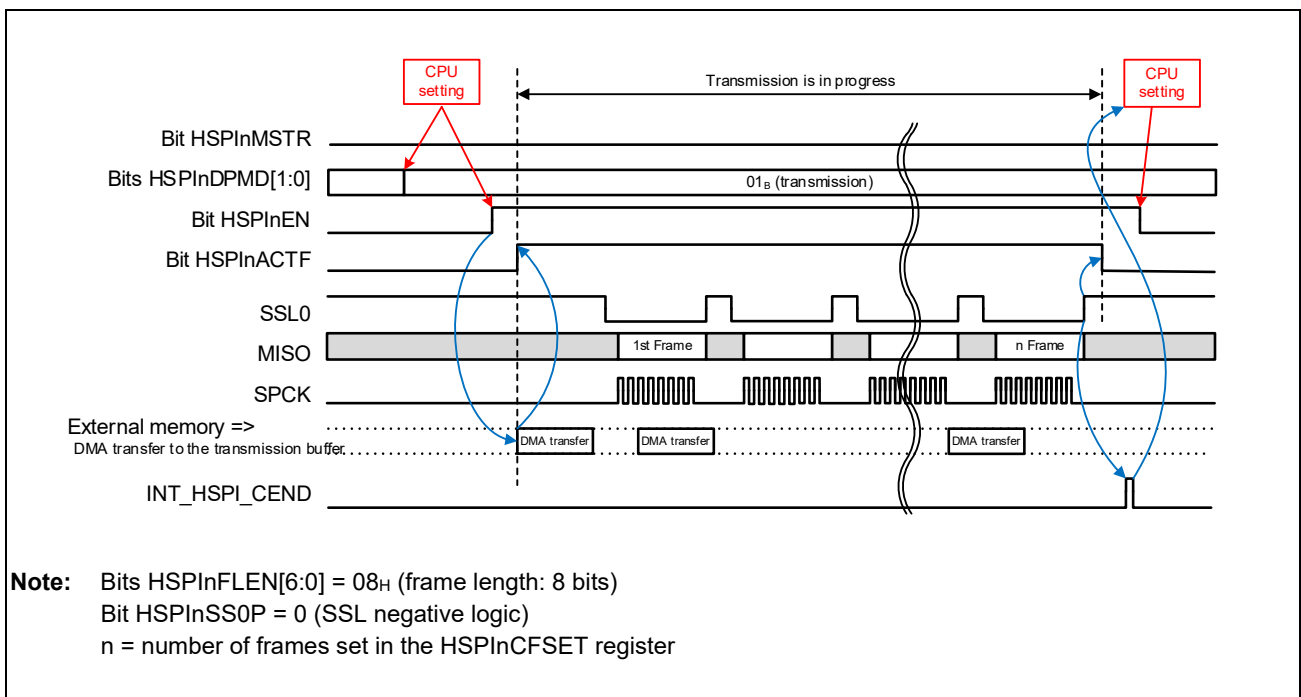


Figure 26.5 Slave Transmission Example

26.4.1.5 Slave Reception

Operation of HS-SPI in slave reception mode is as follows. For the setting procedure, see **Section, 26.5.3.3, Reception Setting Flow**.

1. Set the communications enabling bit (HSPInEN.HSPInEN) to 1.
2. During the wait-for-reception state: Reception begins once SPCK clock (SPCK input pin) starts and continues for the period over which the SSL0 input signal is asserted.
3. The receiving block (de-serializer) stores the frame data received from the MOSI input pin in the reception buffer.
4. The DMAC in HS-SPI transfers received data from the reception buffer to the address specified in the Reception Data Transfer Destination Address Setting Register (HSPInRDAR).
5. Once the number of frames specified in the Communication Frame Count Setting Register (HSPInCFSET) have been transferred, a communication complete interrupt request (INT_HSPI_CEND) is output.

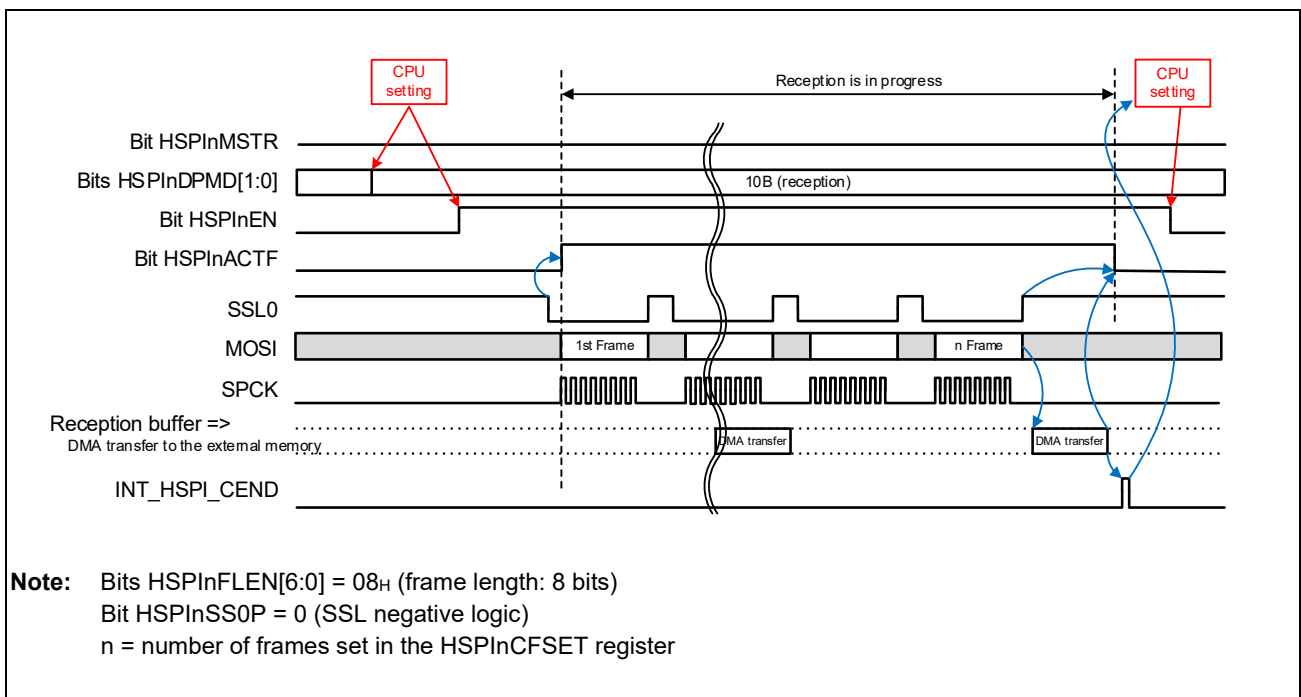


Figure 26.6 Slave Reception Example

26.4.1.6 Slave Reception (When the Virtual Port Output is Enabled)

Operation of HS-SPI in slave reception mode is as follows. For the setting procedure, see **Section 26.5.3.4, Virtual Port Reception Setting Flow**.

1. Set the communications enabling bit (HSPInEN.HSPInEN) to 1.
2. During the wait-for-reception state: Reception begins once SPCK clock (SPCK input pin) starts and continues for the period over which the SSL0 input pin is asserted.
3. The receiving block (de-serializer) transfers data frames received on the MOSI input pin to the virtual port in frame units.
In this case, the received data frames are not stored in the reception buffer and are not DMA-transferred to the external memory.
4. Once the data frames have been transferred to the virtual port, a communication complete interrupt request (INT_HSPI_CEND) is output.

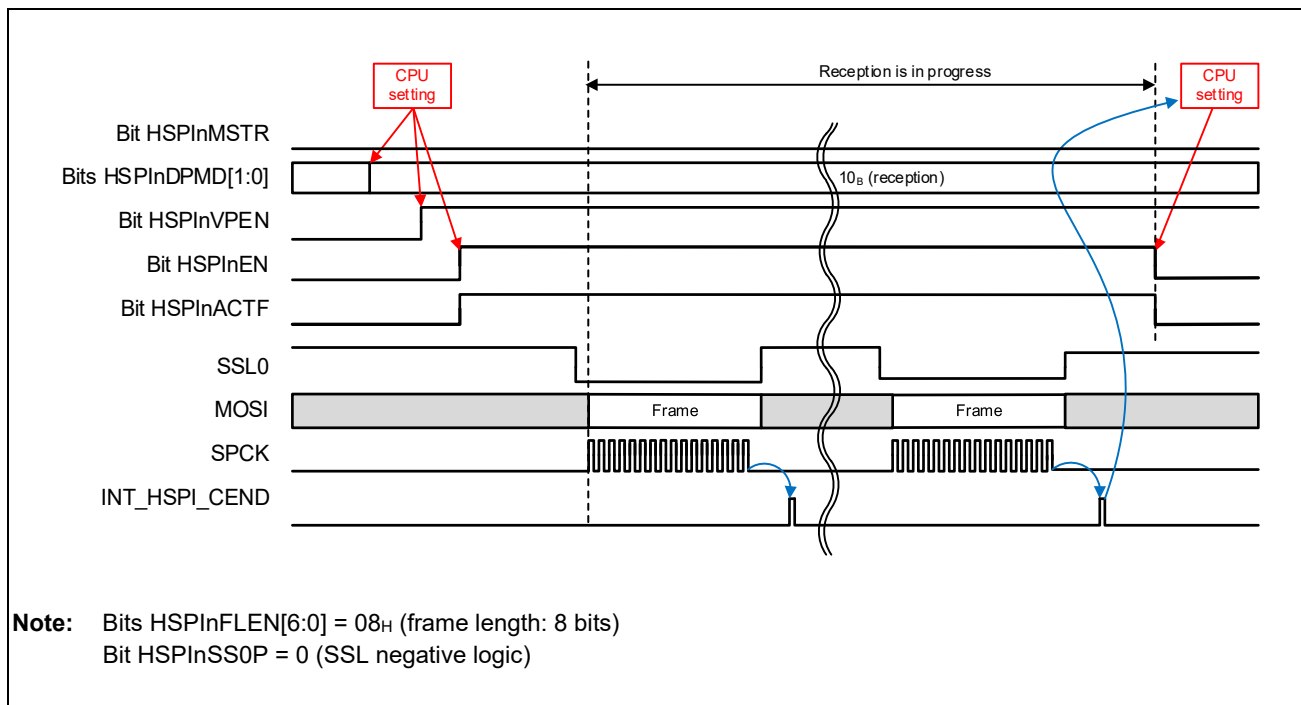


Figure 26.7 Slave Reception Example (When the Virtual Port Output is Enabled)

26.4.2 Control on the HS-SPI Pins

Table 26.37 lists controls on the pins of the HS-SPI.

Table 26.37 Relationship between Communication Mode and Pins

Mode	Communication Mode Register (HSPInMD)		State of Pin			
	HSPInMSTR	HSPInDPMD[1:0]	SSL0	SPCK	MOSI	MISO
Master transmission/ reception	1	00	Output	Output	Output	Input
Master transmission		01	Output	Output	Output	Unused
Slave transmission/ reception	0	00	Input	Input	Input	Output* ¹
Slave transmission		01	Input	Input	Unused	Output* ¹
Slave reception		10	Input	Input	Input	Unused

Note 1. State of MISO pin is output while the HSPInEN.HSPInEN bit is 1 and the SSL0 signal is asserted. (State of MISO pin is Hi-Z while the SSL0 signal is negated.)

26.4.3 Data Format

26.4.3.1 Frame Data Format

The frame data format is described below.

For details on parity, see **Section 26.4.4, Parity Bit**.

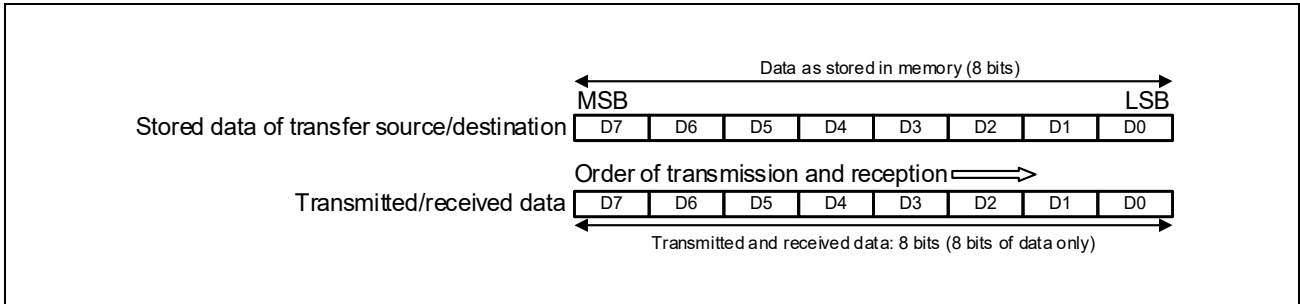


Figure 26.8 Data Length for Transfer: 8 bits, No Parity Bit, MSB First, Transmission/Reception

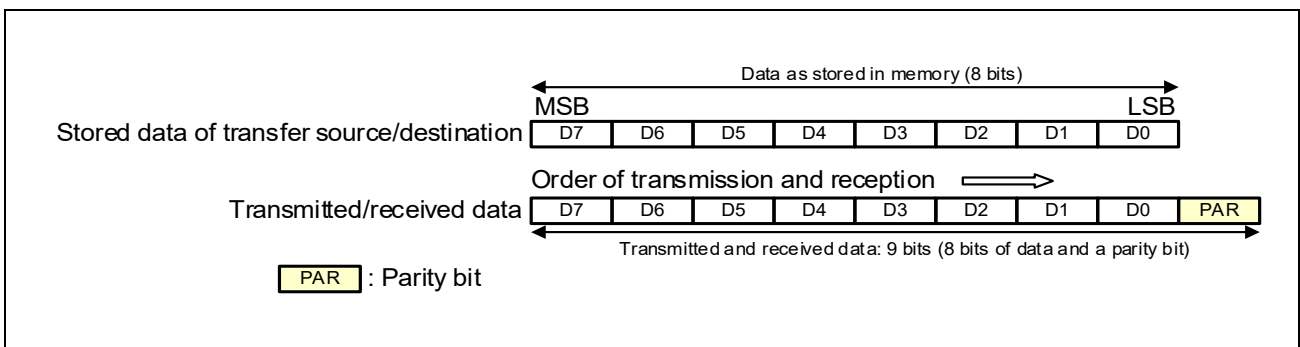


Figure 26.9 Data Length for Transfer: 8 Bits, Parity Bit, MSB first, Transmission/Reception

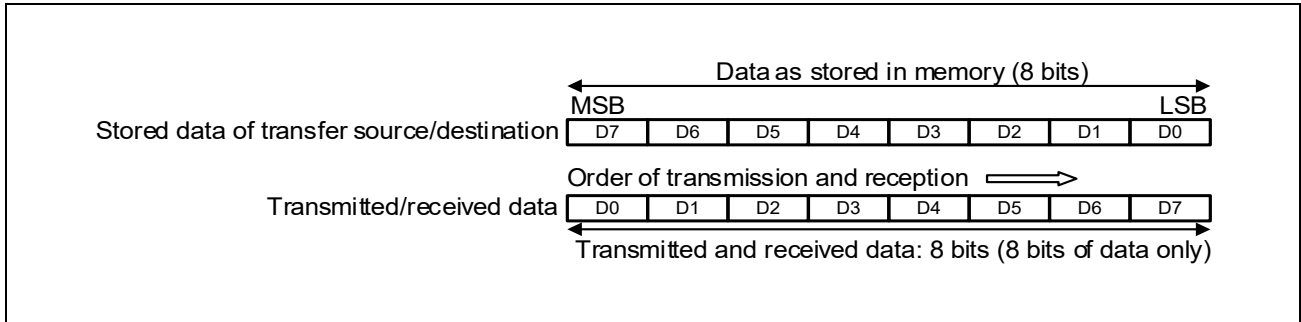


Figure 26.10 Data Length for Transfer: 8 Bits, No Parity Bit, LSB First, Transmission/Reception

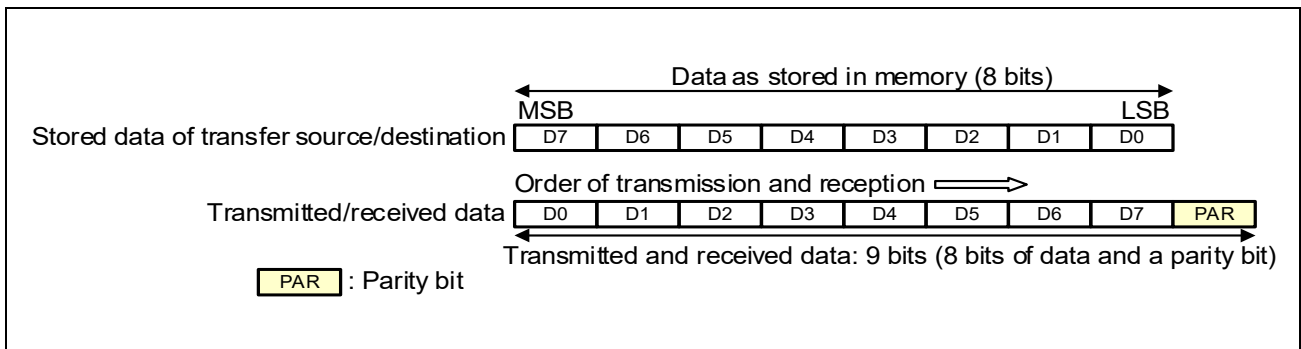


Figure 26.11 Data Length for Transfer: 8 Bits, Parity Bit, LSB First, Transmission/Reception

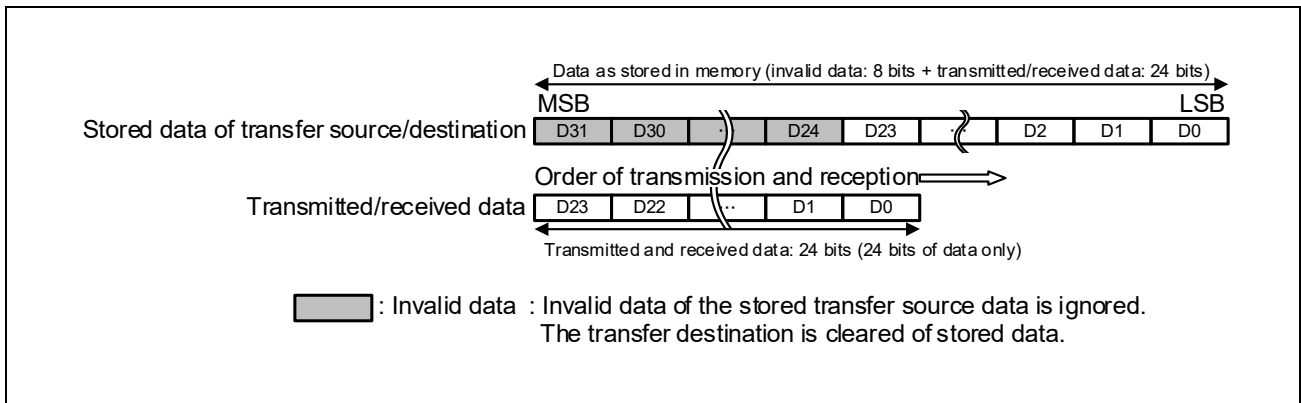


Figure 26.12 Data Length for Transfer: 24 Bits, No Parity Bit, MSB First, Transmission/Reception

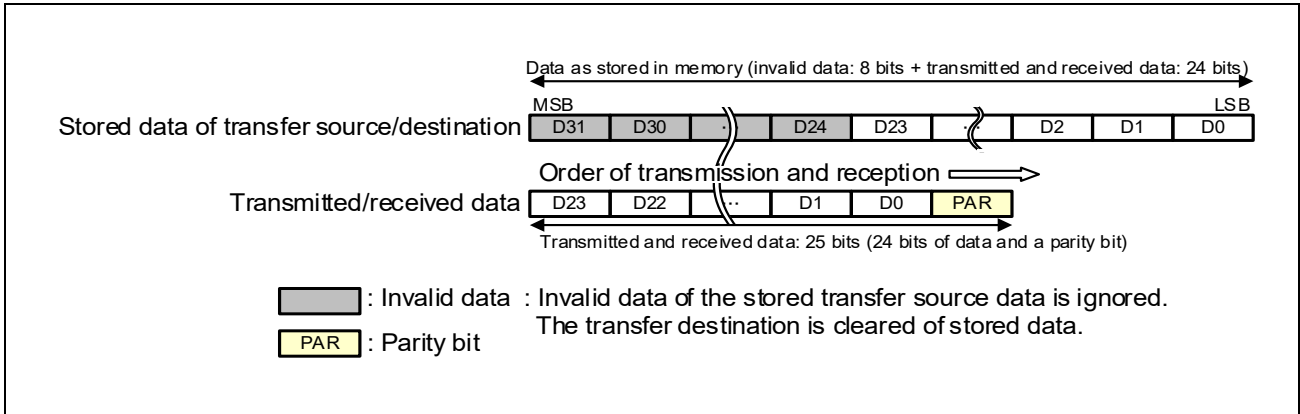


Figure 26.13 Data Length for Transfer: 24 Bits, Parity Bit, MSB First, Transmission/Reception

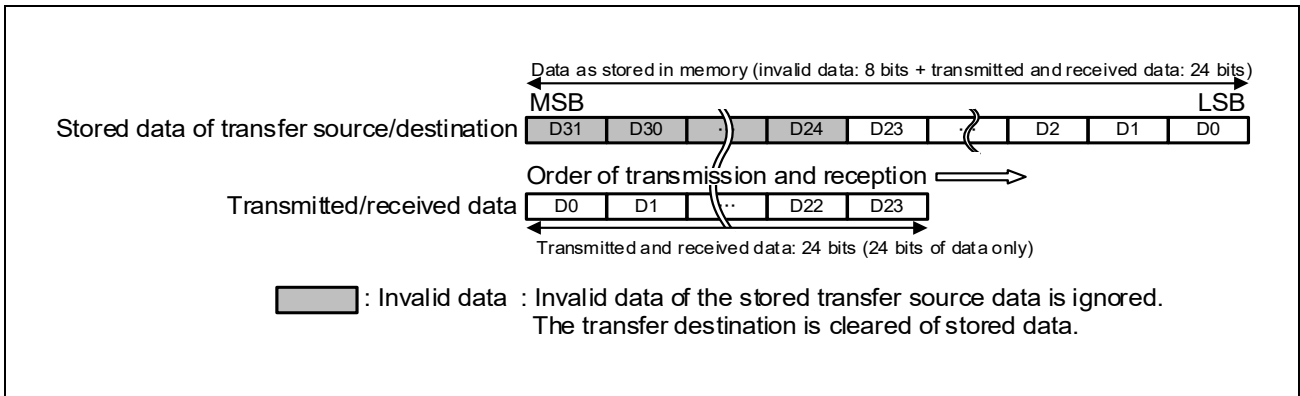


Figure 26.14 Data Length for Transfer: 24 Bits, No Parity Bit, LSB First, Transmission/Reception

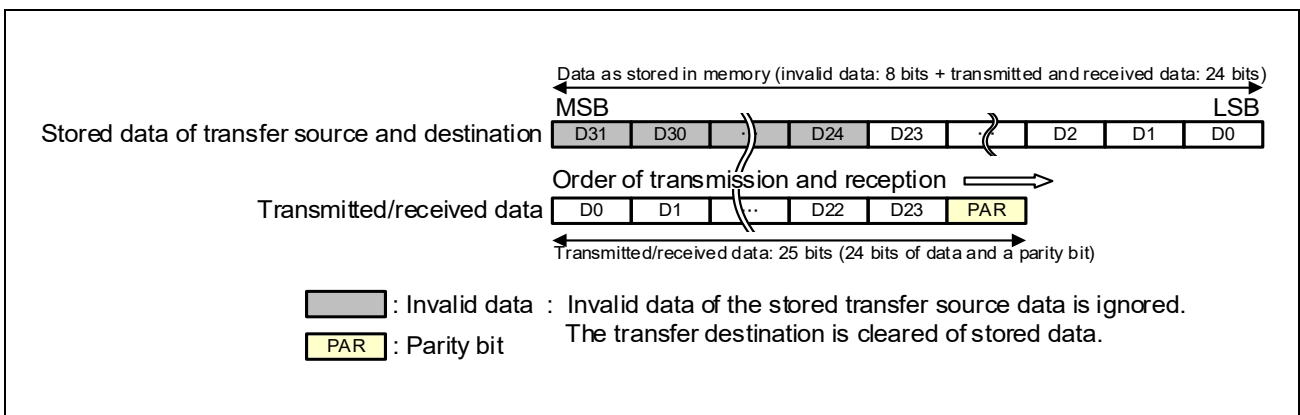


Figure 26.15 Data Length for Transfer: 24 Bits, Parity Bit, LSB First, Transmission/Reception

26.4.3.2 Relationship between the Serial Clock and Serial Data

The relationship between the clock and data depending on the values of the HSPInCTL.HSPInCPOL bit and HSPInCPHA bit is shown below.

- Figure 26.16 to Figure 26.19 are examples of MSB first.

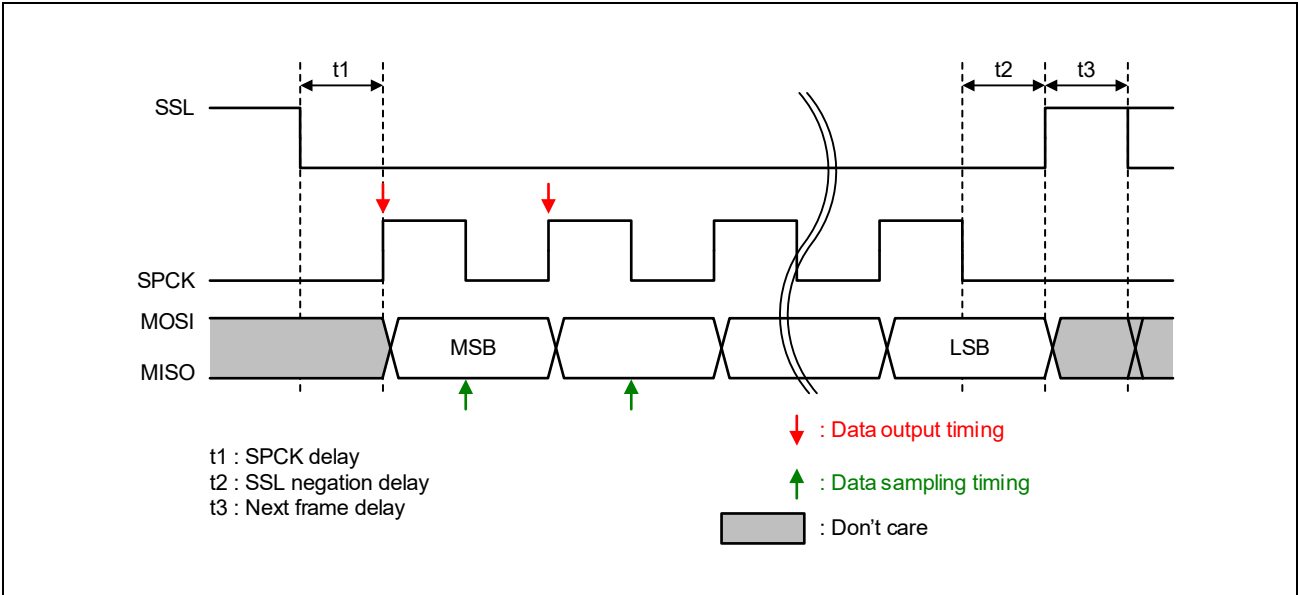


Figure 26.16 Relationship between the SPCK Pin and Data (Mode 1 (Default): HSPInCPOL = 0, HSPInCPHA = 1)

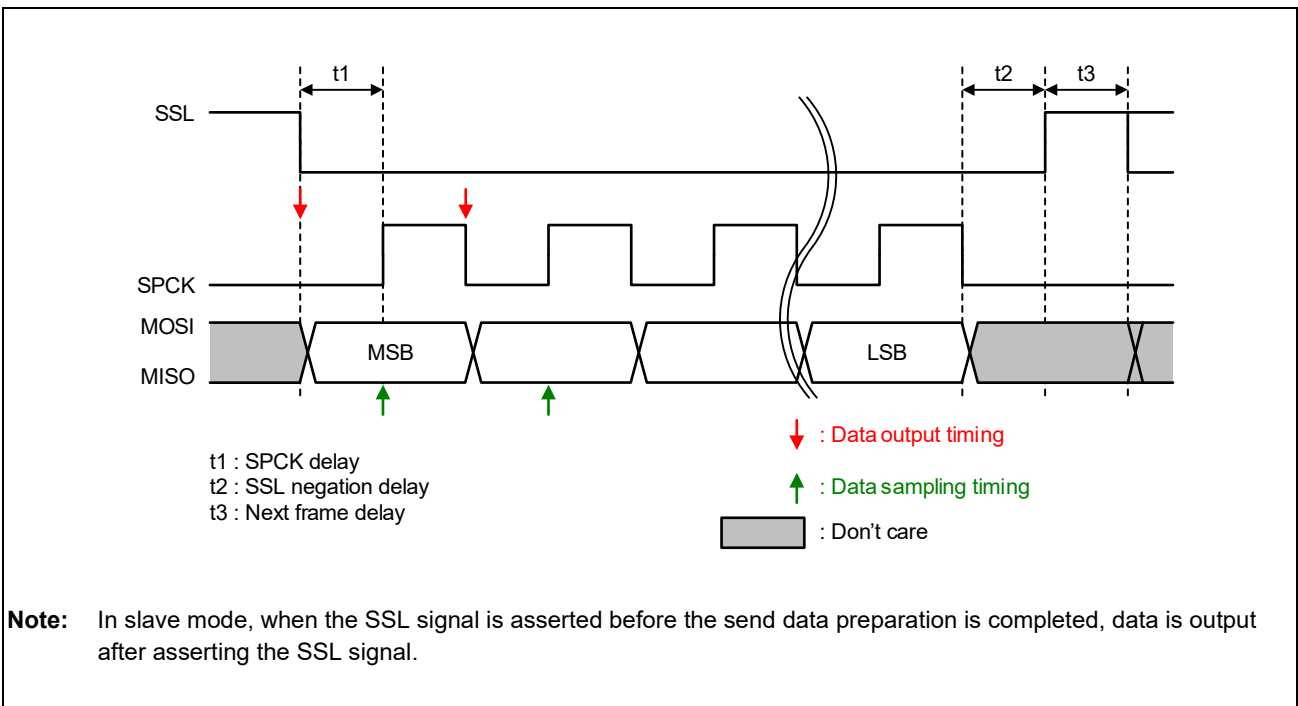


Figure 26.17 Relationship between the SPCK Pin and Data (Mode 0: HSPInCPOL = 0, HSPInCPHA = 0)

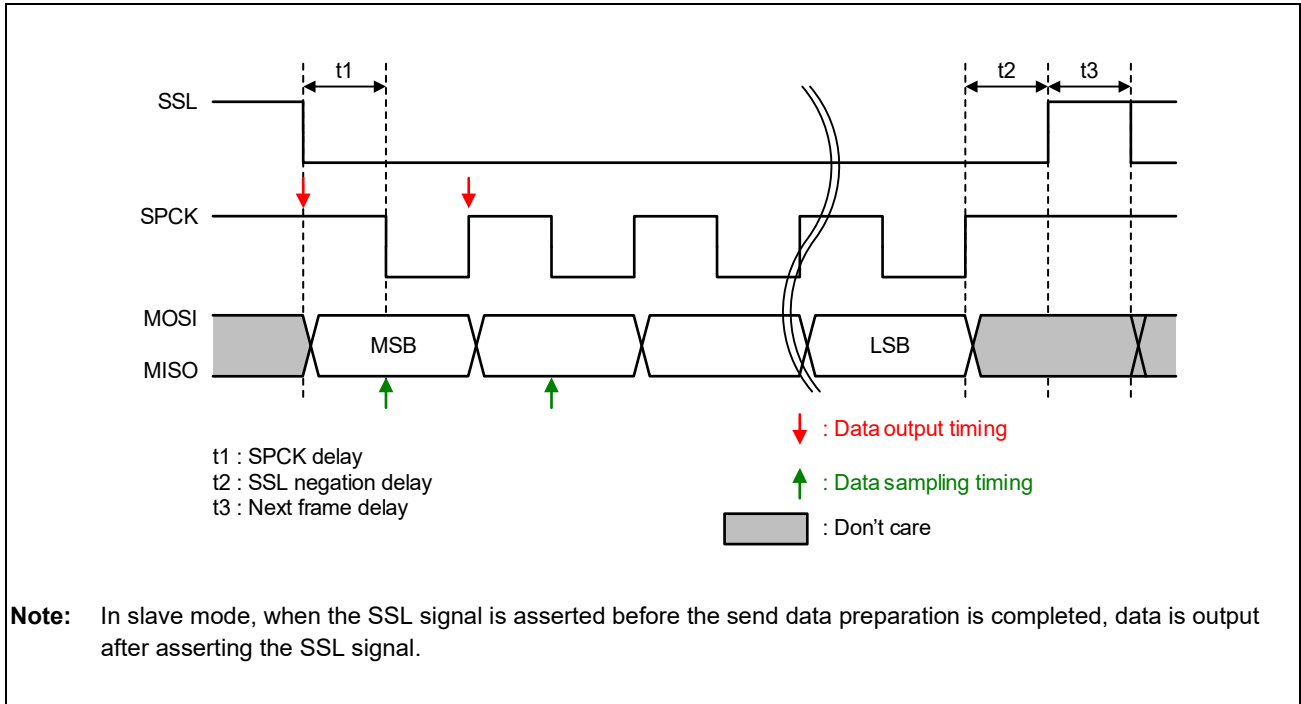


Figure 26.18 Relationship between SPCK Pin and Data (Mode 2: HSPInCPOL = 1, HSPInCPHA = 0)

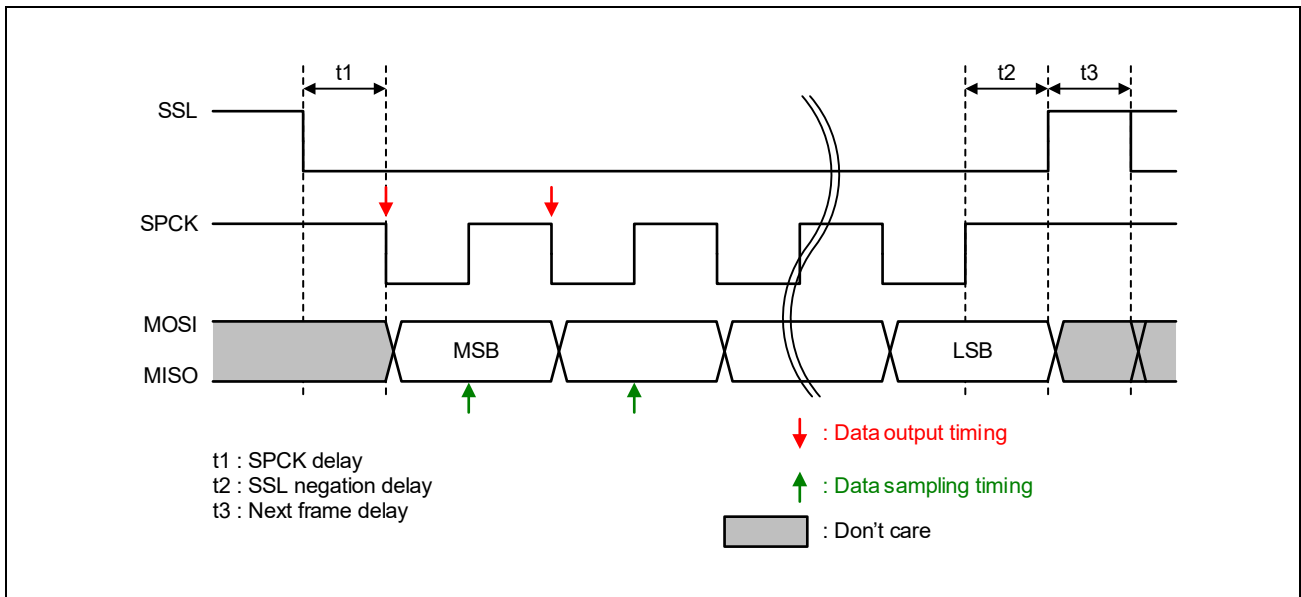


Figure 26.19 Relationship between SPCK Pin and Data (Mode 3: HSPInCPOL = 1, HSPInCPHA = 1)

26.4.3.3 MSB First and LSB First

Either MSB or LSB first for the serial data array can be selected by setting the LSB first selection bit (HSPInCTL.HSPInLSBEN). Selection of MSB or LSB first does not affect the position of the parity bit.

Table 26.38 shows the arrangement of data in memory for both MSB and LSB first operation.

Table 26.38 Data Storage Format

	63	56	55	48	47	40	39	32	31	24	23	16	15	8	7	0
	address+7		address+6		address+5		address+4		address+3		address+2		address+1		address	
0000 0000 _H	Data byte 7	Data byte 6	Data byte 5	Data byte 4	Data byte 3	Data byte 2	Data byte 1	Data byte 0								
0000 0008 _H	Data byte 15	Data byte 14	Data byte 13	Data byte 12	Data byte 11	Data byte 10	Data byte 9	Data byte 8								
0000 0010 _H	Data byte 23	Data byte 22	Data byte 21	Data byte 20	Data byte 19	Data byte 18	Data byte 17	Data byte 16								
0000 0018 _H	Data byte 31	Data byte 30	Data byte 29	Data byte 28	Data byte 27	Data byte 26	Data byte 25	Data byte 24								

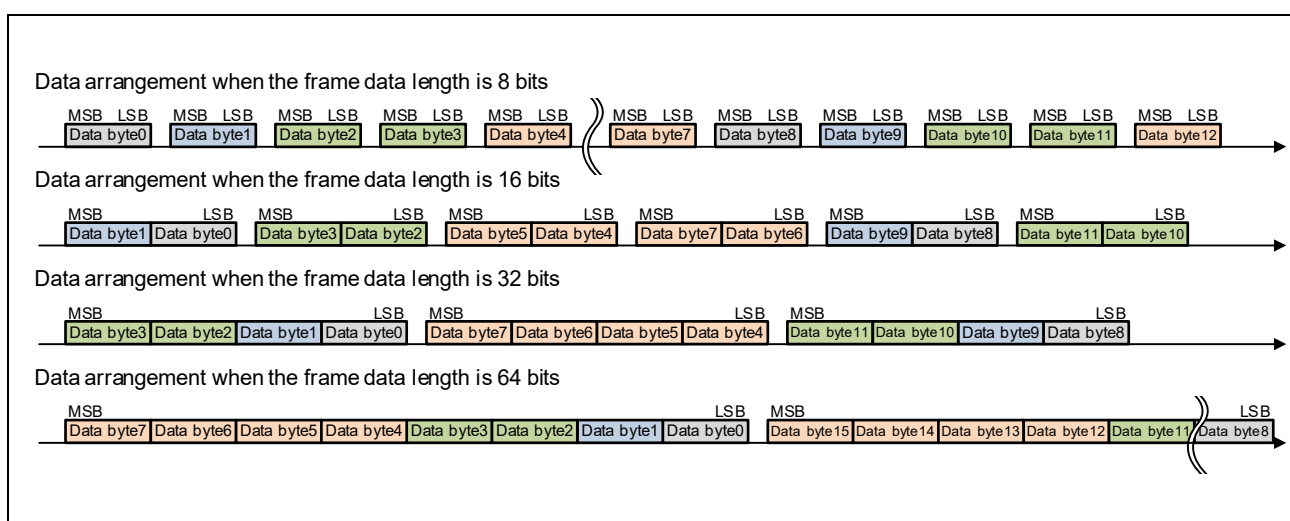


Figure 26.20 Order of Byte Frames in MSB-First Serial Output

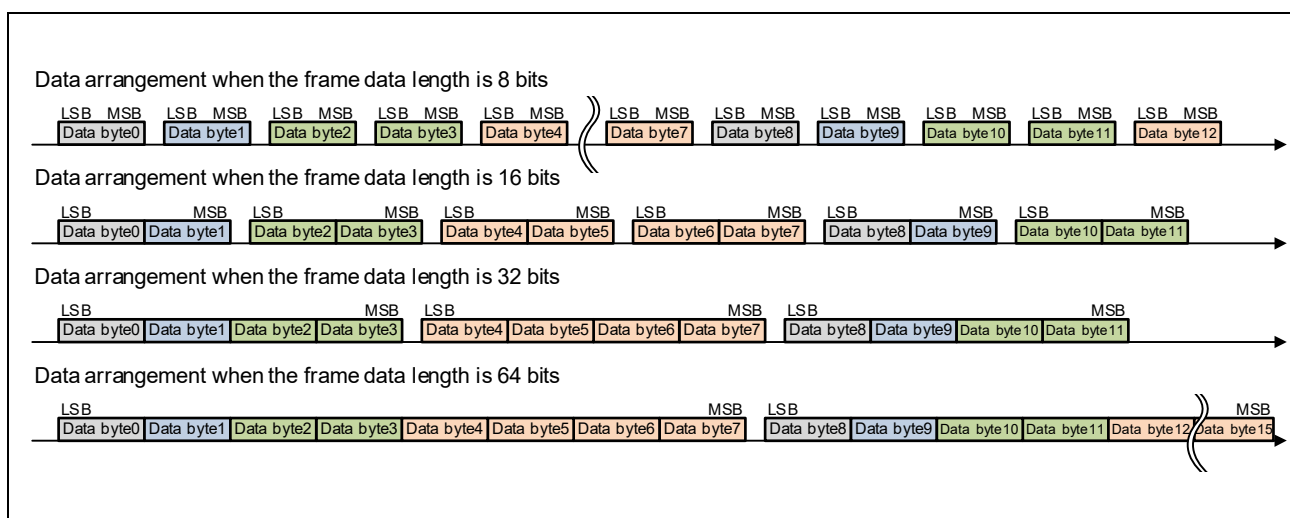


Figure 26.21 Order of Byte Frames in LSB-First Serial Output

26.4.4 Parity Bit

Parity bits are used to detect bit errors in transferred data. The same type of parity bit is used for transmission and reception.

One-bit errors can be detected through either even or odd parity.

For parity bit position within data frames, see **Section 26.4.3.1, Frame Data Format**.

When communication is suspended while receiving data (the HSPInEN.HSPInEN bit is 0), the parity cannot be calculated correctly. In this case, parity is not calculated nor is an interrupt request output.

26.4.4.1 Even Parity

(1) When data is transmitted

The value of the parity bit is controlled so that the number of bits with the value 1 in the transmitted data, including the parity bit itself, must be an even number.

The value of the parity bit is as follows.

An odd number of bits in the data for transmission have the value 1:1

An even number of bits in the data for transmission have the value 1:0

(2) When data is received

A parity error occurs when the result of counting the number of bits with the value 1 in the received data, including the parity bit, is an odd number.

26.4.4.2 Odd Parity

(1) When data is transmitted

The value of the parity bit is controlled so that the number of bits with the value 1 in the transmitted data, including the parity bit itself, must be an odd number.

The value of the parity bit is as follows.

An odd number of bits in the data for transmission have the value 1:0

An even number of bits in the data for transmission have the value 1:1

(2) When data is received

A parity error occurs when the result of counting the number of bits with the value 1 in the received data, including the parity bit, is an even number.

26.4.4.3 Parity Omission

A parity bit is not added to the transmitted data.

Reception of data proceeds assuming that there is no parity bit during reception. Since there is no parity bit, parity errors do not occur.

26.4.5 Buffer Structure

26.4.5.1 Structure of the Transmission Buffer

The transmission buffers form a double-buffer structure. Each transmission buffer can store 64 bytes of transmission data.

The buffer requests a transfer of data to the internal DMAC when either buffer0 or buffer1 becomes empty. During the request, the buffer can continue transmission by using the buffer that is not empty.

Requested data transfer amounts of the internal DMAC in its transfer of the first and last portions of the data vary depending on the settings of the Transmission Data Transfer Source Address Setting Register (HSPInTSAR), Communications Frame Count Setting Register (HSPInCFSET) and frame length selection bits (HSPInMD.HSPInFLEN[6:0]).

The internal DMAC is requested to transfer 64-byte increments of data from the buffer except for the first and last rounds of DMA transfer.

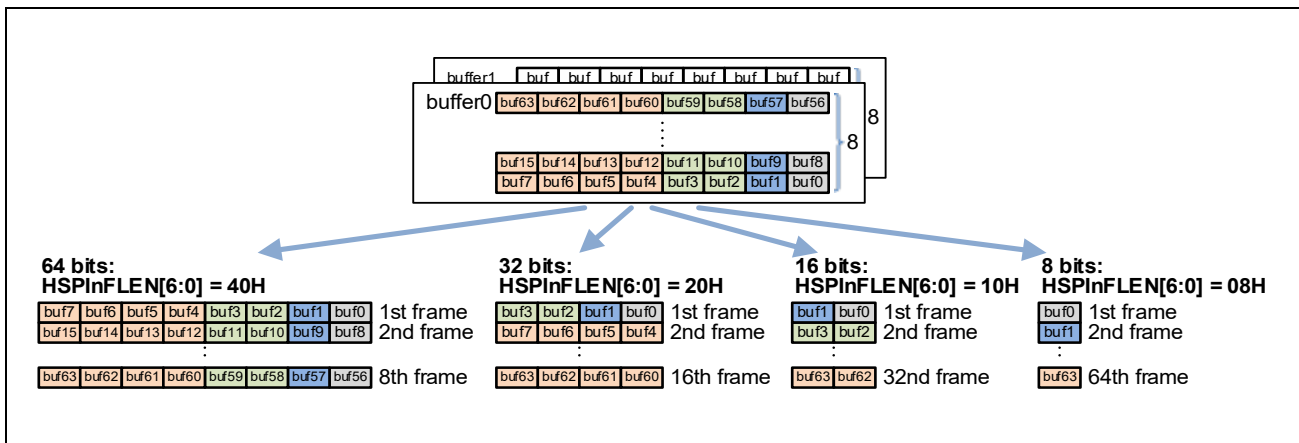


Figure 26.22 Structure of the Transmission Buffer (When Frame Data Length is 64, 32, 16, or 8 bits)

26.4.5.2 Reception Buffer Structure

The reception buffers form a double-buffer structure. Each reception buffer can store 64 bytes of transmission data. The buffer requests a transfer of data to the internal DMAC when either buffer0 or buffer1 becomes full. During the request, the buffer can continue receiving by using the buffer that is not full.

The Reception Data Transfer Destination Address Setting Register (HSPInRDAR), the Communication Frame Count Setting Register (HSPInCFSET) or frame length selection bit (HSPInMD.HSPInFLEN[6:0]) determines the amount of transmit data to be requested to the internal DMAC during DMA transfer of the first and last data.

The internal DMAC is requested to transfer 64-byte of data from the buffer except for the first and last data.

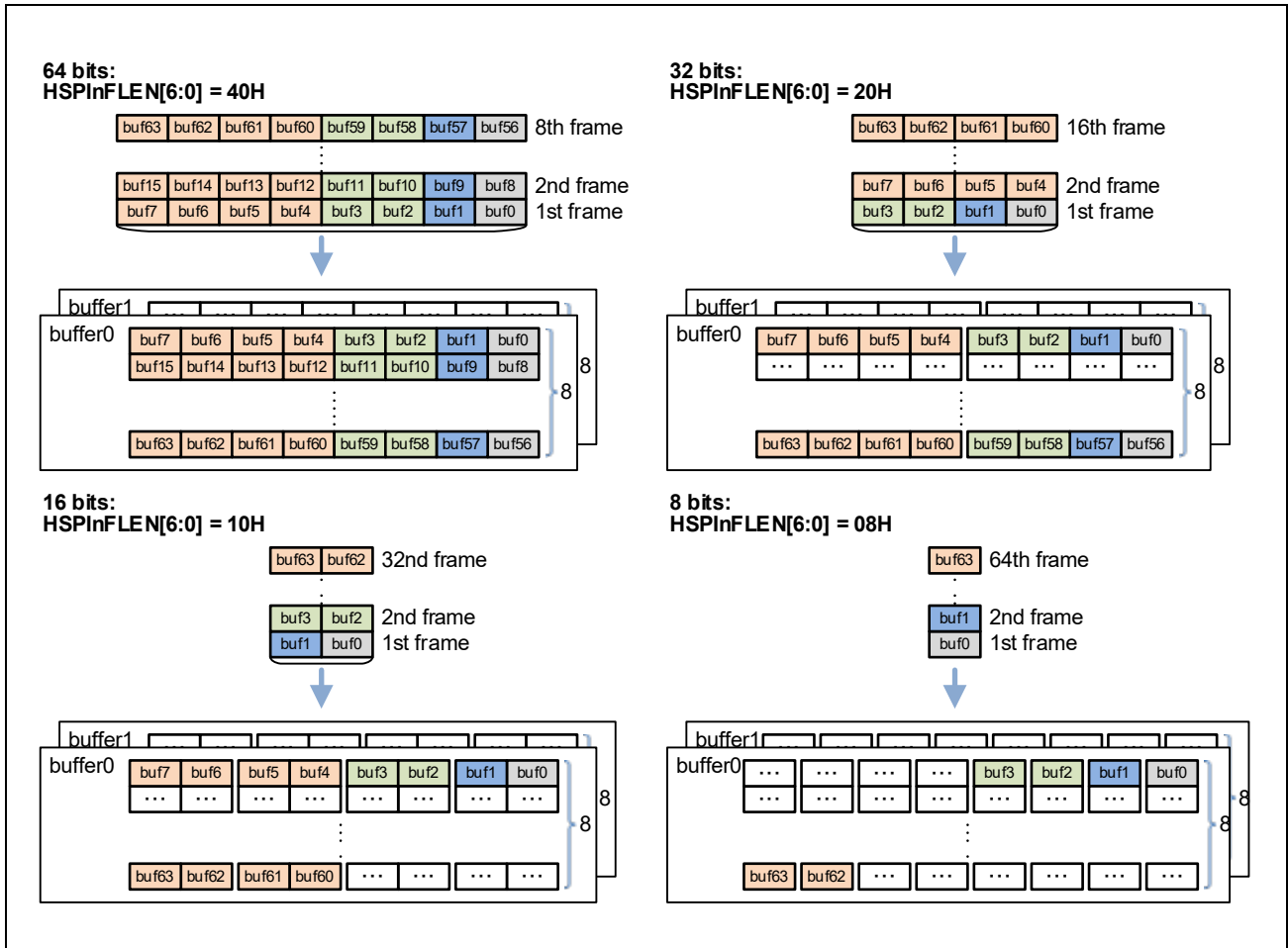


Figure 26.23 Reception Buffer Structure (Frame Lengths of 64, 32, 16, or 8 bits)

26.4.6 Interrupt Requests

26.4.6.1 Interrupt Sources

The HS-SPI issues four interrupt requests grouped into two types: INT_HSPI_BERR and INT_HSPI_CSTART, each of which is output in response to a single interrupt source, and INT_HSPI_ERR and INT_HSPI_CEND, each of which is output in response to several interrupt sources. The individual interrupt sources for INT_HSPI_ERR are overflow errors in reception, underrun errors in slave mode transmission, and parity errors in reception. The individual interrupt sources for INT_HSPI_CEND are completion of transfer and, in slave mode reception, completion of reception for the virtual port. Each interrupt request has an output enabling bit, a flag bit, and a flag clearing bit.

Writing to a flag clearing bit clears the flag. Writing 1 to the flag clearing bit corresponding to a generated interrupt clears that flag bit (writing 0 to the flag bit does not clear it.) .

An interrupt request is output as a high-level pulse with the width of 1 cycle of PCLK.

The source conditions, flag bits, and control bits for each of the interrupts are listed in **Table 26.39** and interrupt requests are output with the timing shown in **Figure 26.24**.

Table 26.39 HS-SPI Interrupt Requests and Interrupt Sources

Interrupt Request	Interrupt Source	Output Enabling Bit	Flag Bit	Flag Clearing Bit
INT_HSPI_BERR	H-Bus error	HSPInREN.BEE	HSPInIRST.BEF	HSPInIRCL.BEC
INT_HSPI_ERR	Reception overflow error	HSPInREN.ROVFEE	HSPInIRST.ROVFEF	HSPInIRCL.ROVFEC
	Transmission underrun error in slave mode	HSPInREN.TUDREE	HSPInIRST.TUDREF	HSPInIRCL.TUDREC
	Reception parity error	HSPInREN.PEE	HSPInIRST.PEF	HSPInIRCL.PEC
INT_HSPI_CSTART	Communication start in slave mode	HSPInREN.CSTAE	HSPInIRST.CSTAF	HSPInIRCL.CSTAC
INT_HSPI_CEND	Communication completion	HSPInREN.CENDE	HSPInIRST.CENDF	HSPInIRCL.CENDC
	Virtual port reception completion	HSPInREN.VPE	HSPInIRST.VPF	HSPInIRCL.VPC

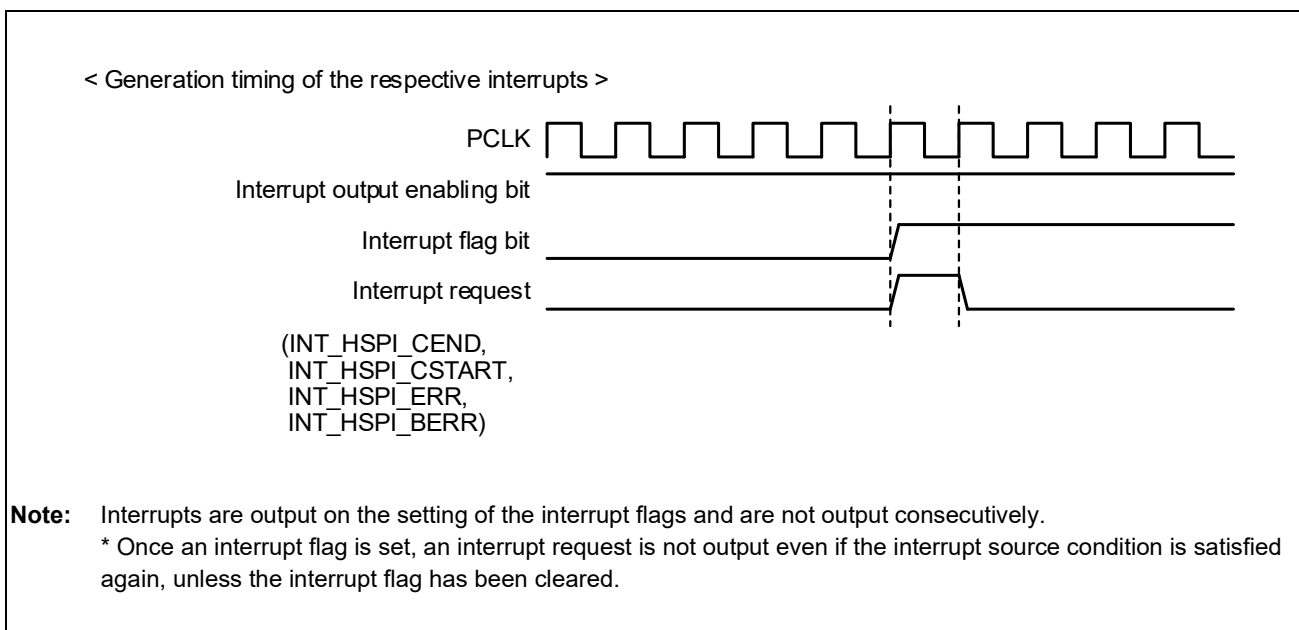


Figure 26.24 Interrupt Request Output Timing Diagram

26.4.6.2 H-Bus Error Interrupt Requests

When receiving an error response during DMA transfer, the DMAC outputs an H-Bus error interrupt request (INT_HSPI_BERR). When an H-Bus error interrupt is generated and while the H-Bus error flag (HSPInIRST.HSPInBEF) is 1, communication complete interrupt requests (INT_HSPI_CEND) will not be output. In addition, H-Bus error interrupt request (INT_HSPI_BERR) is not output as long as the H-Bus error flag (HSPInIRST.HSPInBEF) is 1.

When a H-Bus error occurs, transmission and reception continues. Reading of data from or writing of data to an external memory does not proceed correctly, so operation is described as follows.

Transmission: When the number of frames specified in the Communication Frame Count Setting Register (HSPInCFSET) have been transferred, transmission of data is stopped. At this time, a communication complete interrupt request (INT_HSPI_CEND) is not output.

Subsequent frame data after occurrence of an H-Bus error is undefined.

Reception: When the number of frames specified in the Communication Frame Count Setting Register (HSPInCFSET) have been transferred, reception of data is stopped. At this time, a communication complete interrupt request (INT_HSPI_CEND) is not output.

After an H-Bus error has occurred, received data frames are not stored in the external memory, so there will be a shortfall of valid data in external memory.

To start communication using the communication enabling bit (HSPInEN.HSPInEN) or the communication start trigger bit (HSPInEN.HSPInSTATRG), start by clearing the H-Bus error flag to 0, and then enable communication (by setting the HSPInEN.HSPInEN bit to 1). When an H-Bus error interrupt is generated, the service routine should include processing to handle the error, suspension and initialization.

26.4.6.3 Error Interrupt Requests

The error interrupt request (INT_HSPI_ERR) is output in three cases:

- a) When the reception buffer overflows during master transmission/reception, slave transmission/reception, or slave reception,
- b) When the transmission buffer underruns during slave transmission/reception or slave transmission, or
- c) When a parity error occurs during master transmission/reception, slave transmission/reception, or slave reception.

Interrupt requests (INT_HSPI_ERR) are not output in response to further errors as long as the error interrupt flag for any of these three errors is 1.

A communication complete interrupt request (INT_HSPI_CEND) is not output as long as one of the error interrupt flags (HSPInIRST.HSPInROVFEF and HSPInIRST.HSPInTUDREF) is 1.

While the reception parity error flag (HSPInIRST.HSPInPEF) is 1, a communication complete interrupt request (INT_HSPI_CEND) is output.

To start communication using the communication enabling bit (HSPInEN.HSPInEN) or the communication start trigger bit (HSPInEN.HSPInSTATRG), start by clearing the reception overflow error flag (HSPInIRST.HSPInROVFEF), transmission underrun error flag (HSPInIRST.HSPInTUDREF) and the reception parity error flag (HSPInIRST.HSPInPEF) to 0. When an error interrupt is generated, the service routine should include processing to handle the error, suspension and initialization.

(1) Reception Overflows

Since the reception buffer has a double-buffer structure, one buffer can continue to receive data while performing DMA transfer of data.

Further data for storage in the reception buffers arriving while both of the buffers are full leads to a reception overflow error and setting of the reception overflow error flag (HSPInIRST.HSPInROVFEF) to 1. However, the reception of data continues even after a reception overflow has occurred. Data in the reception buffer that was currently receiving data are overwritten in the order received, from oldest to most recent.

(2) Underruns during Slave Transmission

When the communication clock and SSL become active with the transmission buffer empty, a transmission underrun error occurs and the transmission underrun error flag (HSPInIRST.TUDREF) is set to 1. Transmission continues even after a transmission underrun. However, valid frame data is not transmitted.

A transmission underrun is more likely if the transmission data transfer source address is close to the second half of a range between the 64-byte boundaries, since this may make it impossible to prepare data for transmission in time.

(3) Reception Parity Errors

When the parity value calculated based on the reception frame data differs from the value (HSPInCTL.HSPInPRSL[1] = 0: even parity, HSPInCTL.HSPInPRSL[1] = 1: odd parity) specified in the parity setting bit (HSPInCTL.HSPInPRSL[1:0]), a reception parity error occurs and the reception parity error flag (HSPInIRST.HSPInPEF) is set to 1. The reception parity error flag (HSPInIRST.HSPInPEF) is set when an error is detected in the reception section (de-serializer).

Even after a reception parity error has occurred, reception continues.

26.4.6.4 Communication Start Interrupt Requests

When assertion of the signal on the SSL0 pin [SSL0 level is low when SSL0 polarity bit (HSPInSSCTL.HSPInSS0P) is 0, and SSL0 level is high when the bit is 1] is detected in slave operation, a communication start interrupt request (INT_HSPI_CSTART) is output.

Communication start interrupt requests (INT_HSPI_CSTART) are not output as long as the communication start interrupt flag (HSPInIRST.HSPInCSTAF) is 1. To start communication using the communication enabling bit (HSPInEN.HSPInEN) or the communication start trigger bit (HSPInEN.HSPInSTATRG), start by clearing the communication start interrupt flag (HSPInIRST.HSPInCSTAF) to 0, then enable communication.

26.4.6.5 Communication Complete Interrupts

A communication complete interrupt request (INT_HSPI_CEND) is output when as many frames as are specified in the Communication Frame Count Setting Register (HSPInCFSET) have been transferred, or when one frame has been received while using the virtual port in slave reception.

When the virtual port is not in use (HSPInVPEN.HSPInVPEN = 0), and if the H-Bus error flag (HSPInIRST.HSPInBEF), the reception overflow error flag (HSPInIRST.HSPInROVFEF), the transmission underrun error flag (HSPInIRST.HSPInTUDREF), or the communication complete flag (HSPInIRST.HSPInCENDF) is 1, communication complete interrupt requests (INT_HSPI_CEND) will not be output. When the virtual port is in use (HSPInVPEN.HSPInVPEN = 1) and the virtual port reception complete flag (HSPInIRST.HSPInVPF) is 1, communication complete interrupt requests (INT_HSPI_CEND) will not be output.

Communication complete interrupt requests (INT_HSPI_CEND) are output while the reception parity error flag (HSPInIRST.HSPInPEF) is 1.

To start communication using the communication enabling bit (HSPInEN.HSPInEN) or the communication start trigger bit (HSPInEN.HSPInSTATRG), start by clearing each interrupt status flag to 0, and then enable communication.

26.4.7 Reception and Transmission during Suspension of Transfer

When the communications enabling bit (HSPInEN.HSPInEN) is set to 0 while a frame is being received (before sampling of the last bit of the frame data length specified in the frame data length setting bit (HSPInMD.HSPInFLEN[6:0])), or when the SSL0 input signal is negated [SSL0 level is high when the SSL0 polarity bit (HSPInSSCTL.HSPInSSOP) is 0, and SSL0 level is low when the bit is 1], received data frames shorter than the specified length are judged to be invalid. In these cases, operation of the HS-SPI is as follows:

<Reception>

- When master and slave reception are suspended, the received data is discarded and not counted as being received.
- In slave reception, when the frame data judged to have been invalid is re-sent and that data is successfully received, the sent data is stored in the reception buffer and is counted as received frame data.
- When a frame judged to have been invalid is not re-sent, the number of frames for reception specified in the Communication Frame Count Setting Register (HSPInCFSET) is not attained, so a communication complete interrupt request (INT_HSPI_CEND) is not output.
If timeout is being managed outside the HS-SPI, it is judged to be a reception timeout.

<Transmission>

- When transfer is suspended during master transmission, after the frame already accepted by the transmission section has been transmitted, HS-SPI suspends transfer.
- When transfer is suspended during slave transmission, HS-SPI suspends communication without waiting for completion of transmission of a frame accepted for transmission. When transfer is suspended by negating the SSL0 input signal, if the number of frames specified in the Communication Frame Count Setting Register (HSPInCFSET) has been transferred, a communication complete interrupt request (INT_HSPI_CEND) is output. However, since transfer of frame data after suspension cannot be guaranteed, resend the frame data.

26.4.8 Operations in cases of Frame Overrun during Transmission

For frame overruns (transmitted frames exceeding the number specified in the Communication Frame Count Setting Register (HSPInCFSET)), if the communication clock and SSL signal are input from an external device during slave operation, the transmission section does not operate and data following the last frame is not transmitted. This does not result in a transmission underrun error.

26.4.9 Operation in the cases of Frame Overrun during Reception

After the number of transmitted frames reaches the value specified in the Communication Frame Count Setting Register (HSPInCFSET), any frames that exceed the specified number of frames (overrun frames) are discarded.

When an excess frame that has a reception parity error is received, an H-Bus error interrupt request (INT_HSPI_ERR) is output.

26.4.10 Timeouts

In slave mode, when the first assertion of the signal SSL0 pin [level is high when the SSL0 polarity bit (HSPInSSCTL.HSPInSS0P) is 0 and level is low when the bits is 1] is detected, a communication start interrupt request (INT_HSPI_CSTART) is output. This interrupt signal and an external timer can be used to monitor the completion of transfer within the expected period and therefore detect any communications timeout errors.

Figure 26.25 shows an example of an external timer detecting an incomplete transfer due to received data having been discarded during the transmission/reception of the second frame.

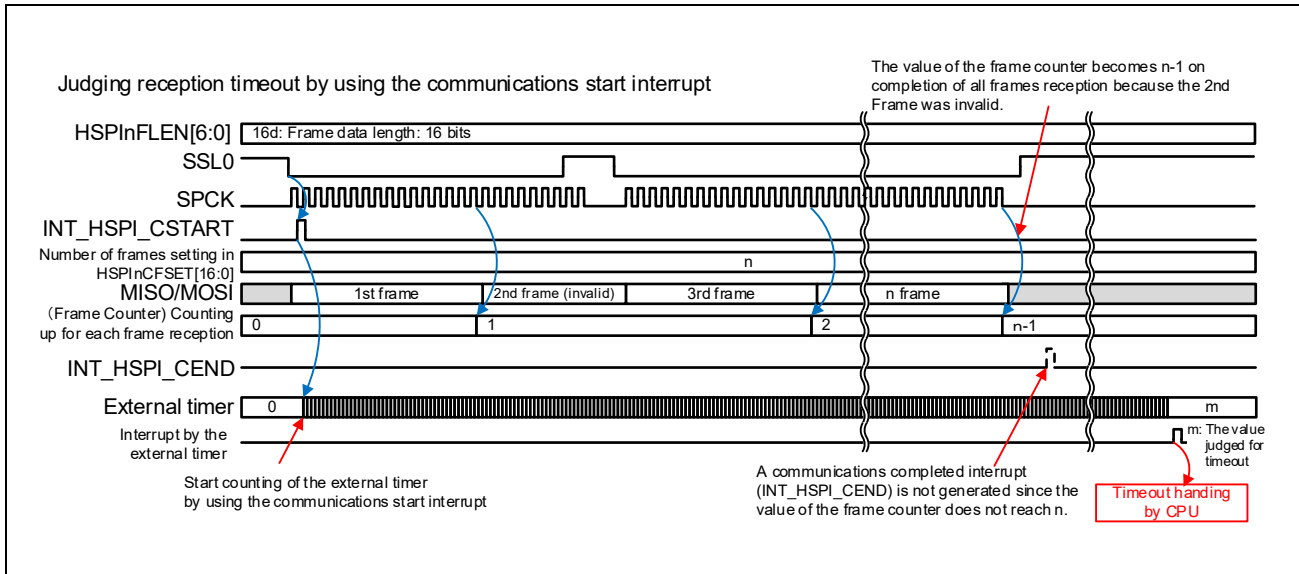


Figure 26.25 Example of Operation When a Timeout Error is Detected

26.4.11 Burst Operation

HS-SPI burst operation can be selected during continuous transfer in master operation by setting the burst operation selecting bit (HSPInCTL.HSPInBRST) to 1.

- (1) Normal operation (when the HSPInCTL.HSPInBRST bit is set to 0)

In normal operation, SSL0 is negated for every frame. (The figure below is when the SSL polarity setting bit (HSPInSSCTL.HSPInSS0P) is 0)

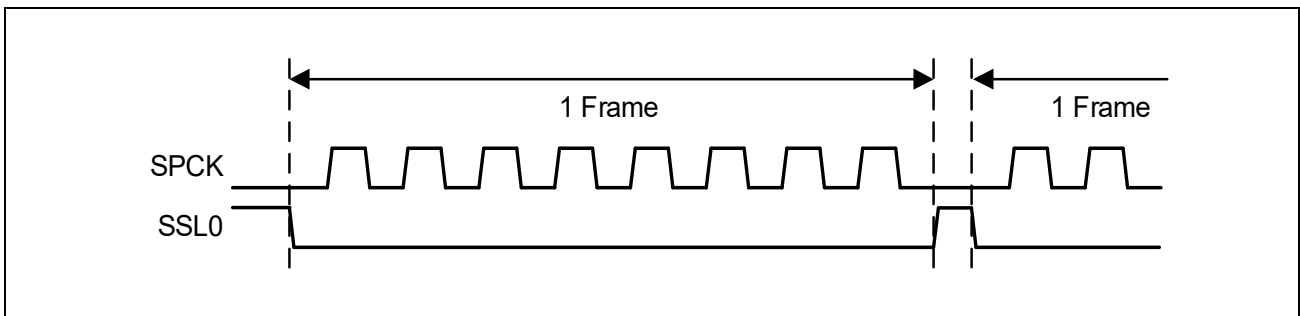


Figure 26.26 Communication Clock and SSL Waveforms during Normal Operation

- (2) Burst operation (when the HSPInCTL.HSPInBRST bit is set to 1)

In burst operation, SSL0 is not negated after the transfer of each frame. If the load on the H-Bus becomes such that transferred data cannot be received, the communication clock (SPCK) may be stopped in the midst of burst operation while this condition persists.

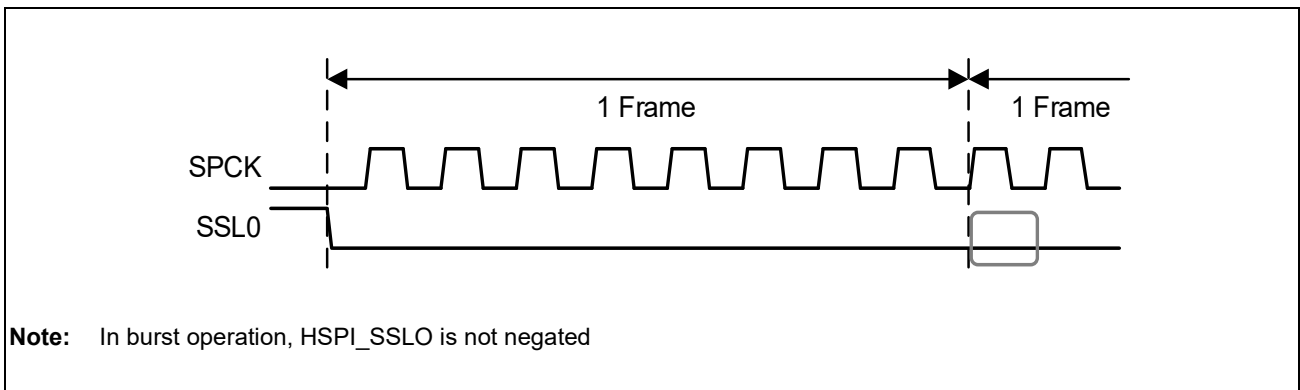


Figure 26.27 Communication Clock and SSL Waveforms during Burst Operation

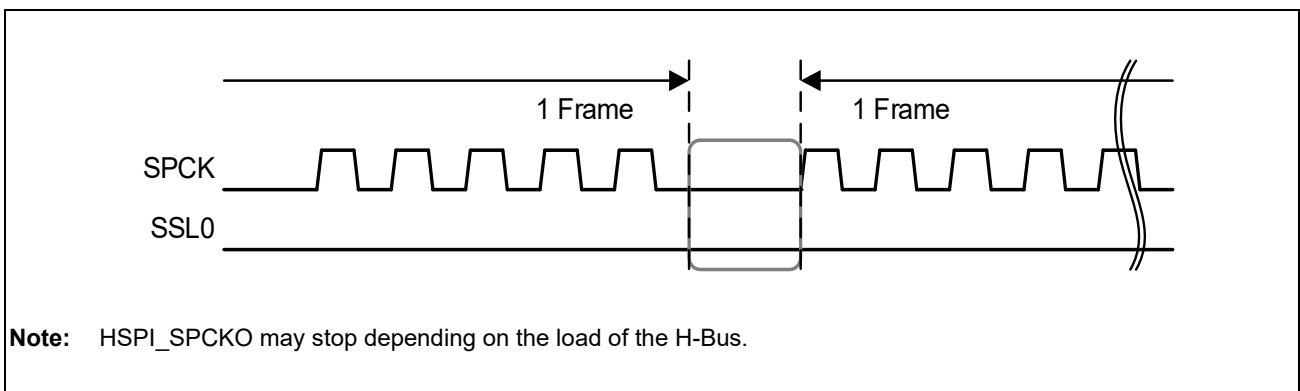


Figure 26.28 Waveforms When Communication Clock Stops during Burst Operation

26.4.12 Virtual Port Function

The received frame data can be output to the virtual port by setting the HSPInVPEN.HSPInVPEN bit. After converting MSB/LSB, received frame data is output to the virtual port in frame units. For operation using the virtual port, see **Section 26.4.1.6, Slave Reception (When the Virtual Port Output is Enabled)**.

Figure 26.29 shows an example of output waveforms on the HSSPIn_RDATA[63:0] and virtual port pins.

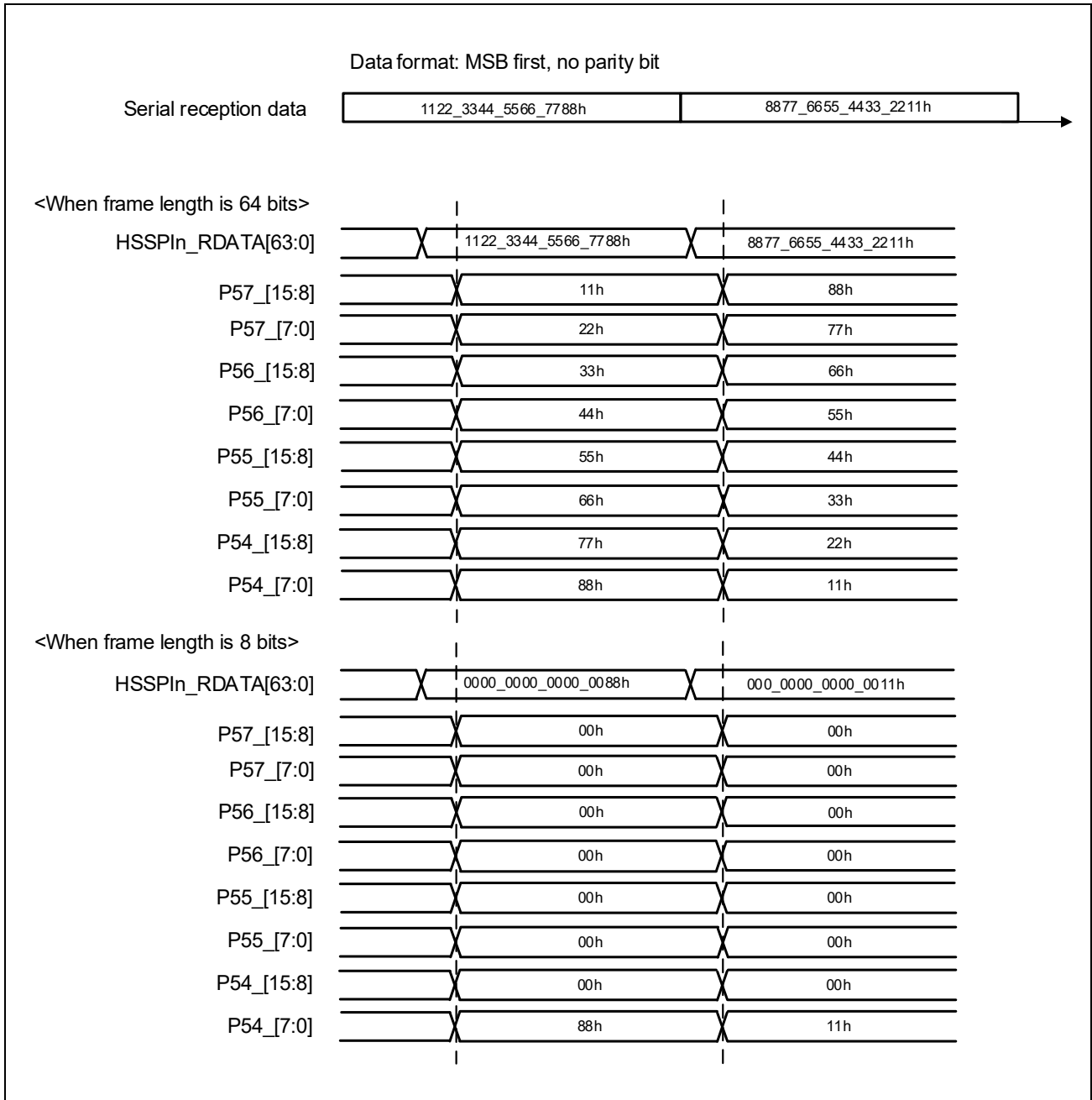


Figure 26.29 Waveforms of Received Data Output When Virtual Port Enabled

26.5 Operation

26.5.1 Power-on, Initialization and Stop Processing Flows

Figure 26.30 is a flowchart of power-on, **Figure 26.31** is a flowchart of initialization and **Figure 26.32** is a flowchart of stop processing after communication is prohibited.

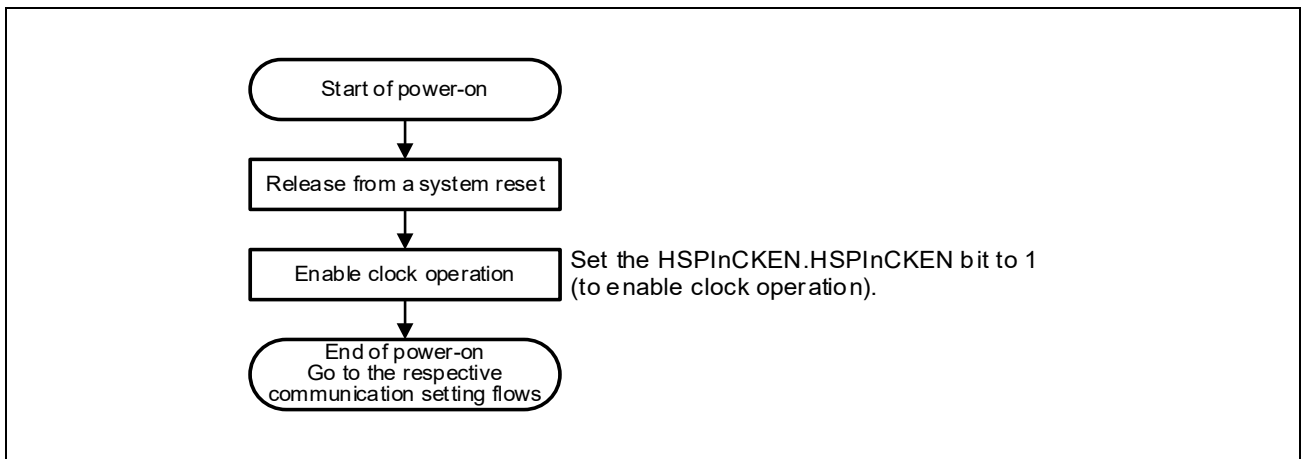


Figure 26.30 Power-on Flow

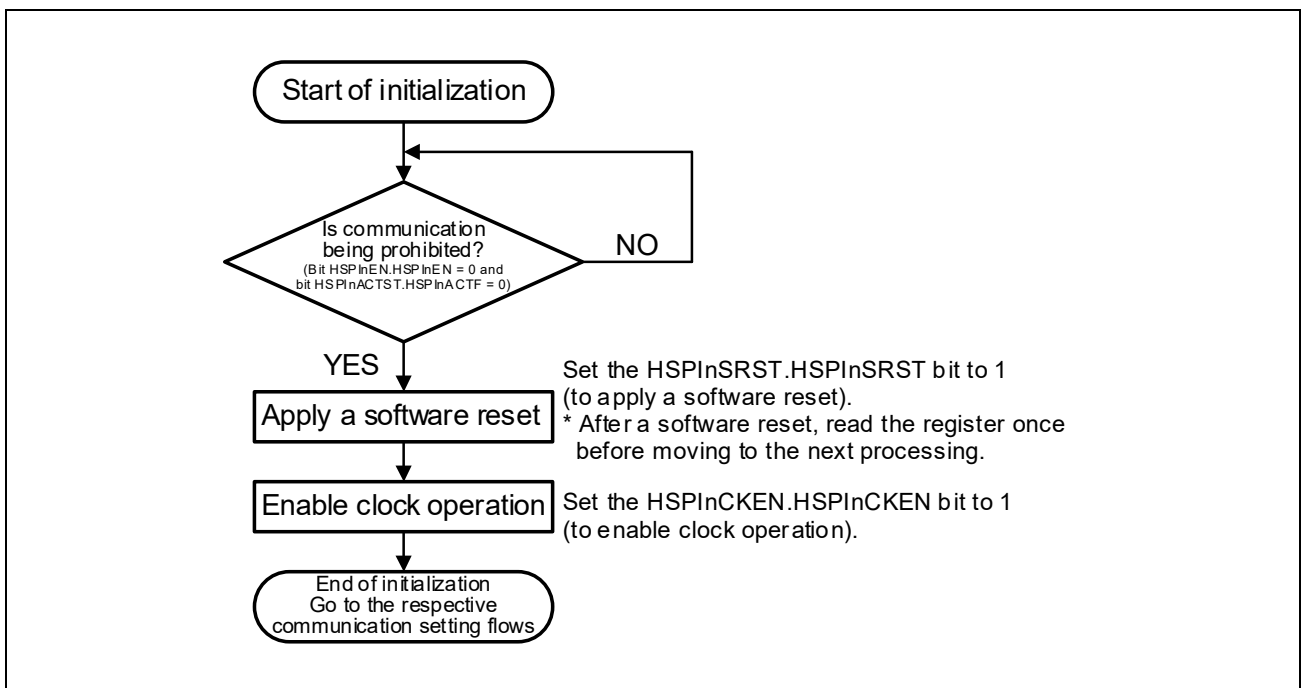


Figure 26.31 Initialization Flow

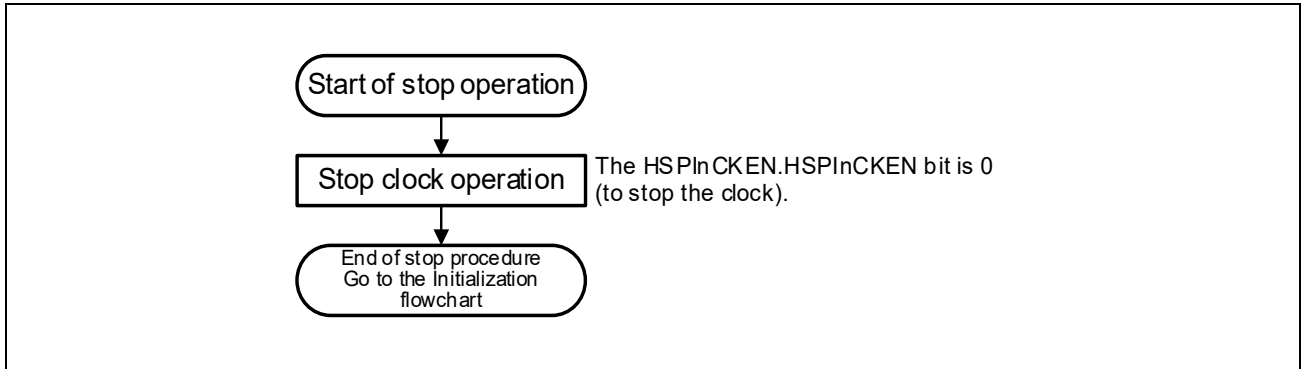


Figure 26.32 Stop Processing Flow

26.5.2 Master Mode

26.5.2.1 Transmission and Reception Setting Flows

Figure 26.33 shows an example of the process flow for transmission and reception settings in master mode.

When starting the transfer of HS-SPI, the pin settings must be completed in advance.

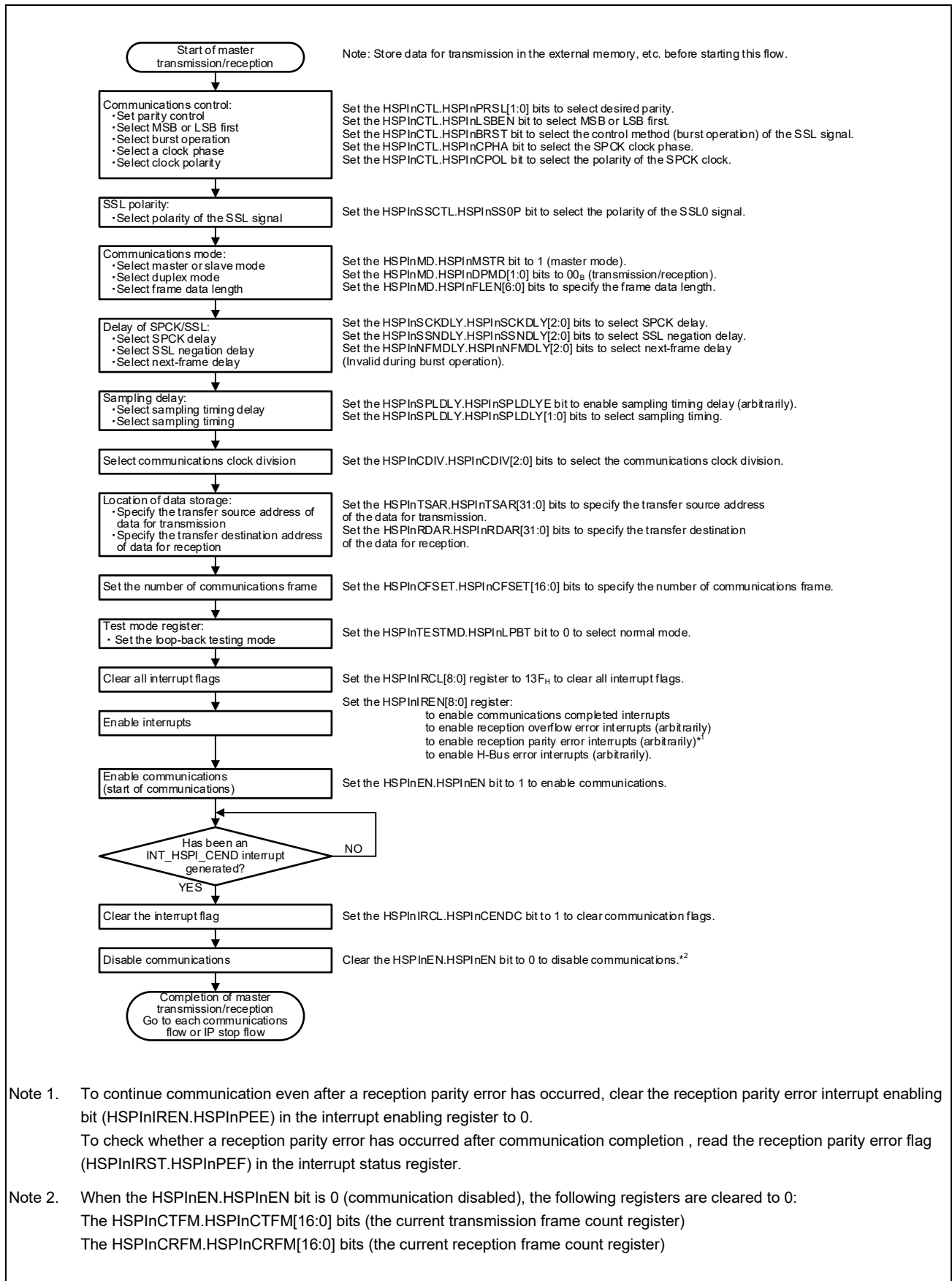


Figure 26.33 Master Mode Transmission and Reception Setting Flow Examples

26.5.2.2 Transmission Setting Flow

Figure 26.34 shows an example of the flow for transmission setting in master mode.

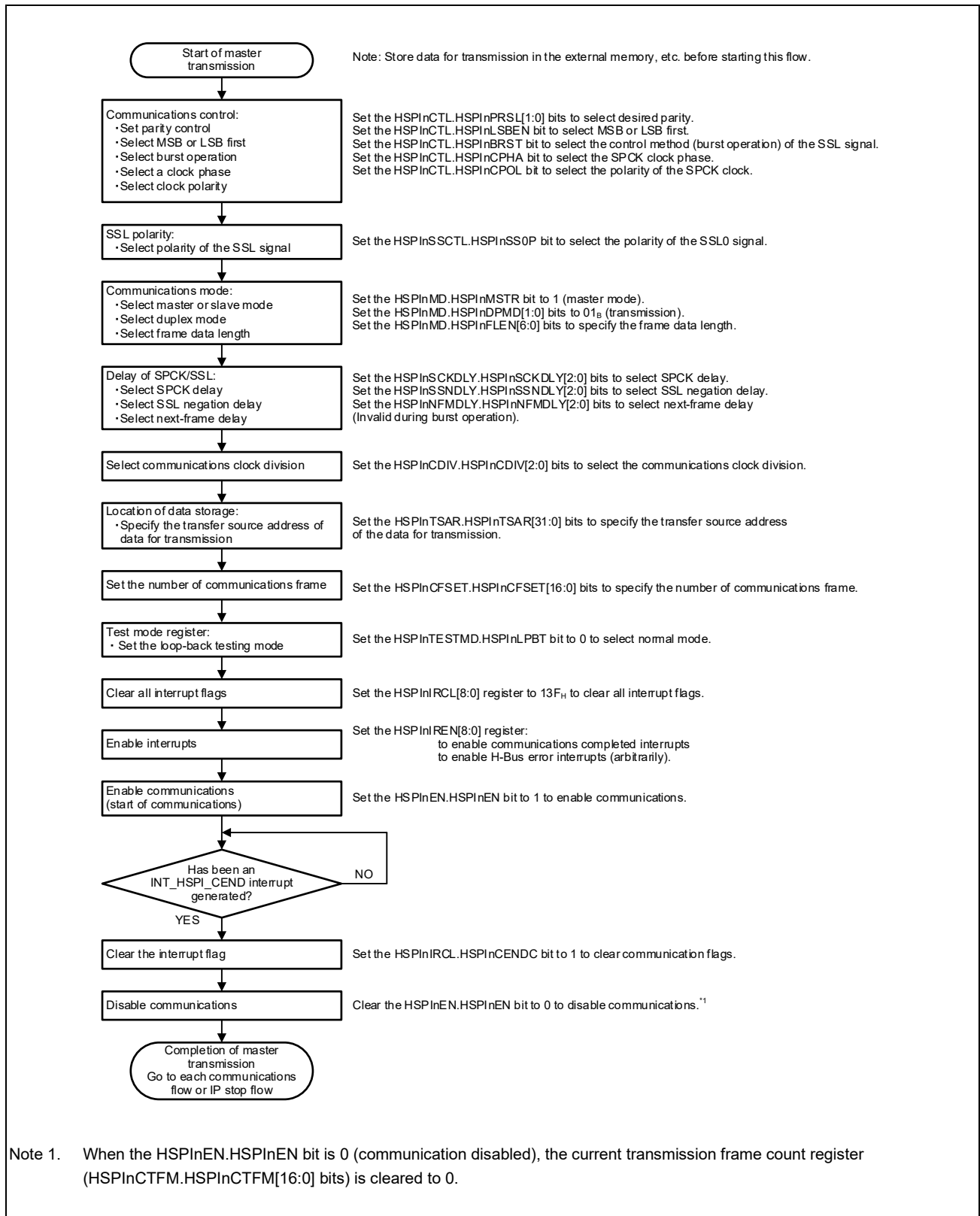


Figure 26.34 Transmission Setting Flow Example in Master Mode

26.5.2.3 Loop-back Test Setting Flow

Figure 26.35 shows an example of the flow for loop-back test setting in master mode.

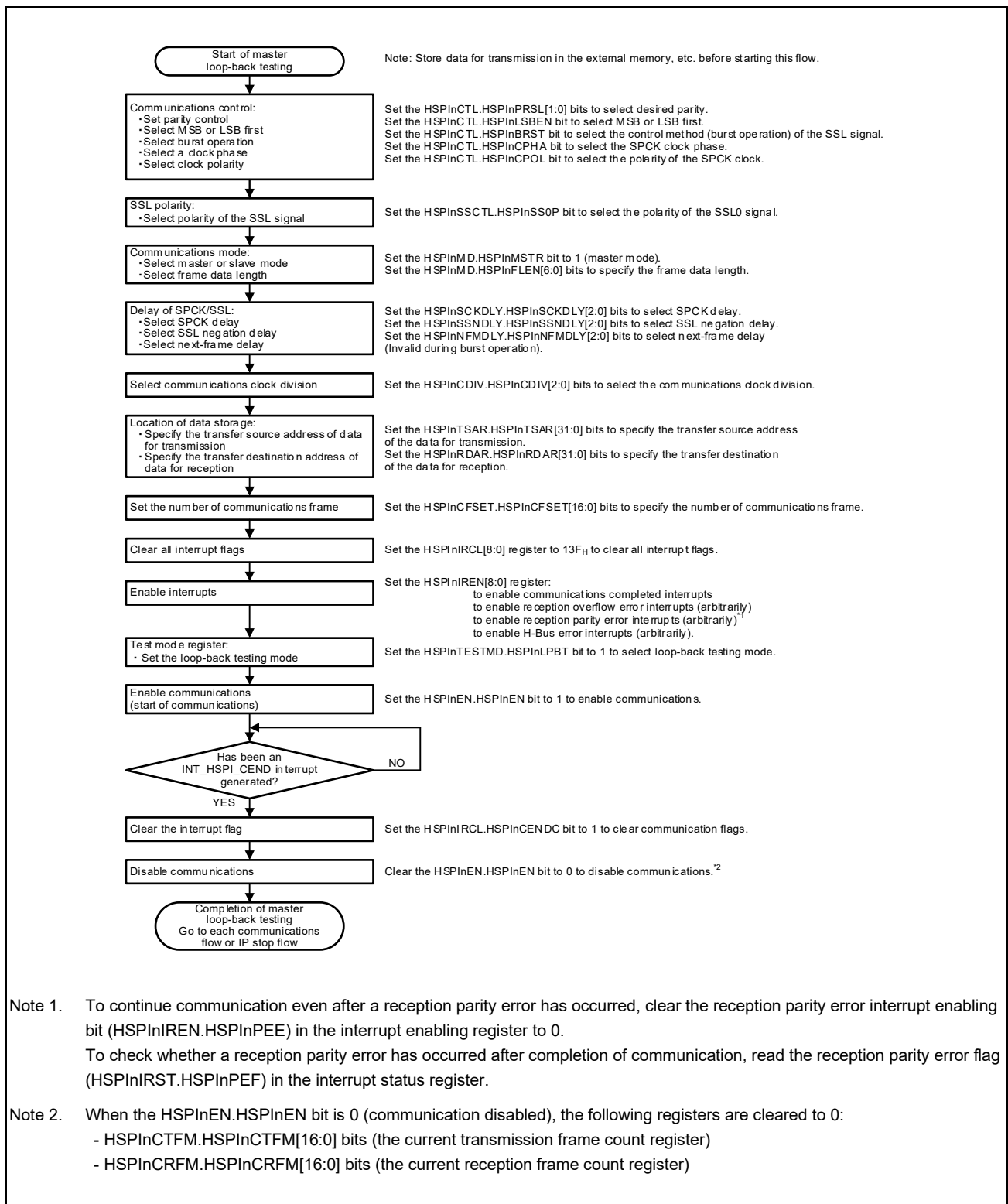


Figure 26.35 Master Mode Loop-back Test Setting Flow

26.5.3 Slave Mode

26.5.3.1 Transmission and Reception Setting Flows

Figure 26.36 shows an example of the process flow for transmission and reception settings in slave mode.

When starting the transfer of HS-SPI, the pin settings must be completed in advance.

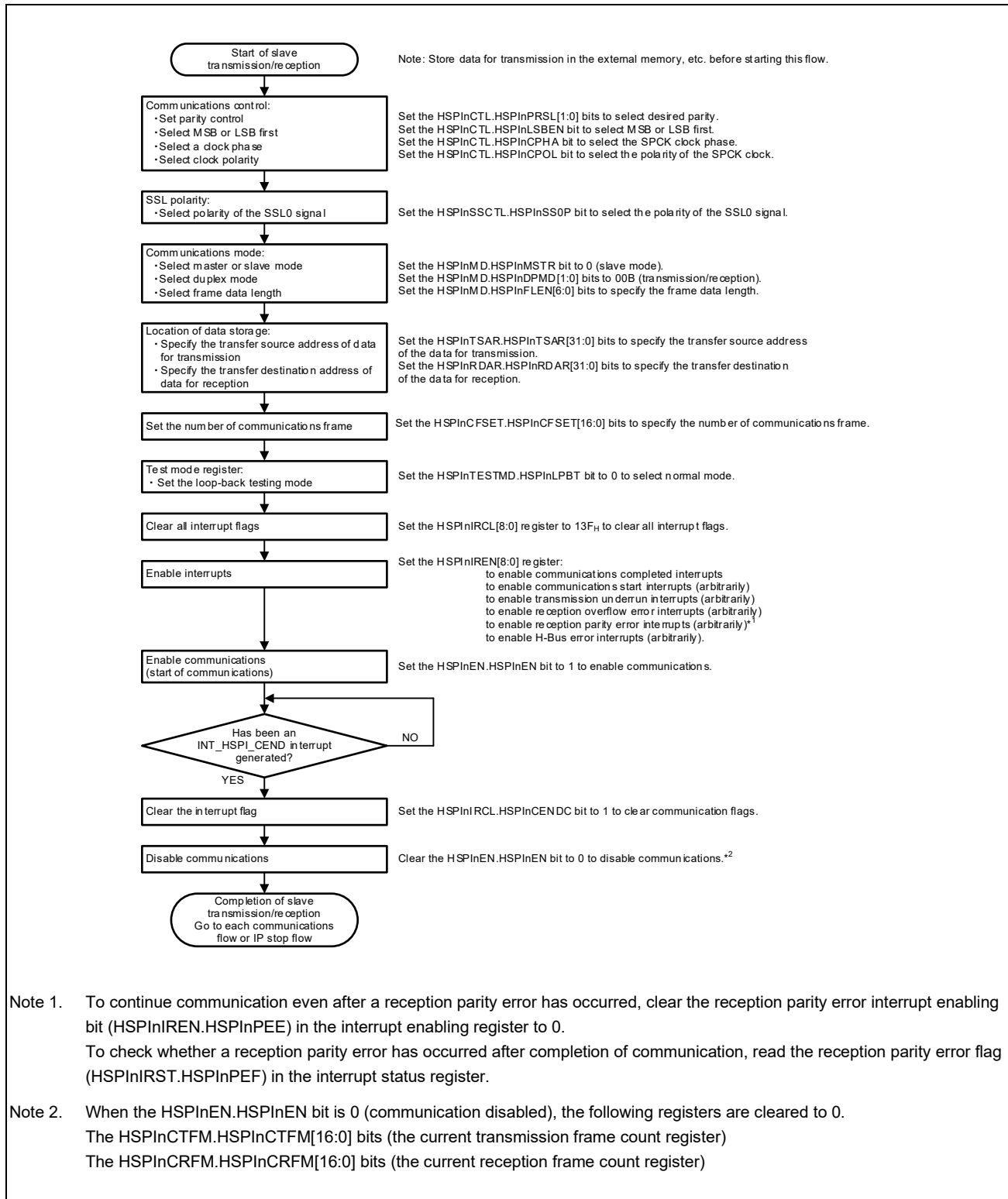


Figure 26.36 Example of Flow of Transmission/Reception Setting in Slave Mode

26.5.3.2 Transmission Setting Flow

Figure 26.37 shows an example of the flow for transmission setting in slave mode.

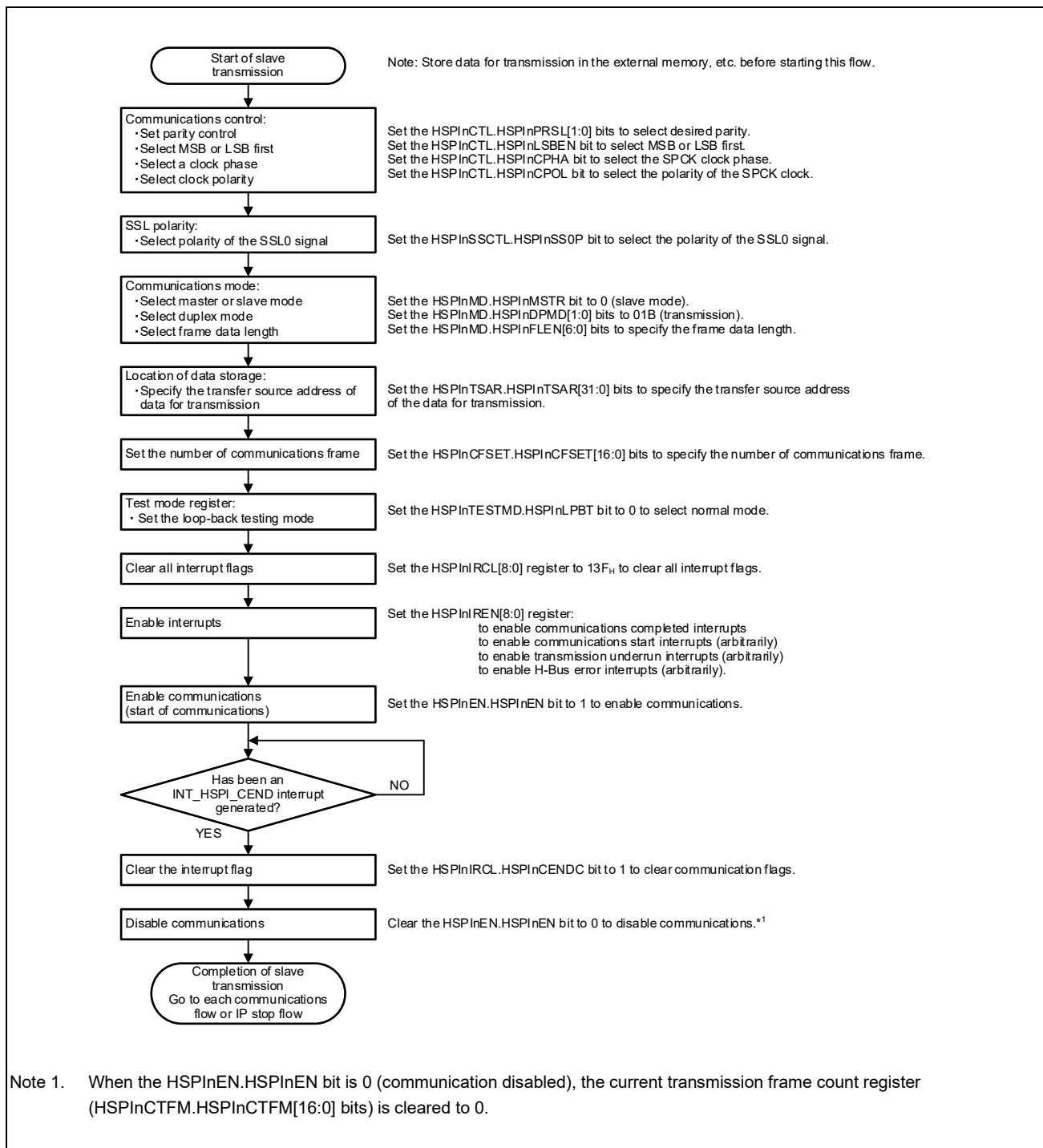


Figure 26.37 Slave Mode Transmission Setting Flow Example

26.5.3.3 Reception Setting Flow

Figure 26.38 shows an example of the flow for reception setting in slave mode.

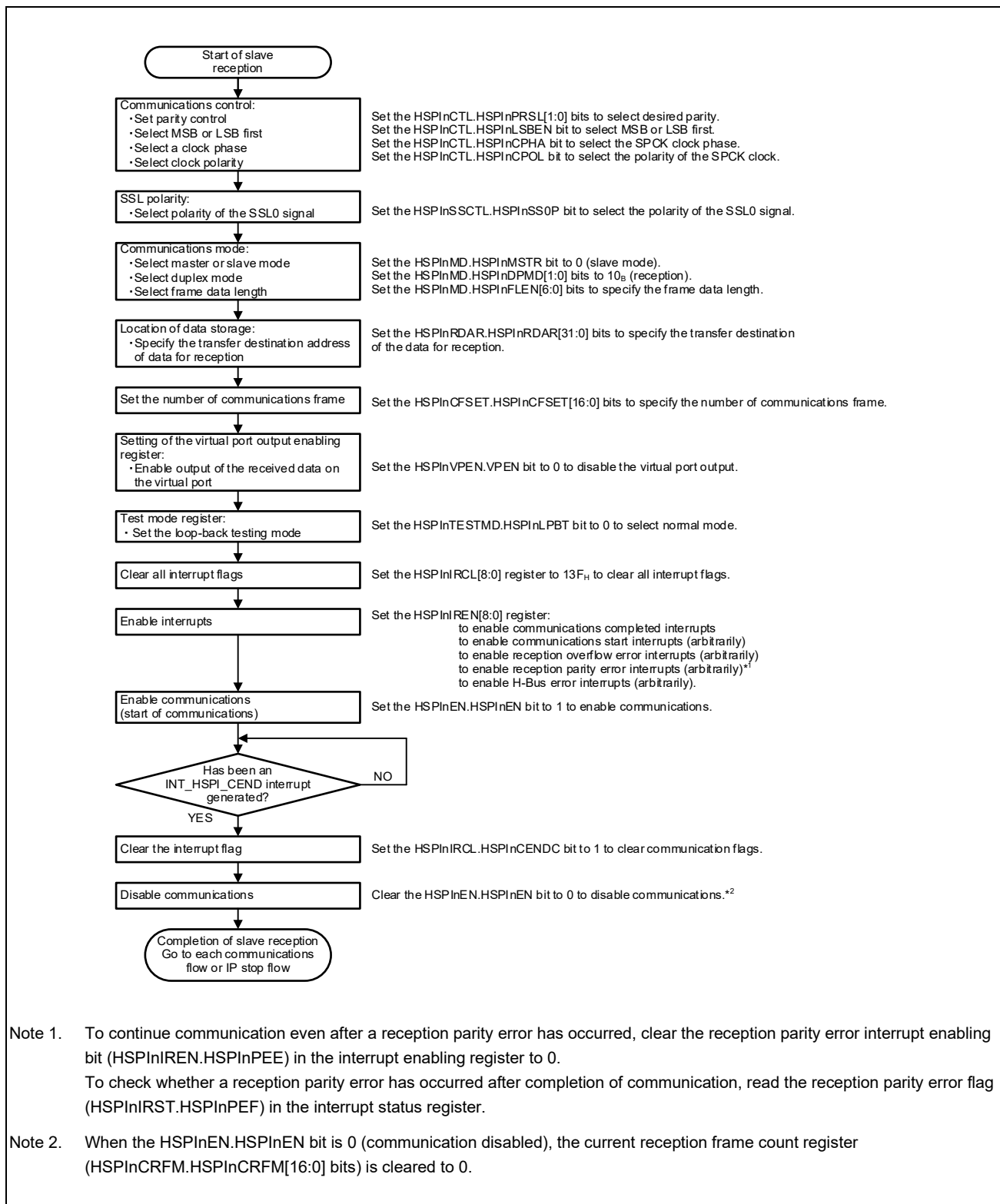


Figure 26.38 Slave Mode Reception Setting Flow Example

26.5.3.4 Virtual Port Reception Setting Flow

Figure 26.39 shows an example of the flow for the reception setting using the virtual port.

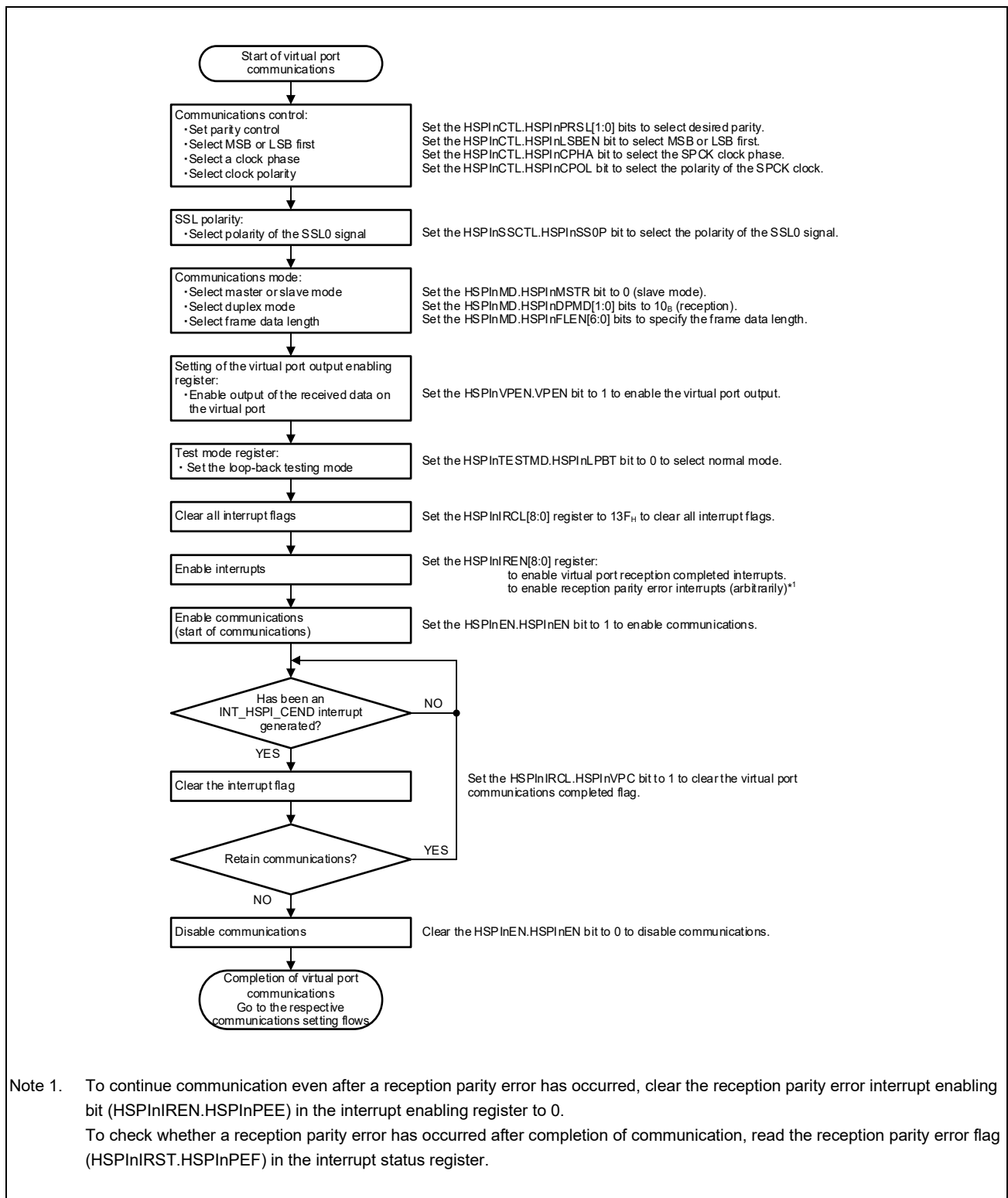


Figure 26.39 Slave Mode Virtual Port Reception Setting Flow Example

26.5.3.5 Flow of Loop-back Test Setting

Figure 26.40 shows an example of the flow for the loop-back test setting in slave mode.

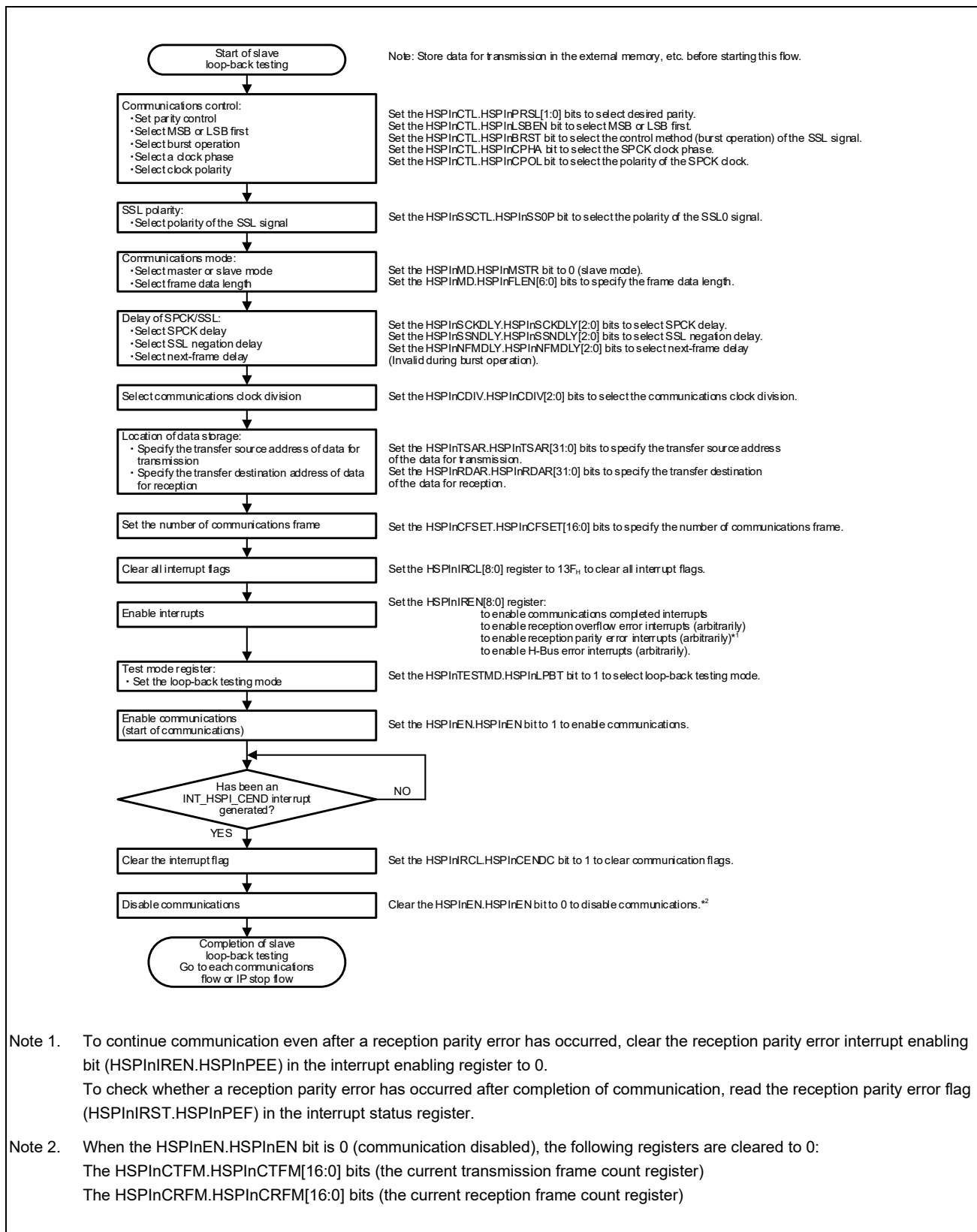


Figure 26.40 Slave Mode Loop-back Test Setting Flow

26.5.4 Flow of Re-send Using Communications Start Trigger

Figure 26.41 is a flowchart of the resend process using the communication start trigger.

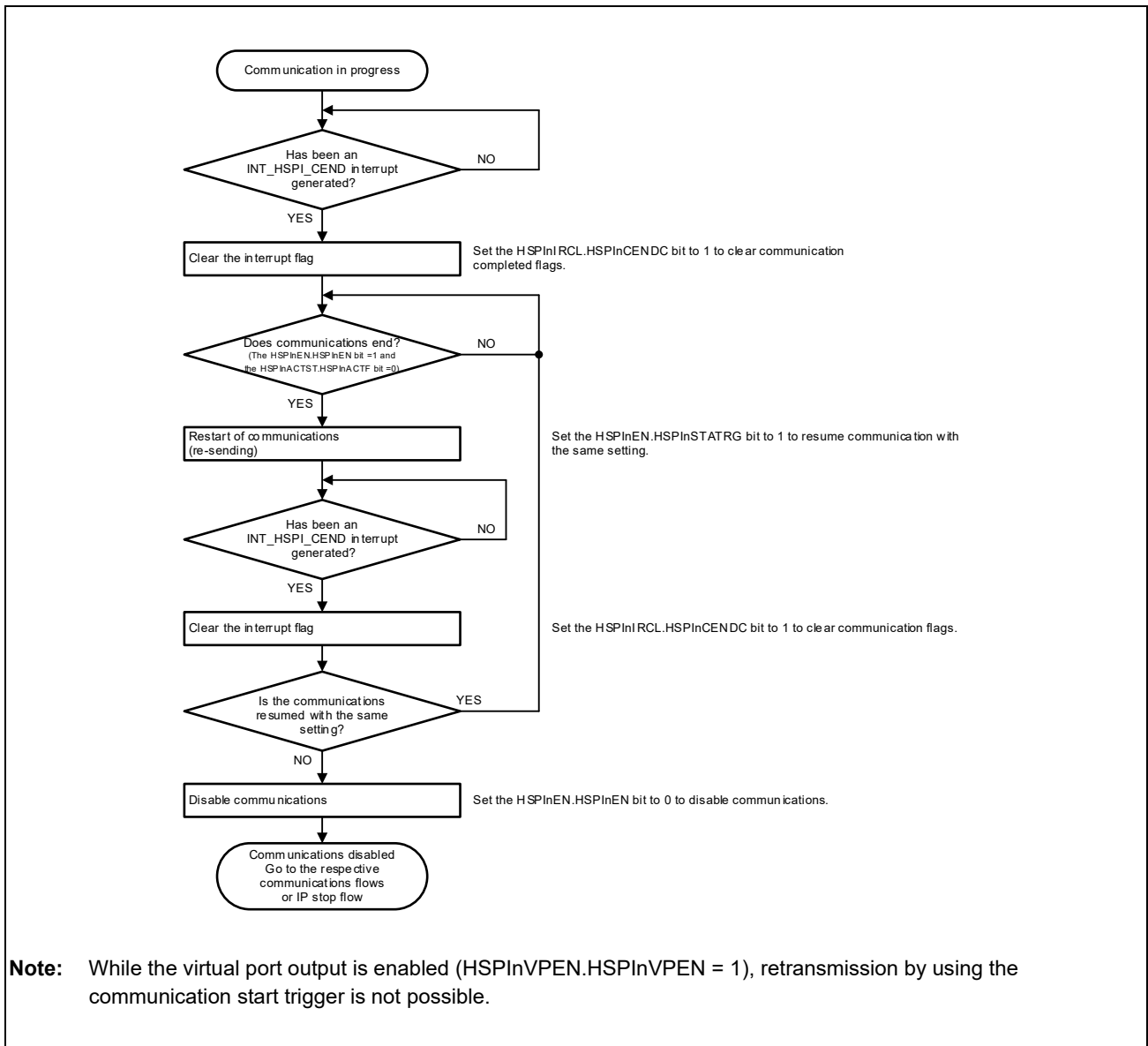


Figure 26.41 Flow of Re-send Using the Communications Start Trigger

26.5.5 Error Handling Flow

Figure 26.42 is a flowchart of error handling.

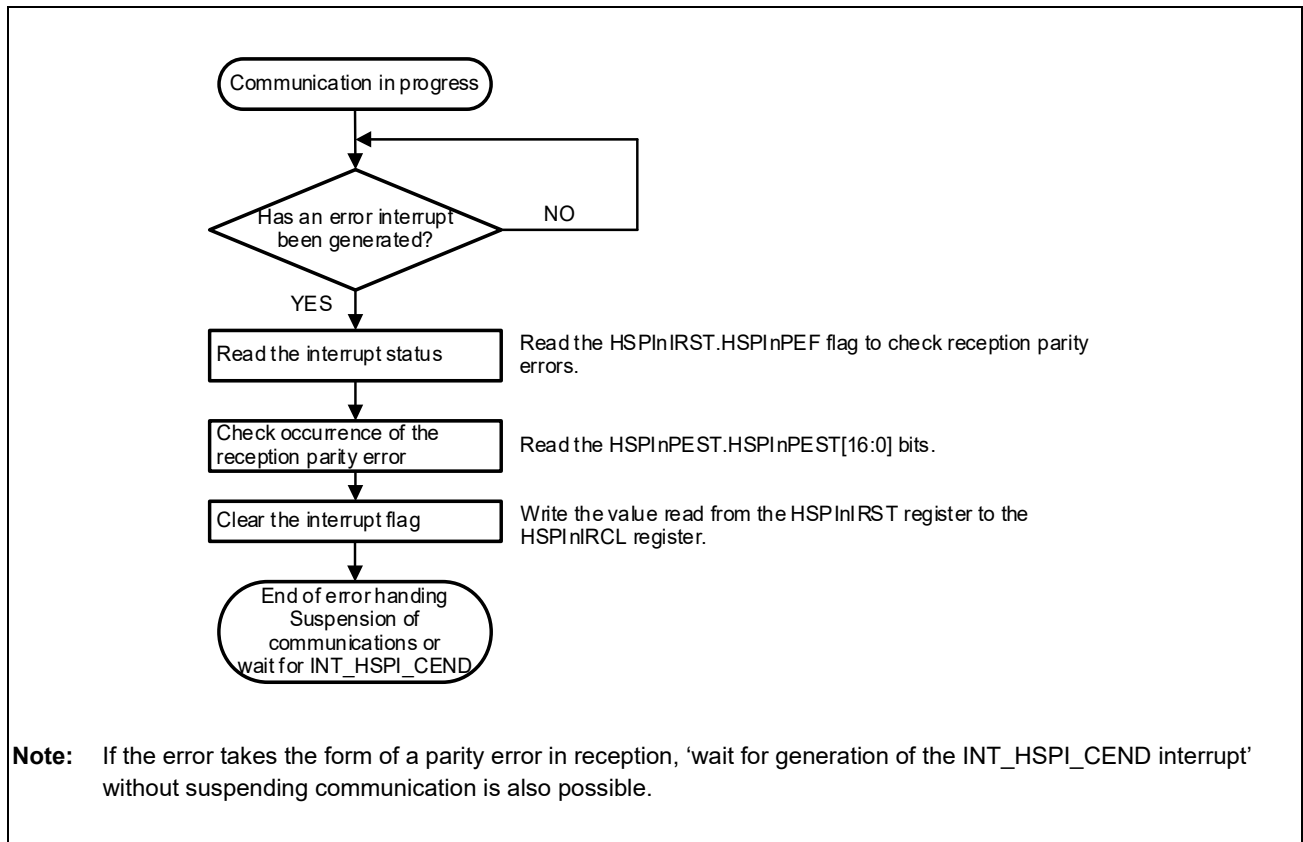


Figure 26.42 Error Handling Flow

26.5.6 Suspend Communication Procedure

Figure 26.43 shows how to suspend communication.

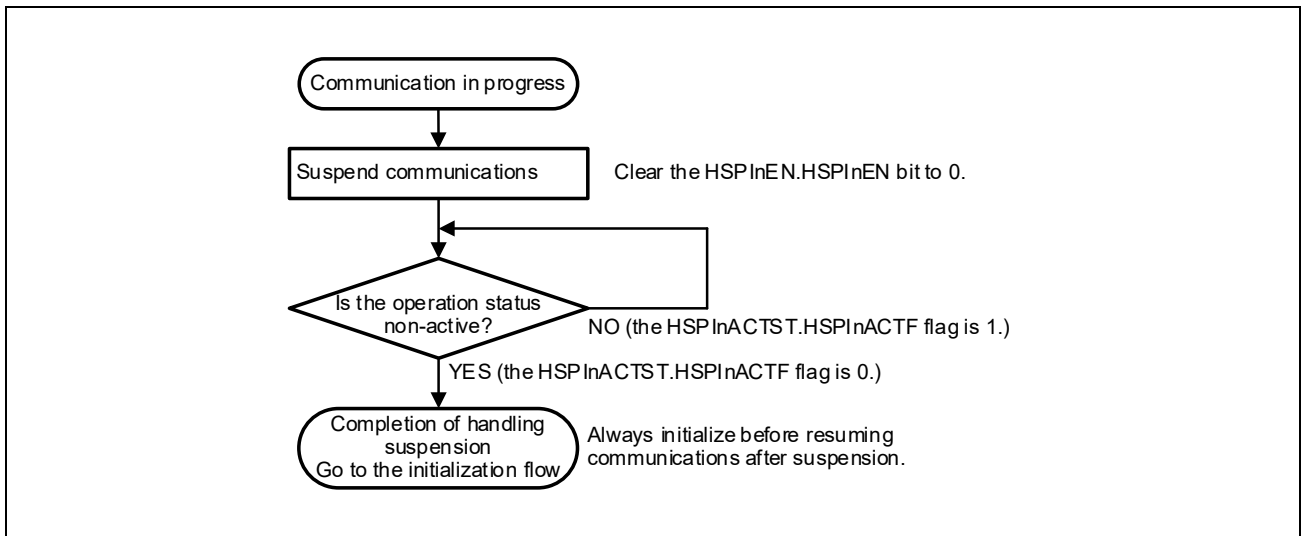


Figure 26.43 Flow of Suspending Communication

26.5.7 Notes for Suspending Communication

Always apply a software reset to resume communication after suspension.

When a software reset is applied while SSL0 is asserted and SPCK is input while slave communication is suspended, operation of the shifters in the transmission or reception sections may become metastable. Use the upper system to ensure a state in which SSL0 is negated or SPCK is not operating.

26.5.8 Loop-Back Testing

The HS-SPI supports loop-back testing mode.

In loop-back testing mode, transmitted data is looped back and received by the HS-SPI. Loop-back testing modes are provided for master and slave operations.

In loop-back testing mode, the communication clock that is generated in the master transmission section and slave select signal are used for communication, regardless of master mode or slave mode.

Table 26.40 summarizes the serial I/F pin states in loop-back testing mode.

Table 26.40 Pin State during Loop-back Testing

Mode	Pin State			
	SSL0	SPCK	MOSI	MISO
Master	Fixed to inactive	Fixed to idling	Fixed to low	Input is invalid
Slave	Input is invalid*1	Input is invalid	Input is invalid	Fixed to low*2

Note 1. Only output control of the MISO pin is enabled.

Note 2. The state of the MISO pin is low output while the HSPInEN.HSPInEN bit is 1 and the SSL0 signal is asserted. (The state of the MISO pin is Hi-Z while the SSL0 signal is negated.)

26.6 Notes

26.6.1 Notes for Suspending Communication

Always apply a software reset to resume communication after suspension. For details, see **Sections 26.4.7, Reception and Transmission during Suspension of Transfer, 26.5.6, Suspend Communication Procedure, and 26.5.7, Notes for Suspending Communication.**

26.6.2 Register Write Protection and Software Resets

26.6.2.1 Write Protection

Register write protection functions only while the communication enabling bit (HSPInEN.HSPInEN) is 1 and writing to respective registers targeted for write protection is not disabled.

Write these registers for write protection while the communication enabling bit (HSPInEN.HSPInEN) is 0 and the operation status flag (HSPInACTST.HSPInACTF) is 0.

26.6.2.2 Registers for Software Resets

Write protection does not depend on the status of the operation status flag (HSPInACTST.HSPInACTF). When a software reset is applied while the operation status flag (HSPInACTST.HSPInACTF) is 1, however, communication is immediately suspended even while in progress. The H-Bus transaction is also suspended.

Therefore, write the Software Reset Register while the communication enabling bit (HSPInEN.HSPInEN) is 0 and the operation status flag (HSPInACTST.HSPInACTF) is 0.

26.6.2.3 Notes when Using Communications Start Trigger Bit

To use the communication start trigger bit (HSPInEN.HSPInSTATRG), write 1 to the bit while the communication enabling bit (HSPInEN.HSPInEN) is 1 and communication is disabled (when the operation status flag (HSPInACTST.HSPInACTF) is 0).

26.6.3 DMA Function

26.6.3.1 DMA Operation during Suspended Communication

When the communication enabling bit (HSPInEN.HSPInEN) is modified from communication enabled (HSPInEN is 1) to communication disabled (HSPInEN is 0), a DMA transfer stops after completion of the DMA transfer in progress (H-Bus transaction), not immediately.

26.6.3.2 DMA Operation in Response to H-Bus Errors

When an error response is reported by the H-Bus during DMA transfer, the H-Bus error flag (HSPInIRST.HSPInBEF) becomes 1 and DMA transfer continues.

For details on H-Bus errors, see **Section 26.4.6.2, H-Bus Error Interrupt Requests**.

26.6.3.3 Transmission Underruns in Slave Mode

If the address from which DMA transfer is to start is close to the second half of a range between the 64-byte boundaries and the address exceeds the boundary soon after the start of transfer, a transmission underrun is more likely since this may make it impossible to prepare data for transmission in time.

26.6.3.4 DMA Operation When Virtual Port Output Enabled

Setting the virtual port output enabling bit (HSPInVPEN.HSPInVPEN) to 1 outputs the serial received data to the virtual port (HSPI_RDATA[63:0]) in slave reception. Note that received data is not stored in the reception buffer, nor is received data DMA-transferred to external memory in this case.

26.6.4 Data Storage and Address Space

Store the frames of data for transmission in alignment with the external memory area. For the order of transmission data, see **Figure 26.20** and **Figure 26.21**.

The frames for transmission are read from the external memory in units of data reading of the specified frame length.

The frame data for reception is also stored in the external memory in units of data writing of the frame data length specified by the DMAC in the HS-SPI.

Table 26.41 Relationship between the Frame Length and Data Reading/Writing Units

Frame length	HSPInMD.HSPInFLEN[6:0]	Data reading/writing units
8 bits	08 _H	8 bits
16 bits	10 _H	16 bits
24 or 32 bits	18 _H , 20 _H	32 bits
40, 48, 56, or 64 bits	28 _H , 30 _H , 38 _H , 40 _H	64 bits

- The area that support unaligned transfers of HS-SPI
- When the address where reading or writing starts is not within the 64-byte boundary, the burst length of the first H-Bus request is adjusted so that the second and subsequent H-Bus requests will be on 64-byte boundaries as shown in **Figure 26.44**. For this reason, the first H-Bus request is unaligned transfer from within the 64-byte boundaries. The last H-Bus request is transferred to byte-lanes within the 64-byte boundaries. At this time, no data is written to the invalid data area which includes no valid data, and these data are discarded in reading.

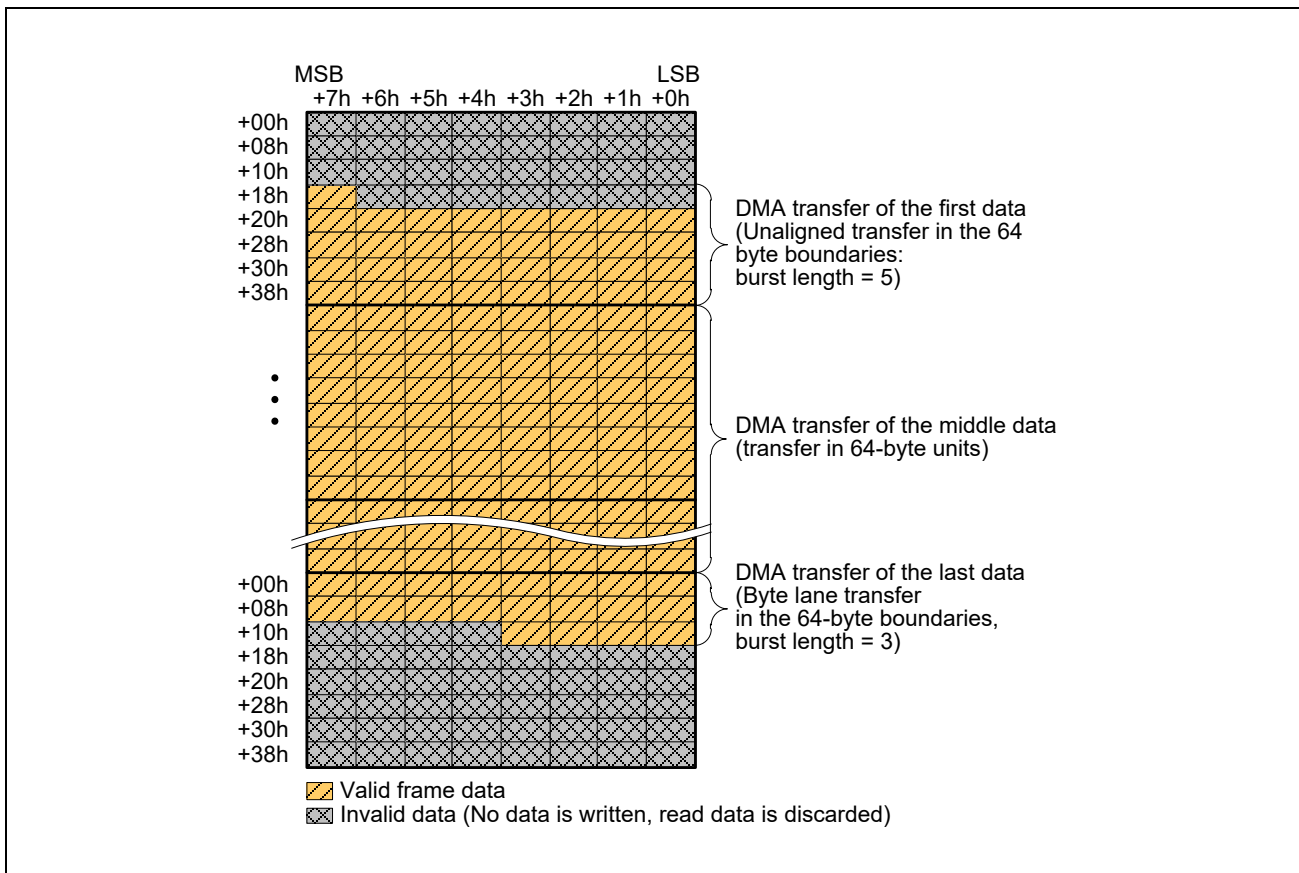


Figure 26.44 Communication Frame Data Storage Area (8-bit Frame Data Length)

- The area that does not support unaligned transfers of HS-SPI

When the address where reading or writing is to start is not within the 64-byte boundary, the burst length of the first H-Bus request is adjusted so that the second and subsequent H-Bus requests will be on 64-byte boundaries as shown in **Figure 26.45**. The last H-Bus request is transferred to byte-lanes within the 64-byte boundaries. At this time, no data are written to the invalid data area, and these data are discarded in reading.

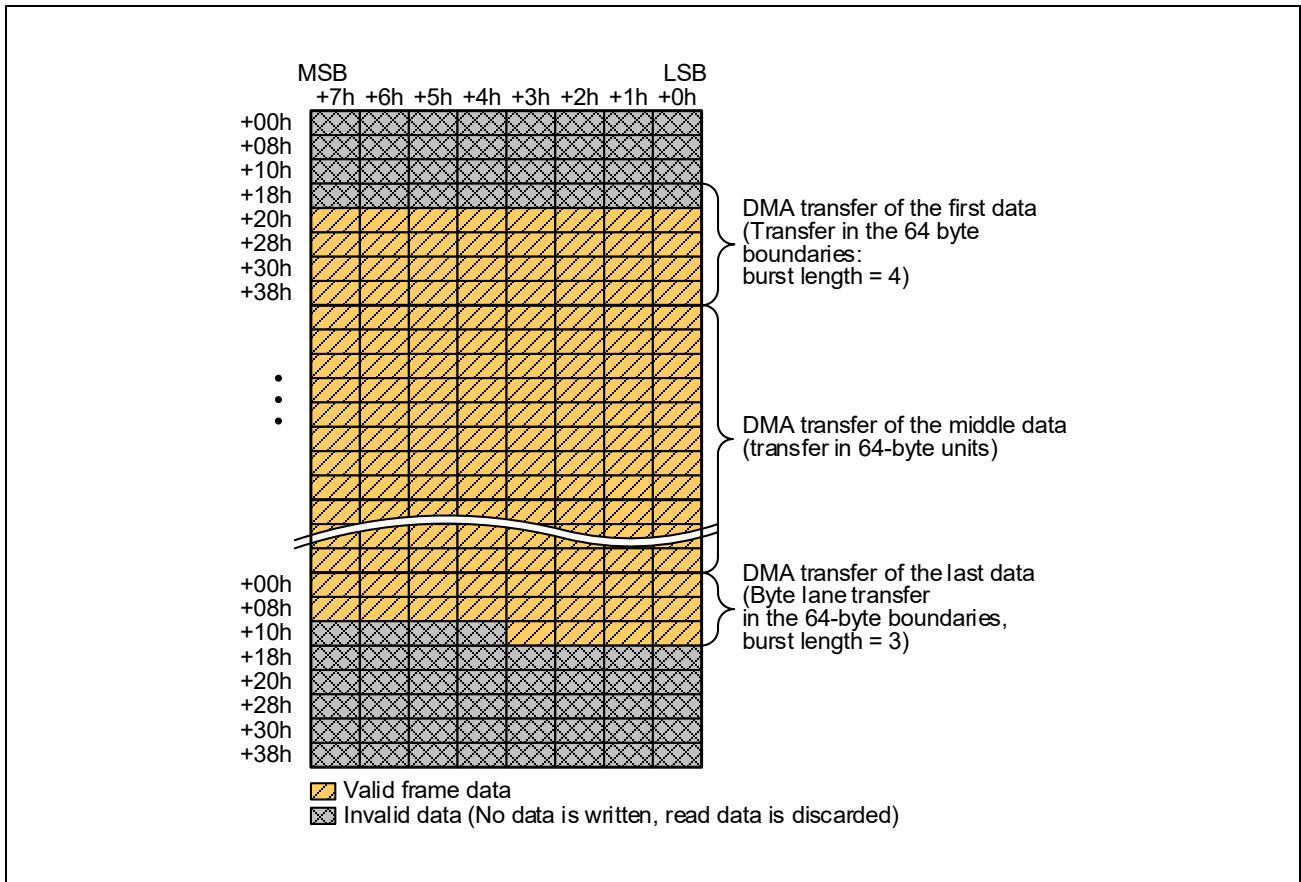


Figure 26.45 Communications Frame Data Storage Area (8-bit Frame Data Length)

26.6.5 Errors

26.6.5.1 Resume Transfer after an Error Occurs

To resume transfer after an error occurs, service routines should include processes to handle the error, suspension and initialization.

26.6.5.2 Operations When a Reception Overflow Occurs

The number of frames transferred from external devices becomes inconsistent in cases of reception overflow. The communication complete flag (HSPInIRST.HSPInCENDF) may not be set to 1 even after the reception overflow error flag (HSPInIRST.HSPInROVFEF) is cleared to 0. When a reception overflow occurs, disable transfer (HSPInEN.HSPInEN = 0) without waiting for completion.

26.6.5.3 Transmission Underrun in Slave Mode

The number of frames transferred from the external devices becomes inconsistent in cases of transmission underrun. The communication complete flag (HSPInIRST.HSPInCENDF) may not be set to 1 even after the transmission underrun error flag (HSPInIRST.HSPInTUDREF) is cleared to 0. When a transmission underrun occurs, disable transfer (HSPInEN.HSPInEN = 0) without waiting for completion.

Section 27 Renesas High-Speed Serial I/F (RHSIF)

This section contains a generic description of the Renesas high-speed Serial I/F (RHSIF). The first part in this section describes all this product specific properties, such as the number of units and register base addresses. The remainder of this section describes the functions and registers of the RHSIF.

27.1 Feature

27.1.1 Number of Units and Channels

This microcontroller has the following number of Renesas high-speed Serial I/F (RHSIF) units.

Table 27.1 Number of Units

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Units	1	1
Name	RHSIFn (n = 0)	RHSIFn (n = 0)

Table 27.2 Number of Units

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Units	1	1
Name	RHSIFn (n = 0)	RHSIFn (n = 0)

Note: Regarding the unit index “n” in this section, the individual RHSIF units are identified by the index “n”.

27.1.2 Register Base Address

RHSIFn base addresses are listed in the following table. RHSIFn register addresses are given as offsets from the base address in general.

Table 27.3 Register Base Address

Base Address Name	Base Address	Bus Group
<RHSIF0_L1_base>	FF00 0000 _H	Peripheral Group 9
<RHSIF0_L2_base>	1000 0000 _H	H-Bus Group 0

27.1.3 Clock Supply

Clock supply by and to RHSIFn is listed in the following table.

Table 27.4 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
RHSIFn (L2)	H-Bus clock	CLK_HBUS
RHSIFn (L1)	PCLK	CLK_HBUS
	CLK20MHz	CLK_MOSC
	Communication clock (Internal)	
	RHSIF0_REFCLK	HSIF0_REFCLK
	Communication clock (External)	

27.1.4 Interrupt Requests

RHSIFn interrupt requests are listed in the following table.

Table 27.5 Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number	DTS Trigger Number
RHSIF0					
INTRHSIF0TXCMP	RHSIFnTXCMP (n = 0)	Transmit complete interrupt (unit n)	456	—	—
INTRHSIF0TXEX	RHSIFnTXERR (n = 0)	Transmit exception interrupt (unit n)	457	—	—
INTRHSIF0RXCMP	RHSIFnRXCMP (n = 0)	Receive complete interrupt (unit n)	458	—	—
INTRHSIF0RXEX	RHSIFnRXERR (n = 0)	Receive exception interrupt (unit n)	459	—	—
INTRHSIF0ICLCR	RHSIFnRXICLC (n = 0)	ICLC receive interrupt (unit n)	460	—	—
INTRHSIF0INTCH0	int_hsif_ch0	Channel 0 interrupt	461	group0-185	group1-93
INTRHSIF0INTCH1	int_hsif_ch1	Channel 1 interrupt	462	group0-186	group1-94
INTRHSIF0INTCH2	int_hsif_ch2	Channel 2 interrupt	463	group0-187	group1-95
INTRHSIF0INTCH3	int_hsif_ch3	Channel 3 interrupt	464	group0-188	group1-96
INTRHSIF0STR	int_hsif_str	Stream interrupt	465	—	—
INTRHSIF0ERR	int_hsif_err	Error interrupt	466	—	—

27.1.5 Reset Sources

RHSIFn reset sources are shown below. RHSIFn is initialized by the following reset sources.

Table 27.6 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
RHSIFn	All registers	✓	✓	✓	✓	—	—

27.1.6 External Input/Output Signals

External input/output signals of RHSIF units are listed below.

Table 27.7 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
RHSIF0		
RHSIF0_REFCLK	RHSIF0 Reference clock input/output	HSIF0_REFCLK
RHSIF0_RXDN	RHSIF0 receive data differential input	HSIF0_RXDN
RHSIF0_RXDP	RHSIF0 receive data differential input	HSIF0_RXDP
RHSIF0_TXDN	RHSIF0 transmission data differential output	HSIF0_TXDN
RHSIF0_TXDP	RHSIF0 transmission data differential output	HSIF0_TXDP

27.2 Overview

27.2.1 Functional Overview

- Four channels, including one channel with data streaming capability
- Bus master interface used by target node to access shared memory
- H-Bus bus master supporting “Write/Read” command
 - Writing a single 8 / 16 / 32 bit data value into the register of a target device
 - Reading a single data from 8 / 16 / 32 bit register of a target device
 - Support of 32-bit address range
- Built-in DMA controller supporting “Stream write” command
 - 128 or 256-bit data transfer on streaming channel
 - Up to two outstanding requests of “Stream write” are supported by the initiator
- Transfers protected by CRC16
- Programmable time outs for detection of blocked answer transfers
- Automatic frame transfer ID generation for detection of dropped frames
- Fixed priority channel selection
- Remote trigger of event/interrupt in the target device by the initiator
- Access protection from an external master
 - Can configure four access windows
 - Can configure access rule (Prohibited, R, W, RW) at each window
- Security function
 - Authentication at the time of initial communication
 - ID authentication
 - Challenge and response authentication
- Point-to-point high speed serial communication between two devices
- Full duplex communication
- Supports dual mode (register configurable Master/Slave).
- Supports asynchronous data transfer at data rates up to a maximum of 160 M Baud
- Transmits and receives data, CTS, ICLC and unsolicited frames
- Supports automatic ping response generation in slave mode
- Supports detection of unsupported channel number and unsupported payload sizes
- The interface is based on IEEE 1596.3-1996 reduced range link LVDS IOs
- Built-in self test functions

27.2.2 Block Diagram

The following figure shows a top level block diagram of the RHSIF module.

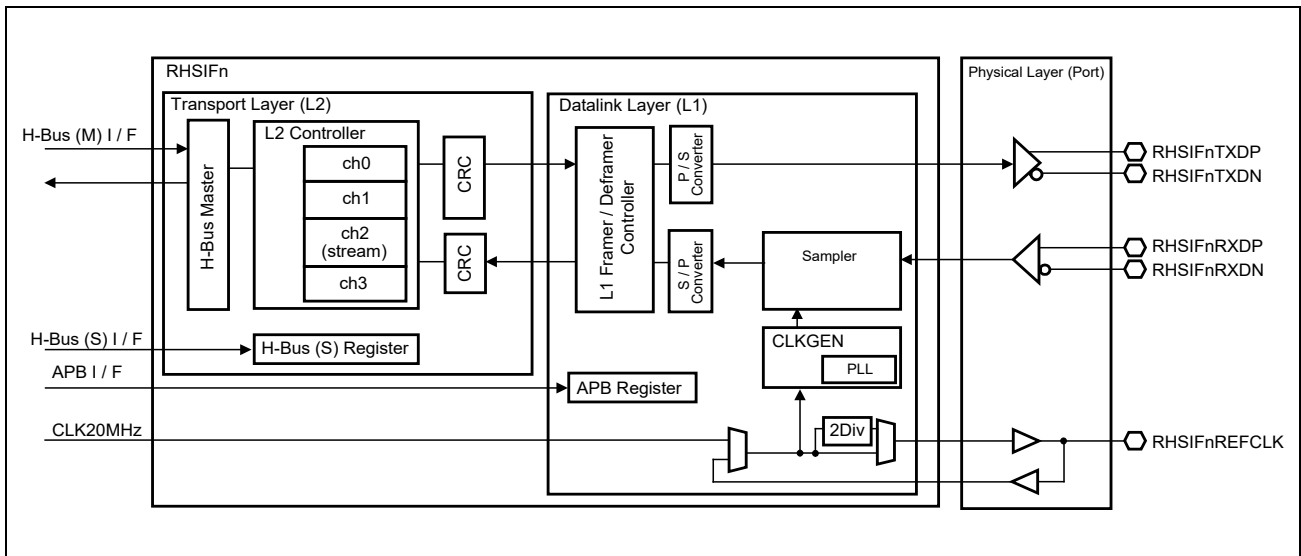


Figure 27.1 Module Block Diagram

27.2.3 Communication Protocol

27.2.3.1 Protocol Definitions

Table 27.8 Protocol Definitions

Item	Definition	Comment
Master	The module that supplies the clock and controls Link initialization on the Slave interface.	L1
Slave	The module that receives the clock.	L1
Initiator	The module that sends an L2 frame with either the Write, Read, Stream, Event, or ID command	L2
Target	The module that sends an L2 frame with either the Read Answer, ACK, or NACK command	L2
TxLink	Point-to-point connection executing transfer from Master to Slave.	L1
RxLink	Point-to-point connection executing transfer from Slave to Master.	L1

27.2.3.2 Connection

Figure 27.2 shows the connection between master and slave of RHSIF.

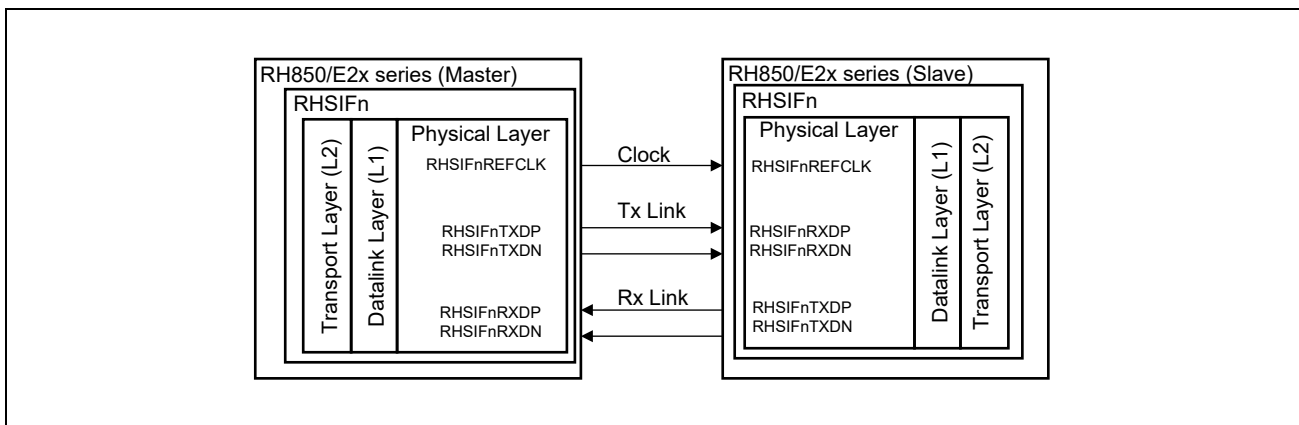


Figure 27.2 Connection between Master and Slave of RHSIF

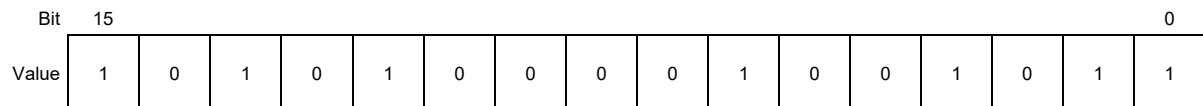
27.2.3.3 L1 Frame

The L1 Frame Format is shown below.



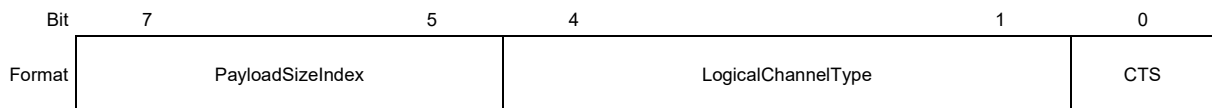
(1) SyncCode

The value in this field is assigned to SyncCode, and the value is fixed in A84B_H



(2) L1 Header Format

The value in this field is assigned to L1 Header.



(a) List of Payload Size Index

Table 27.9 shows the details of the payload size index values.

Table 27.9 List of Payload Size Index

Payload Size Index	Payload Size	Total Frame Size* ¹	Comment
000	8	32	Interface Control
001	32	56	—
010	64	88	—
011	96	120	—
100	128	152	Not used
101	256	280	Not used
110	160	184	—
111	288	312	—

Note 1. "Endbit" is excluded from Total Frame Size.

(b) List of Logical Channel Types

Table 27.10 shows the details of Logical Channel Type values.

Table 27.10 List of Logical Channel Types

Logical Channel Type	Logical Channel Type Master	Logical Channel Type Slave
0000	Interface Control	Interface Control PING answer (32 bit value)
0001	Reserved for future use	
0010	Reserved for future use	
0011	CTS frame for flow control	
0100	Data Channel A	
0101	Data Channel B	
0110	Data Channel C	
0111	Data Channel D	
1000-1111	Reserved for future use	

(c) List of CTS

Table 27.11 shows the details of CTS values.

Table 27.11 List of CTS

CTS	Function
0	BackPressure
1	Ready

(3) L1 Payload

The value in this field is assigned to L1 Payload as ICLC Command.

**(a) List of ICLC Commands**

Table 27.12 shows the details of ICLC command values.

Table 27.12 List of ICLC Commands

Value (hex)	Function
0x00	PING (Sends by master interface. Slave sends back a fixed 32-bit payload result (0xABCDEF01))
0x02	Slave interface PLL start (in preparation for high speed mode)
0x04	Slave interface PLL stop (after fallback from high speed mode)
0x08	Select Slow Speed mode for transfers from the Master interface to the Slave interface
0x10	Select Fast Speed mode for transfers from the Master interface to the Slave interface
0x20	Select Slow Speed mode for transfers from the Slave interface to the Master interface
0x80	Select Fast Speed mode for transfers from the Slave interface to the Master interface
0x31	Enable Slave interface transmitter
0x32	Disable Slave interface transmitter
0x34	Turn on test mode (Send 101010 ... on RX line continuously using actual configured RX line data rate; cancelled by issuing "test mode off" command.)
0x38	Turn off test mode (Cancels the clock test mode and payload loopback)
0xFF	Turn on payload loopback (incoming frames at Slave interface are looped back until cancelled by a Frame containing "test mode off".)

(4) Endbit

The value in this field is assigned to Endbit.

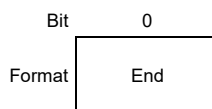
**(a) List of Endbit**

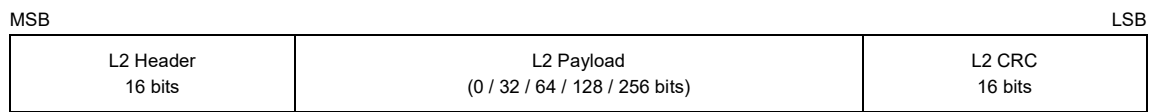
Table 27.13 shows the details of the Endbit values.

Table 27.13 List of Endbit

Endbit	Function
0	Normal
1	Sleep

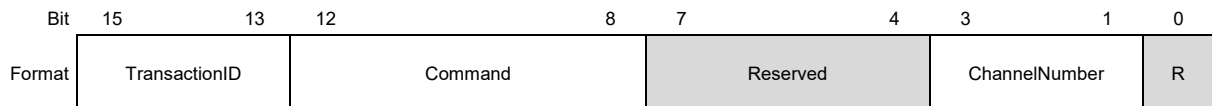
27.2.3.4 L2 Frame

The L2 Frame Format is shown below.



(1) L2 Header Format

The value in this field is assigned to L2 Header.



(a) Transaction ID

The value in this field is assigned to the outgoing frame. Afterwards, it is automatically increased and the new value is assigned to the next outgoing frame. It wraps around.

(b) List of L2 Commands

Table 27.14 shows the details of the L2 Command values.

Table 27.14 List of L2 Commands

Header Bit Number					Command
12	11	10	9	8	
0	0	0	0	0	Reads 8 bits
0	0	0	0	1	Reads 16 bits
0	0	0	1	0	Reads 32 bits
0	0	0	1	1	Reserved
0	0	1	0	0	Writes 8 bits
0	0	1	0	1	Writes 16 bits
0	0	1	1	0	Writes 32 bits
0	0	1	1	1	Reserved
0	1	0	0	0	ACK
0	1	0	0	1	NACK (ACK Error)
0	1	0	1	0	Read Answer
0	1	0	1	1	Reserved
0	1	1	0	0	Event Command (Trigger)
0	1	1	0	1	Reserved
0	1	1	1	0	Reserved
0	1	1	1	1	Reserved
1	0	0	0	0	Reserved
1	0	0	0	1	Reserved
1	0	0	1	0	ID Read (Read JTAG ID)*1
1	0	0	1	1	Reserved
1	0	1	0	0	Reserved
1	0	1	0	1	Reserved
1	0	1	1	0	Reserved
1	0	1	1	1	Stream
1	1	0	0	0	Reserved
1	1	0	0	1	Reserved
1	1	0	1	0	Reserved
1	1	0	1	1	Reserved
1	1	1	0	0	Reserved
1	1	1	0	1	Reserved
1	1	1	1	0	Reserved
1	1	1	1	1	Reserved

Note 1 Refer to **Section 46, Boundary Scan, Table 46.5** for details.

(c) List of Channel Number

Table 27.15 shows the details of the Channel Number values.

Table 27.15 List of Channel Number

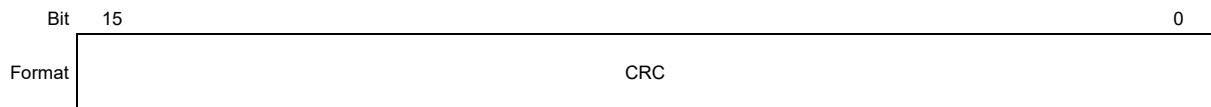
HSSL Channel Name	Channel Number Coding in Header			Comment
	HSSL Header (Binary)		Data Link Layer Header (Binary)	
	Code Table I	Code Table II		
0 (channel A)	000	100	0100	In use
1 (channel B)	001	101	0101	In use
2 (channel C)	010	110	0110	In use
3 (channel D)	011	111	0111	In use
4 (channel E)	100	000	1000	Reserved
5 (channel F)	101	001	1001	Reserved
6 (channel G)	110	010	1010	Reserved
7 (channel H)	111	011	1011	Reserved

(2) L2 Payload

The value in this field is assigned to actual data.

(3) L2 CRC

The value in this field is assigned to CRC.



The last field of the L2 frame contains the CRC field. To setup the CRC value, this protocol uses the following algorithm:

- CRC-16-CCITT polynomial $x^{16} + x^{12} + x^5 + 1$

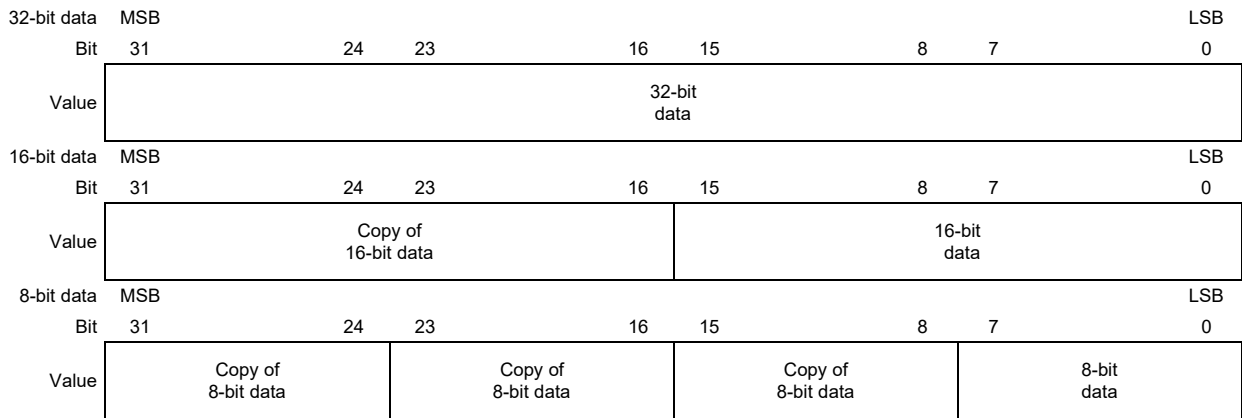
With the following standard properties:

- seed: 0xFFFF
- calculation direction: Most significant bit first (most significant bit to least significant bit in the header, and then the most significant bit to least significant bit in the payload)
- CRC result direction: Most significant bit first

27.2.3.5 Data Format

(1) Non Stream data format

Data is mapped into a 32-bit wide frame for 8 to 16 bits with no specific data frames defined. The following pictures show the mapping of 8 / 16 / 32 bit data into the 32 bit wide data frame.

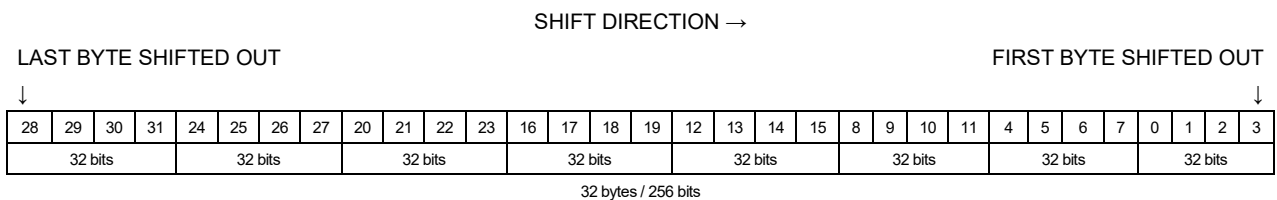


(2) Stream data format

(a) 256-bit Data Stream Format

Data is transmitted in a stream frame as a sequence of 8 times a 32-bit value. The initiator and target modules must be able to accept the data sequence in this format.

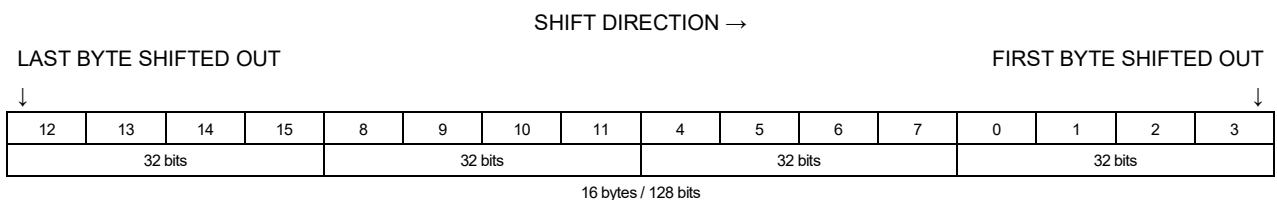
The bytes are sent with the most significant bit sent first.



(b) 128-bit Data Stream Format

Data is transmitted in a stream frame as a sequence of 4 times a 32-bit value. The initiator and target modules must be able to accept the data sequence in this format.

The bytes are sent with the most significant bit sent first.



27.2.3.6 All Sending and Receiving Frames

Figure 27.3 shows all sending and receiving frames.

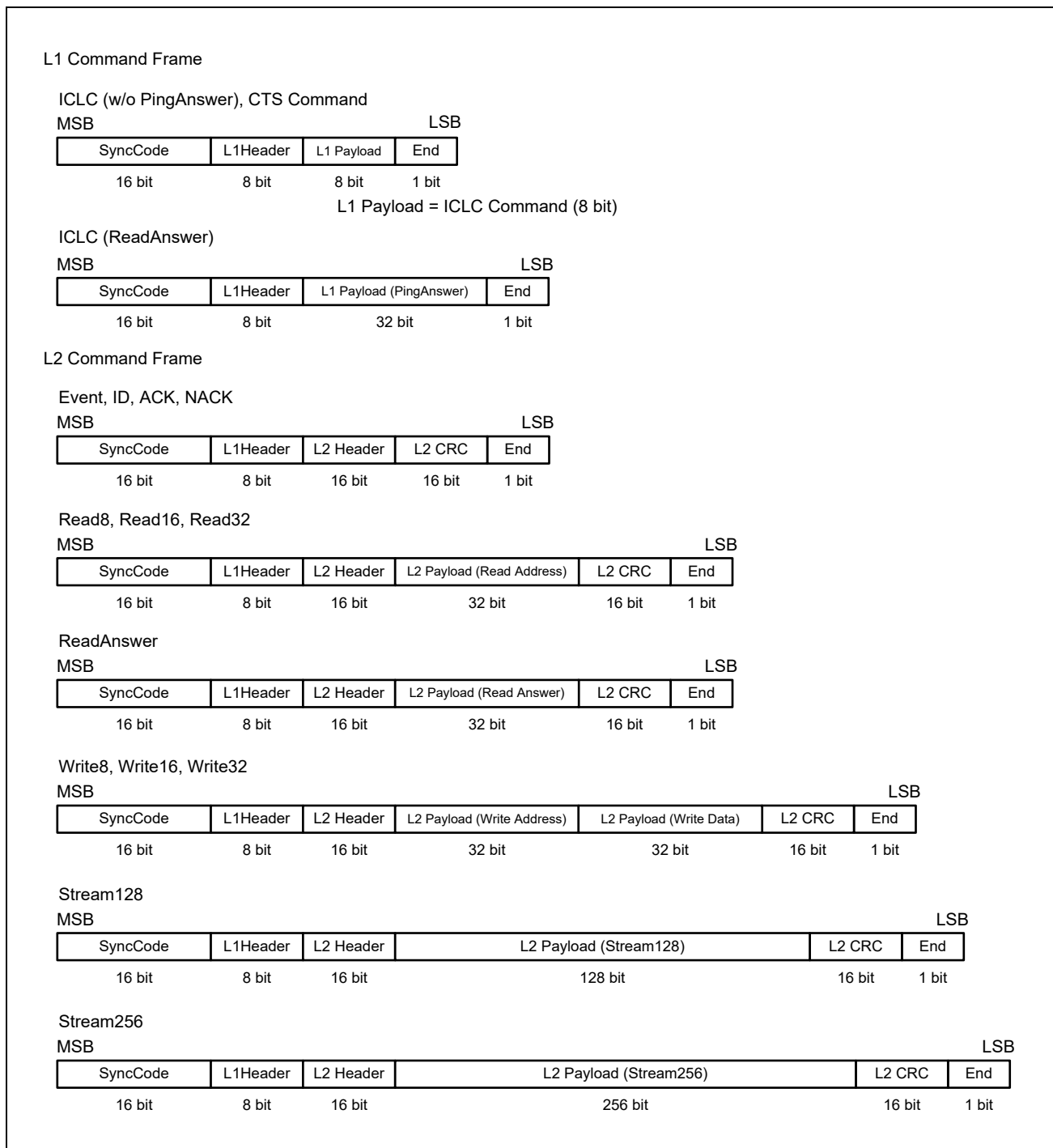


Figure 27.3 All Sending and Receiving Frames

27.2.3.7 Number of Transfer Bits in each Frame

Table 27.16 Number of Transfer Bits in each Frame List

#	Layer	L1			L2			L1 END	Total Bit	Command	Direction	
		Sync	Header	Payload	Header	Payload	CRC					
1	L1	16	8	8	—	—	—	1	33	ICLC (w/o PingAnswer)	Master → Slave	
2										CTS*1		
3				32					57	ICLC PingAnswer	Slave → Master	
4	L2	16	8	—	16	—	16	1	57	ID, Event	Initiator → Target	
5										ACK, NACK	Target → Initiator	
6										89	Read8, Read16, Read32	Initiator → Target
7											ReadAnswer	Target → Initiator
8										121	Write8, Write16, Write32	Initiator → Target
9										185	Stream128	Initiator → Target
10										256	Stream256	Initiator → Target

Note 1. Payload size is fixed to 8-bit data when transmitting and can be any size when receiving.

27.2.3.8 List of Command and Response

Table 27.17 List of Command and Response

#	Layer	Command Issue	Response	
			Normal	Abnormal
1	L1	Ping	PingAnswer	NoResponse
2	L2	ID	ReadAnswer	NACK
3		Event	ACK	NoResponse
4		Read8, Read16, Read32	ReadAnswer	NACK
5		Write8, Write16, Write32	ACK	NACK
6		Stream128, Stream256	ACK	NACK

27.2.3.9 Command Pipelining

The protocol does not allow pipelining of commands. The initiator shall not send any new command before the target has returned an answer (ACK, NACK, Read Answer). If the target receives a command before it is able to answer the previous command, it shall ignore the command.

There is one exception to this rule. Stream Commands can be pipelined. This feature is only available for Stream Commands. The initiator is allowed to send the next command before it has received an answer from the target linked per Transaction ID to the command sent before. The pipeline depth is limited to a sequence of two commands in the current protocol.

In summary, the initiator is allowed to send a Stream Command, which can be immediately followed by a second Stream Command. After the second command, the initiator has to stop and wait for an answer from the target.

The target shall be capable of receiving pipelined Stream Commands up to a pipeline depth of two Stream Commands.

27.3 Operation

27.3.1 Operation Flow

The RHSIF can perform data transmission/reception after the initial setting is configured according to the setup flow shown in **Figure 27.4** below. For details about the initial setting for L2 and L1, refer to **Section 27.4, Transport Layer (L2)** and **Section 27.5, Datalink Layer (L1)**, respectively.

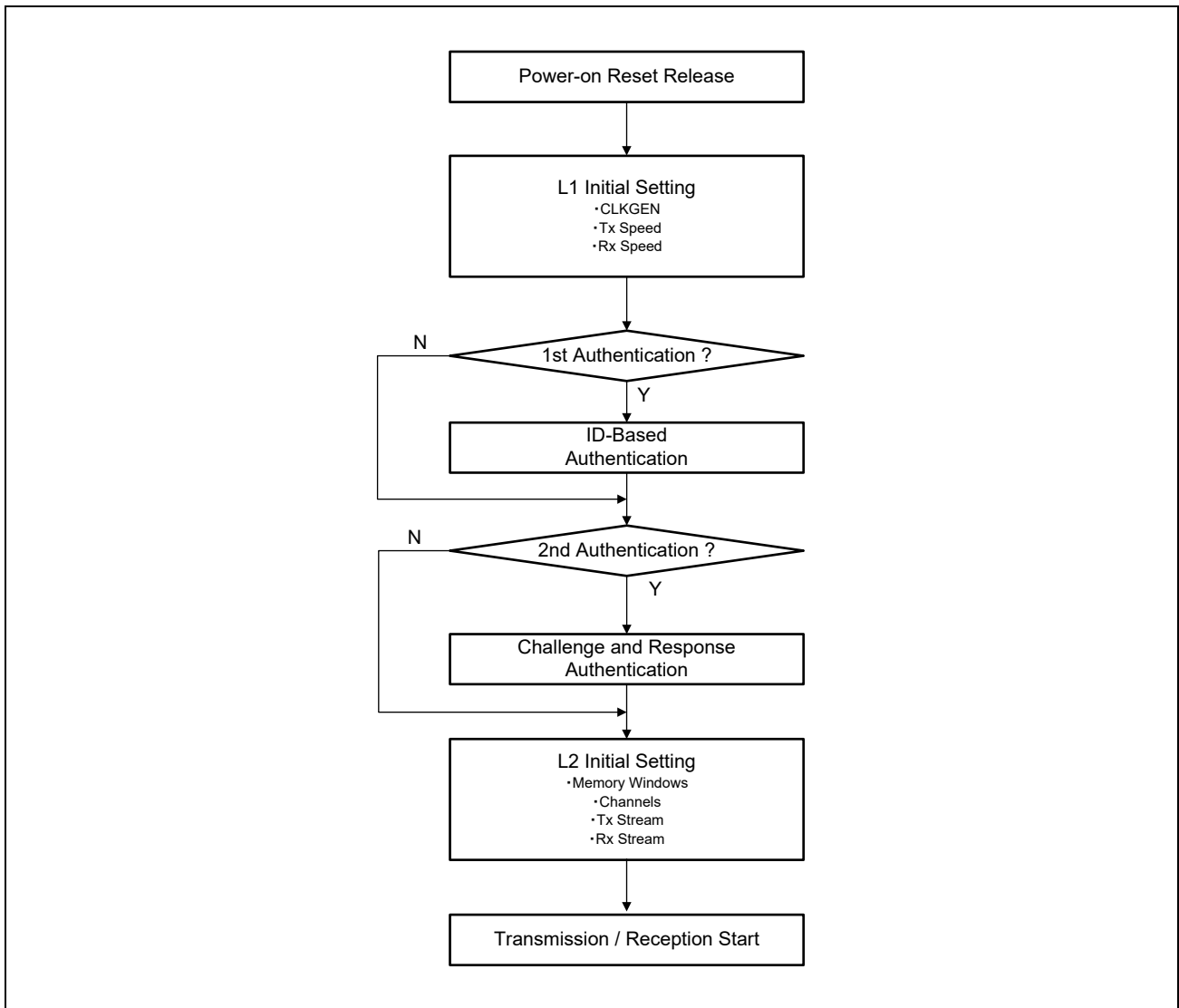


Figure 27.4 Operation Flow

27.4 Transport Layer (L2)

27.4.1 Overview

27.4.1.1 Functional Overview

The Renesas High-speed Serial Interface Transport Layer IP (hereinafter referred to as “L2”) is compliant with the following specifications.

L2 supports following features/functions.

- Supports 4 channels
 - Supports stream transfer on channel 2
 - Channel arbitration: Fixed (Higher) Ch.0 → Ch.1 → Ch.2 → Ch.3 (lower)
- Generates header
 - Manages transaction-ID
 - Supports both channel number code table I and II
- Generates and validates CRC-16
- Supports both initiator node and target node
- Initiator node function
 - Generates and transmit following request Commands
 - Read Command (8 bits, 16 bits, 32 bits)
 - Write Command (8 bits, 16 bits, 32 bits)
 - Event Command
 - ID Command
 - Stream Command (256 bits, 128 bits)
 - Number of outstanding requests: Non-Stream = 1 / channel, Stream = 2 (channel 2 only)
 - Detects reply timeout
 - The timeout value is programmable by register setting
 - Implements DMA controller for stream transmission
- Target node function
 - Number of acceptable requests: Non-Stream = 1 / channel, Stream = 2 (channel 2 only)
 - Translates HSSL Commands to H-Bus master requests
 - Refer to the **Section 38, Functional Safety** about the H-Bus SPID setting.
 - Supports the access protect function from an external master
 - Supports 4 areas, independent of channel
 - Can configure access rule (prohibited, read only, write only, read/write)
 - Can configure memory access window
 - Supports special authentication based on the security policy of this product
 - 256-bit ID authentication
 - “Challenge and response” authentication

- Generates and transmits the following reply Commands as target node
 - ACK/NACK Command
 - Read Answer Command
- Implements the DMA controller for stream reception
- Generates interrupt by receiving Event Command

27.4.1.2 Block Diagram

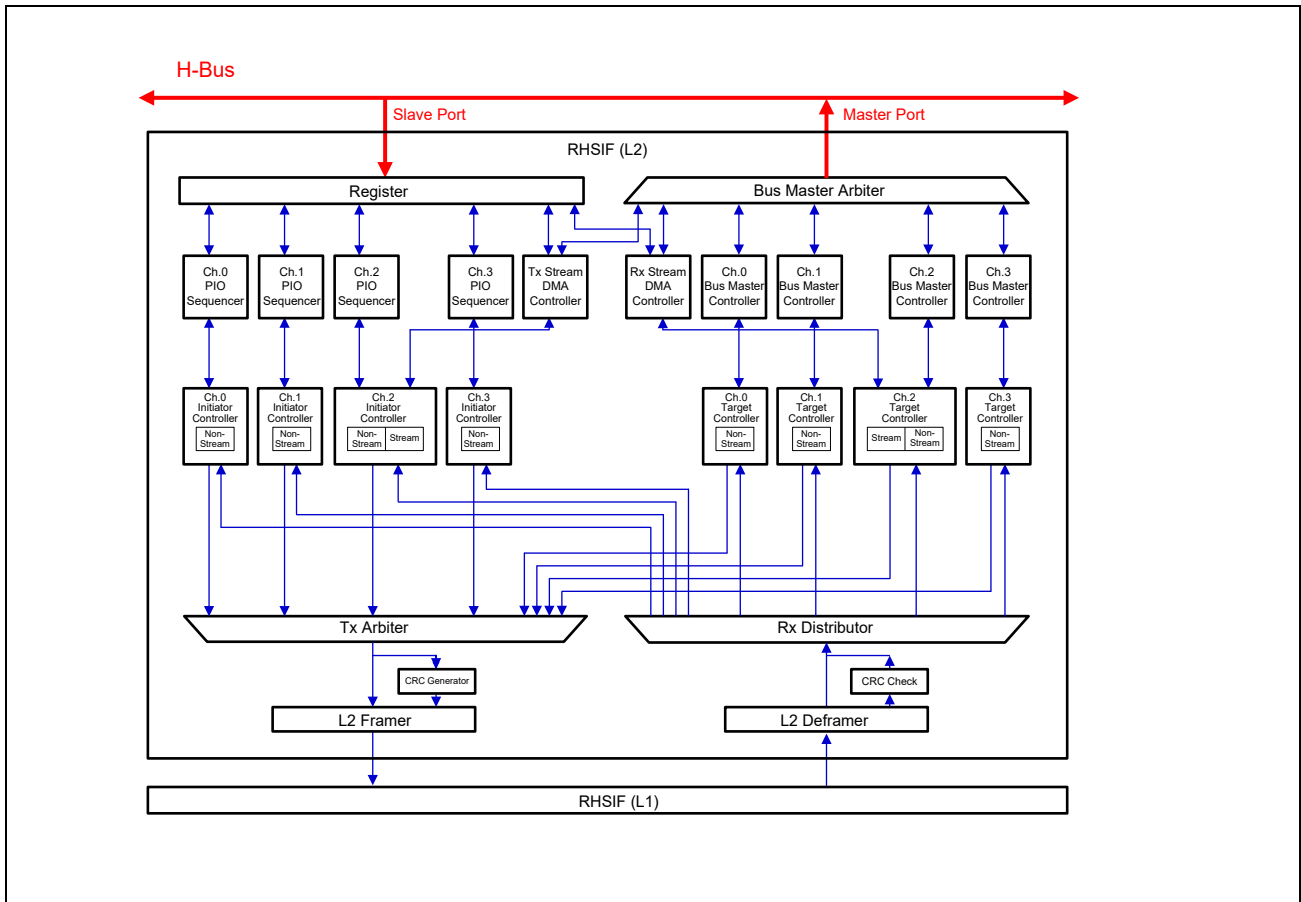


Figure 27.5 Block Diagram of L2

27.4.2 Registers

This section describes the L2 registers in detail.

In case of a concurrent update of the same register bit, L2 module write (hardware write) has higher priority than CPU write (software write).

27.4.2.1 List of Registers

The register map of an L2 module is shown in **Table 27.18**.

The L2 module remains in reset while RESET.

The values shown in **Table 27.18** in the registers description under “Values after Reset” are related to this reset state.

For further information about the L2 operation modes, refer to **Section 27.4.3, Operation**.

The specific base address of a module for an L2 module needs to be added to each of the specified offset addresses.

Table 27.18 Registers Address Map

Module Name	Register Name	Symbol	Address	Access	Access Protection
RHSIFn	Module Mode Register	RHSIFnMMD	<RHSIFn_L2_base> + 000 _H	32, 16, 8	-
	Module Control Register	RHSIFnMCT	<RHSIFn_L2_base> + 008 _H	32, 16, 8	(*1)
	Module Status Register	RHSIFnMST	<RHSIFn_L2_base> + 010 _H	32, 16, 8	-
	Module Interrupt Status Register	RHSIFnMIST	<RHSIFn_L2_base> + 018 _H	32, 16, 8	-
	Module Reply Timeout Time Register	RHSIFnMRT	<RHSIFn_L2_base> + 020 _H	32, 16, 8	-
	Module Memory Window A Start Address Register	RHSIFnMWAA	<RHSIFn_L2_base> + 040 _H	32, 16, 8	(*1)
	Module Memory Window A Size Register	RHSIFnMWAS	<RHSIFn_L2_base> + 048 _H	32, 16, 8	(*1)
	Module Memory Window B Start Address Register	RHSIFnMWBA	<RHSIFn_L2_base> + 050 _H	32, 16, 8	(*1)
	Module Memory Window B Size Register	RHSIFnMWBS	<RHSIFn_L2_base> + 058 _H	32, 16, 8	(*1)
	Module Memory Window C Start Address Register	RHSIFnMWCA	<RHSIFn_L2_base> + 060 _H	32, 16, 8	(*1)
	Module Memory Window C Size Register	RHSIFnMWCS	<RHSIFn_L2_base> + 068 _H	32, 16, 8	(*1)
	Module Memory Window D Start Address Register	RHSIFnMWDA	<RHSIFn_L2_base> + 070 _H	32, 16, 8	(*1)
	Module Memory Window D Size Register	RHSIFnMWDS	<RHSIFn_L2_base> + 078 _H	32, 16, 8	(*1)
	Stream Tx Mode Register	RHSIFnSTMD	<RHSIFn_L2_base> + 200 _H	32, 16, 8	-
	Stream Tx Control Register	RHSIFnSTCT	<RHSIFn_L2_base> + 208 _H	32, 16, 8	-
	Stream Tx Status Register	RHSIFnSTST	<RHSIFn_L2_base> + 210 _H	32, 16, 8	-
	Stream Tx Status Clear Register	RHSIFnSTSC	<RHSIFn_L2_base> + 218 _H	32, 16, 8	-
	Stream Tx Interrupt Enable Register	RHSIFnSTIE	<RHSIFn_L2_base> + 220 _H	32, 16, 8	-
	Stream Tx Source Address Register	RHSIFnSTSA	<RHSIFn_L2_base> + 228 _H	32, 16, 8	-
	Stream Tx Byte Count Register	RHSIFnSTBC	<RHSIFn_L2_base> + 230 _H	32, 16, 8	-
Stream Rx Mode Register	RHSIFnSRMD	<RHSIFn_L2_base> + 280 _H	32, 16, 8	-	
Stream Rx Control Register	RHSIFnSRCT	<RHSIFn_L2_base> + 288 _H	32, 16, 8	-	
Stream Rx Status Register	RHSIFnSRST	<RHSIFn_L2_base> + 290 _H	32, 16, 8	-	

Table 27.18 Registers Address Map

Module Name	Register Name	Symbol	Address	Access	Access Protection
	Stream Rx Status Clear Register	RHSIFnSRSC	<RHSIFn_L2_base> + 298 _H	32, 16, 8	-
	Stream Rx Interrupt Enable Register	RHSIFnSRIE	<RHSIFn_L2_base> + 2A0 _H	32, 16, 8	-
	Stream Rx Destination Area Start Address Register	RHSIFnSRDA	<RHSIFn_L2_base> + 2A8 _H	32, 16, 8	-
	Stream Rx Destination Area Size Register	RHSIFnSRDS	<RHSIFn_L2_base> + 2B0 _H	32, 16, 8	-
	Stream Rx Byte Count Register	RHSIFnSRBC	<RHSIFn_L2_base> + 2B8 _H	32, 16, 8	-
	Stream Rx Write Pointer Register	RHSIFnSRWP	<RHSIFn_L2_base> + 2C0 _H	32, 16, 8	-
	Stream Rx Read Pointer Register	RHSIFnSRRP	<RHSIFn_L2_base> + 2C8 _H	32, 16, 8	-
	Authentication ID Data Register 0	RHSIFnAID0	<RHSIFn_L2_base> + 300 _H	32, 16, 8	(*2)
	Authentication ID Data Register 1	RHSIFnAID1	<RHSIFn_L2_base> + 308 _H	32, 16, 8	(*2)
	Authentication ID Data Register 2	RHSIFnAID2	<RHSIFn_L2_base> + 310 _H	32, 16, 8	(*2)
	Authentication ID Data Register 3	RHSIFnAID3	<RHSIFn_L2_base> + 318 _H	32, 16, 8	(*2)
	Authentication ID Data Register 4	RHSIFnAID4	<RHSIFn_L2_base> + 320 _H	32, 16, 8	(*2)
	Authentication ID Data Register 5	RHSIFnAID5	<RHSIFn_L2_base> + 328 _H	32, 16, 8	(*2)
	Authentication ID Data Register 6	RHSIFnAID6	<RHSIFn_L2_base> + 330 _H	32, 16, 8	(*2)
	Authentication ID Data Register 7	RHSIFnAID7	<RHSIFn_L2_base> + 338 _H	32, 16, 8	(*2)
	Authentication Answer Data Register 0	RHSIFnAAD0	<RHSIFn_L2_base> + 340 _H	32, 16, 8	(*3)
	Authentication Answer Data Register 1	RHSIFnAAD1	<RHSIFn_L2_base> + 348 _H	32, 16, 8	(*3)
	Authentication Answer Data Register 2	RHSIFnAAD2	<RHSIFn_L2_base> + 350 _H	32, 16, 8	(*3)
	Authentication Answer Data Register 3	RHSIFnAAD3	<RHSIFn_L2_base> + 358 _H	32, 16, 8	(*3)
	Authentication Response Data Register 0	RHSIFnARD0	<RHSIFn_L2_base> + 360 _H	32, 16, 8	(*2)
	Authentication Response Data Register 1	RHSIFnARD1	<RHSIFn_L2_base> + 368 _H	32, 16, 8	(*2)
	Authentication Response Data Register 2	RHSIFnARD2	<RHSIFn_L2_base> + 370 _H	32, 16, 8	(*2)
	Authentication Response Data Register 3	RHSIFnARD3	<RHSIFn_L2_base> + 378 _H	32, 16, 8	(*2)
	Authentication Error Status Register	RHSIFnAEST	<RHSIFn_L2_base> + 3E0 _H	32, 16, 8	-
	Authentication Error Status Clear Register	RHSIFnAESC	<RHSIFn_L2_base> + 3E8 _H	32, 16, 8	-
	Authentication Error Interrupt Enable Register	RHSIFnAEIE	<RHSIFn_L2_base> + 3F0 _H	32, 16, 8	-
	Channel 0 Mode Register	RHSIFnCMD0	<RHSIFn_L2_base> + 400 _H	32, 16, 8	-
	Channel 0 Control Register	RHSIFnCCT0	<RHSIFn_L2_base> + 408 _H	32, 16, 8	-
	Channel 0 Status Register	RHSIFnCST0	<RHSIFn_L2_base> + 410 _H	32, 16, 8	-
	Channel 0 Status Clear Register	RHSIFnCSC0	<RHSIFn_L2_base> + 418 _H	32, 16, 8	-
	Channel 0 Interrupt Enable Register	RHSIFnCIE0	<RHSIFn_L2_base> + 420 _H	32, 16, 8	-
	Channel 0 Read/Write Address Register	RHSIFnCAR0	<RHSIFn_L2_base> + 428 _H	32, 16, 8	-
	Channel 0 Write Data Register	RHSIFnCWD0	<RHSIFn_L2_base> + 430 _H	32, 16, 8	-
	Channel 0 Read Data Register	RHSIFnCRD0	<RHSIFn_L2_base> + 438 _H	32, 16, 8	-

Table 27.18 Registers Address Map

Module Name	Register Name	Symbol	Address	Access	Access Protection
	Channel 1 Mode Register	RHSIFnCMD1	<RHSIFn_L2_base> + 480 _H	32, 16, 8	-
	Channel 1 Control Register	RHSIFnCCT1	<RHSIFn_L2_base> + 488 _H	32, 16, 8	-
	Channel 1 Status Register	RHSIFnCST1	<RHSIFn_L2_base> + 490 _H	32, 16, 8	-
	Channel 1 Status Clear Register	RHSIFnCSC1	<RHSIFn_L2_base> + 498 _H	32, 16, 8	-
	Channel 1 Interrupt Enable Register	RHSIFnCIE1	<RHSIFn_L2_base> + 4A0 _H	32, 16, 8	-
	Channel 1 Read/Write Address Register	RHSIFnCAR1	<RHSIFn_L2_base> + 4A8 _H	32, 16, 8	-
	Channel 1 Write Data Register	RHSIFnCWD1	<RHSIFn_L2_base> + 4B0 _H	32, 16, 8	-
	Channel 1 Read Data Register	RHSIFnCRD1	<RHSIFn_L2_base> + 4B8 _H	32, 16, 8	-
	Channel 2 Mode Register	RHSIFnCMD2	<RHSIFn_L2_base> + 500 _H	32, 16, 8	-
	Channel 2 Control Register	RHSIFnCCT2	<RHSIFn_L2_base> + 508 _H	32, 16, 8	-
	Channel 2 Status Register	RHSIFnCST2	<RHSIFn_L2_base> + 510 _H	32, 16, 8	-
	Channel 2 Status Clear Register	RHSIFnCSC2	<RHSIFn_L2_base> + 518 _H	32, 16, 8	-
	Channel 2 Interrupt Enable Register	RHSIFnCIE2	<RHSIFn_L2_base> + 520 _H	32, 16, 8	-
	Channel 2 Read/Write Address Register	RHSIFnCAR2	<RHSIFn_L2_base> + 528 _H	32, 16, 8	-
	Channel 2 Write Data Register	RHSIFnCWD2	<RHSIFn_L2_base> + 530 _H	32, 16, 8	-
	Channel 2 Read Data Register	RHSIFnCRD2	<RHSIFn_L2_base> + 538 _H	32, 16, 8	-
	Channel 3 Mode Register	RHSIFnCMD3	<RHSIFn_L2_base> + 580 _H	32, 16, 8	-
	Channel 3 Control Register	RHSIFnCCT3	<RHSIFn_L2_base> + 588 _H	32, 16, 8	-
	Channel 3 Status Register	RHSIFnCST3	<RHSIFn_L2_base> + 590 _H	32, 16, 8	-
	Channel 3 Status Clear Register	RHSIFnCSC3	<RHSIFn_L2_base> + 598 _H	32, 16, 8	-
	Channel 3 Interrupt Enable Register	RHSIFnCIE3	<RHSIFn_L2_base> + 5A0 _H	32, 16, 8	-
	Channel 3 Read/Write Address Register	RHSIFnCAR3	<RHSIFn_L2_base> + 5A8 _H	32, 16, 8	-
	Channel 3 Write Data Register	RHSIFnCWD3	<RHSIFn_L2_base> + 5B0 _H	32, 16, 8	-
	Channel 3 Read Data Register	RHSIFnCRD3	<RHSIFn_L2_base> + 5B8 _H	32, 16, 8	-

Note: Addresses other than those listed above are reserved. Do not access reserved addresses.

Note 1. Data cannot be written to these bits from the IP's master port. When the RHSIF's master port writes to these bits, these bits have the read-only (R) attribute. These bits can be read from all master devices. Initial value depends on specified setting. For details, refer to each section.

Note 2. Data can be read from or written to these bits only via the master port of RHSIF. When another master device such as a CPU, accesses these bits, these bits have the read-only (R) attribute.

Note 3. Data can be read from or written to these bits only from ICUM. When another master device such as a CPU accesses these bits, these bits have the read-only (R) attribute.

The access size mentioned in **Table 27.18** is required for write access. In addition all registers are readable by 8, 16 and 32 bit access.

The user should not access the registers marked as reserved in **Table 27.18**.

The user should not write bits to a value specified as “invalid” in the description of the registers.

27.4.2.2 Legend

This section explains the module state dependent abbreviations used for the L2 registers description.

Conditions:

- R/W: Bit is readable and writable
- R: Read-only bit; the user cannot write to this bit
- W: Write-only bit; Read value is always “0”

Indexes used for register and register Bit Names

The prefix of the “HSIFn” bit name is omitted.

M: The character “(M)” means “0”, “1”, “2”, “3”, “4”, “5”, “6” or “7”.

N: The character “(N)” means “0”, “1”, “2”, or “3”.

X: The character “(X)” means “A”, “B”, “C”, or “D”.

Reserved bits:

- Always read as 0
- Writing to these bits has no effect (hardware protection)

27.4.2.3 Detailed Register Descriptions

Detailed specifications of each register are described below.

(1) RHSIFnMMD — Module Mode Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CHCT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 27.19 RHSIFnMMD Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CHCT	This bit specifies the type of the channel number code table to be used for L2. Note that this bit can only be changed at the time of initial setting. You cannot change this setting dynamically during system operation. 0: Table II (Ch.0 = 100 _B , Ch.1 = 101 _B , Ch.2 = 110 _B , Ch.3 = 111 _B) 1: Table I (Ch.0 = 000 _B , Ch.1 = 001 _B , Ch.2 = 010 _B , Ch.3 = 011 _B)

(2) RHSIFnMCT — Module Control RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	WDWE	WDRE	—	—	WCWE	WCRE	—	—	WBWE	WBRE	—	—	WAVE	WARE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	*1	*1	R	R	*1	*1	R	R	*1	*1	R	R	*1	*1

Note 1. Data cannot be written to these bits from RHSIF's master port. When RHSIF's master port writes to these bits, these bits have the read-only (R) attribute. These bits can be read from all master devices.

Table 27.20 RHSIFnMCT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 14	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
13	WDWE	This bit specifies whether to enable the write operation for memory window D. 0: Disables writing. 1: Enables writing. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).
12	WDRE	This bit specifies whether to enable the read operation for memory window D. 0: Disables reading. 1: Enables reading. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).
11, 10	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
9	WCWE	This bit specifies whether to enable the write operation for memory window C. 0: Disables writing. 1: Enables writing. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).
8	WCRE	This bit specifies whether to enable the read operation for memory window C. 0: Disables reading. 1: Enables reading. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 27.20 RHSIFnMCT Register Contents (2/2)

Bit Position	Bit Name	Function
5	WBWE	This bit specifies whether to enable the write operation for memory window B. 0: Disables writing. 1: Enables writing. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).
4	WBRE	This bit specifies whether to enable the read operation for memory window B. 0: Disables reading. 1: Enables reading. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	WAVE	This bit specifies whether to enable the write operation for memory window A. 0: Disables writing. 1: Enables writing. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).
0	WARE	This bit specifies whether to enable the read operation for memory window A. 0: Disables reading. 1: Enables reading. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).

(3) RHSIFnMST — Module Status RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AUTS1	AUTS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.21 RHSIFnMST Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	AUTS1	This bit indicates the status of the second authentication of the link partner. 0: Second authentication not completed. 1: Second authentication completed.
0	AUTS0	This bit indicates the status of the first authentication of the link partner. 0: First authentication not completed. 1: First authentication completed.

(4) RHSIFnMIST — Module Interrupt Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DBGS	—	—	AESS	—	—	SRES	STES	CERS3	CERS2	CERS1	CERS0	—	SRCS1	SRCS0	STCS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TERS3	AKRS3	RARS3	—	TERS2	AKRS2	RARS2	—	TERS1	AKRS1	RARS1	—	TERS0	AKRS0	RARS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.22 RHSIFnMIST Register Contents (1/2)

Bit Position	Bit Name	Function
31	DBGS	When set to 1, this bit indicates that there is a debug interrupt factor.
30 to 29	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
28	AESS	When set to 1, this bit indicates that there is an error interrupt factor concerning authentication or window mis-hit.
27 to 26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25	SRES	When set to 1, this bit indicates that there is an error interrupt factor concerning stream reception.
24	STES	When set to 1, this bit indicates that there is an error interrupt factor concerning stream transmission.
23	CERS3	When set to 1, this bit indicates that there is an error interrupt factor concerning channel 3.
22	CERS2	When set to 1, this bit indicates that there is an error interrupt factor concerning channel 2.
21	CERS1	When set to 1, this bit indicates that there is an error interrupt factor concerning channel 1.
20	CERS0	When set to 1, this bit indicates that there is an error interrupt factor concerning channel 0.
19	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
18	SRCS1	When set to 1, this bit indicates that there is a factor for completing stream interrupt reception (of not less than the specified size).
17	SRCS0	When set to 1, this bit indicates that there is a factor for completing stream interrupt reception (of not less than the data size of one frame).
16	STCS	When set to 1, this bit indicates that there is a factor for completing stream interrupt transmission.
15	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
14	TERS3	When set to 1, this bit indicates that there is a factor for receiving an Event Command as a channel 3 interrupt.
13	AKRS3	When set to 1, this bit indicates that there is a factor for receiving an ACK Command as a channel 3 interrupt.
12	RARS3	When set to 1, this bit indicates that there is a factor for receiving a Read Answer Command as a channel 3 interrupt.
11	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
10	TERS2	When set to 1, this bit indicates that there is a factor for receiving an Event Command as a channel 2 interrupt.

Table 27.22 RHSIFnMIST Register Contents (2/2)

Bit Position	Bit Name	Function
9	AKRS2	When set to 1, this bit indicates that there is a factor for receiving an ACK Command as a channel 2 interrupt.
8	RARS2	When set to 1, this bit indicates that there is a factor for receiving a Read Answer Command as a channel 2 interrupt.
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	TERS1	When set to 1, this bit indicates that there is a factor for receiving an Event Command as a channel 1 interrupt.
5	AKRS1	When set to 1, this bit indicates that there is a factor for receiving an ACK Command as a channel 1 interrupt.
4	RARS1	When set to 1, this bit indicates that there is a factor for receiving a Read Answer Command as a channel 1 interrupt.
3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	TERS0	When set to 1, this bit indicates that there is a factor for receiving an Event Command as a channel 0 interrupt.
1	AKRS0	When set to 1, this bit indicates that there is a factor for receiving an ACK Command as a channel 0 interrupt.
0	RARS0	When set to 1, this bit indicates that there is a factor for receiving a Read Answer Command as a channel 0 interrupt.

(5) RHSIFnMRT — Module Reply Timeout Time Register

Value after reset: 0000 F3FF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSCL[3:0]				—	—	RCNT[9:0]									
Value after reset	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.23 RHSIFnMRT Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15 to 12	RSCL[3:0]	These bits specify the reference clock for the reply timer. Note that these bits can be changed only at the time of initial setting. You cannot change the setting dynamically during system operation. 0000: Communication clock (cclk) 0001: Communication clock (cclk) divided by 2 0010: Communication clock (cclk) divided by 4 0011: Communication clock (cclk) divided by 8 0100: Communication clock (cclk) divided by 16 0101: Communication clock (cclk) divided by 32 0110: Communication clock (cclk) divided by 64 0111: Communication clock (cclk) divided by 128 1000: Communication clock (cclk) divided by 256 1001: Communication clock (cclk) divided by 512 1010: Communication clock (cclk) divided by 1024 1011: Communication clock (cclk) divided by 2048 1100: Communication clock (cclk) divided by 4096 1101: Communication clock (cclk) divided by 8192 1110: Communication clock (cclk) divided by 16384 1111: Communication clock (cclk) divided by 32768
11, 10	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	RCNT[9:0]	These bits specify the limit count for the reply timer. If the pulse count of the clock specified by the RSCL bits reaches the value specified in these bits, a reply timeout is determined. When 0 is specified in these bits, the reply timeout does not occur. If, however, no reply Command is detected with 0 specified in these bits, the initiator on the channel is disabled. Therefore, specifying 0 in these bits is prohibited if the system does not have a measure to recover from the status in which the initiator is disabled. Note that these bits can be changed only at the time of initial setting. You cannot change the setting dynamically during system operation.

(6) RHSIFnMW (A, B, C, D) A — Module Memory Window (A, B, C, D) Start Address RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MW(X)A[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MW(X)A[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	R	R

Note 1. This bit is readable/writable (RW) or read-only (R). For details, see the description of the bit.

Table 27.24 RHSIFnMW (A, B, C, D) A Register Contents

Bit Position	Bit Name	Function
31 to 2	MW(X)A	<p>These bits specify the start address of the window that is allowed to be accessed when RHSIF generates an H-Bus read or write request as bus master.</p> <p>Write 0 to all the bits lower than those corresponding to the valid size value set in RHSIFnMW(X)S.</p> <p>Example: When the valid size is 4 kilobytes, set the MW(X)A[11:2] bits to 0000_0000_00b.</p> <p>The MW(X)A[6:2] bits must always be 0.</p> <p>When RHSIFnMCT.W(X)RE = 1 or RHSIFnMCT.W(X)WE = 1, the value of these bits (MW(X)A) must not be changed. For details, see Section 27.4.3.1(1), Setting Memory Windows.</p> <p>Data cannot be written to these bits from RHSIF's master port. When RHSIF's master port writes to these bits, these bits have the read-only (R) attribute. These bits can be read from all master devices.</p>
1, 0	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>

(7) RHSIFnMW (A, B, C, D) S — Module Memory Window (A, B, C, D) Size Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MW(X)S[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MW(X)S[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	R	R

Note 1. This bit is readable/writable (R/W) or read-only (R). For details, see the description of the bit.

Table 27.25 RHSIFnMW (A, B, C, D) S Register Contents

Bit Position	Bit Name	Function
31 to 2	MW(X)S	<p>These bits specify the size of the window that is allowed to be accessed when RHSIF generates an H-Bus read or write request as bus master. Write 1 to all the bits lower than those corresponding to the valid size value, and write 0 to all the bits higher than those corresponding to the valid size value.</p> <p>Examples of settings are as follows:</p> <p>0000_0000_0000_0000_0000_0000_0111_11b: 128 bytes 0000_0000_0000_0000_0000_0000_1111_11b: 256 bytes 0000_0000_0000_0000_0000_0001_1111_11b: 512 bytes ... 0011_1111_1111_1111_1111_1111_1111_11b: 1 GB 0111_1111_1111_1111_1111_1111_1111_11b: 2 GB 1111_1111_1111_1111_1111_1111_1111_11b: 4 GB</p> <p>The MW(X)S[6:2] bits must always be 1.</p> <p>When RHSIFnMCT.W(X)RE = 1 or RHSIFnMCT.W(X)WE = 1, the value of these bits (MW(X)S) must not be changed. For details, see Section 27.4.3.1(1), Setting Memory Windows.</p> <p>Data cannot be written to these bits from RHSIF's master port. When RHSIF's master port writes to these bits, these bits have read-only (R) attribute. These bits can be read from all master devices.</p>
1 to 0	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>

(8) RHSIFnSTMD — Stream Tx Mode Register

Value after reset: 0000 1110_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	STNK	—	—	—	STPS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.26 RHSIFnSTMD Register Contents

Bit Position	Bit Name	Function
31 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	STNK	This bit specifies the operation that is necessary when L2, operating as an initiator node, receives a NACK Command from a link partner in response to a Stream Command the initiator node transmitted. 0: Stops the DMAC. (Same as the operation that is necessary when an error occurs) 1: Continues the DMAC operation. (Same as the operation to be done when an ACK Command is received) Note that this bit can be changed only at the time of initial setting. You cannot change the setting dynamically during system operation.
19 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	STPS	This bit specifies the data payload size of the frame when L2, operating as an initiator node, transmits a Stream Command. Note: Transmission/reception in 128 bits is only available for communication between the E2 series devices with the same STPS values. 0: 256 bits 1: 128 bits Note that this bit can be changed only at the time of initial setting. You cannot change the setting dynamically during system operation.
15 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

(9) RHSIFnSTCT — Stream Tx Control RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STDE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 27.27 RHSIFnSTCT Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	STDE	This bit specifies whether to enable the DMAC. Write 1 to this bit to start the DMAC when L2, operating as an initiator node, transmits a Stream Command. 0: Disables the DMAC. 1: Enables the DMAC. This bit is automatically cleared when the transfer ends (normally or abnormally). For details about abnormal cases, see Section 27.4.3.4, Stream Command Transmission by Initiator Node . During the transfer (while this bit is 1), 0 must not be written to this bit.

NOTE

Note that when using a bit-manipulation instruction for writing to this register, the read-modify-write operation may inadvertently set the STDE bit to 1.

(10) RHSIFnSTST — Stream Tx Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	STE4	—	—	STE3	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	STE2	STE1	—	STE0	—	STC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.28 RHSIFnSTST Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
23	STE4	This bit is set to 1 when a memory window mis-hit occurs and the DMAC stops abnormally (because of the error) after RHSIF, operating as an initiator, has started the DMAC for Stream Command transmission.
22 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	STE3	This bit is set to 1 when a transfer error occurs in the H-Bus and the DMAC stops abnormally (because of the error) after L2, operating as an initiator, has started the DMAC for Stream Command transmission.
19 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	STE2	This bit is set to 1 when a reply Command (ACK, NACK, or Read Answer) causing a transaction ID error is detected and the DMAC stops abnormally (because of the error) after L2, operating as an initiator, has started the DMAC for Stream Command transmission.
4	STE1	This bit is set to 1 when the DMAC stops abnormally because it does not receive any response within the time set in the RHSIFnMRT bits after L2, operating as an initiator, has started the DMAC for Stream Command transmission.
3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	STE0	This bit is set to 1 when an NACK Command is received and the DMAC stops abnormally after L2, operating as an initiator, has started the DMAC for Stream Command transmission. This bit is not set when RHSIFnSTMD.STNK = 1.
1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	STC	This bit is set to 1 when the transfer of all data ends normally after L2, operating as an initiator, has started the DMAC for Stream Command transmission.

(11) RHSIFnSTSC — Stream Tx Status Clear RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	STEC4	—	—	STEC3	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	R	R	W	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	STEC2	STEC1	—	STEC0	—	STCC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	W	W	R	W	R	W

Table 27.29 RHSIFnSTSC Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
23	STEC4	Writing 1 to this bit clears the RHSIFnSTST.STE4 bit. Writing 0 to this bit is ignored.
22 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	STEC3	Writing 1 to this bit clears the RHSIFnSTST.STE3 bit. Writing 0 to this bit is ignored.
19 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	STEC2	Writing 1 to this bit clears the RHSIFnSTST.STE2 bit. Writing 0 to this bit is ignored.
4	STEC1	Writing 1 to this bit clears the RHSIFnSTST.STE1 bit. Writing 0 to this bit is ignored.
3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	STEC0	Writing 1 to this bit clears the RHSIFnSTST.STE0 bit. Writing 0 to this bit is ignored.
1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	STCC	Writing 1 to this bit clears the RHSIFnSTST.STC bit. Writing 0 to this bit is ignored.

(12) RHSIFnSTIE — Stream Tx Interrupt Enable Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	STEE4	—	—	STEE3	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	STEE2	STEE1	—	STEE0	—	STCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R	R/W

Table 27.30 RHSIFnSTIE Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
23	STEE4	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSTST.STE4 bit. 0: Does not assert int_hsif_err when the RHSIFnSTST.STE4 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSTST.STE4 bit is 1.
22 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	STEE3	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSTST.STE3 bit. 0: Does not assert int_hsif_err when the RHSIFnSTST.STE3 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSTST.STE3 bit is 1.
19 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	STEE2	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSTST.STE2 bit. 0: Does not assert int_hsif_err when the RHSIFnSTST.STE2 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSTST.STE2 bit is 1.
4	STEE1	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSTST.STE1 bit. 0: Does not assert int_hsif_err when the RHSIFnSTST.STE1 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSTST.STE1 bit is 1.
3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	STEE0	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSTST.STE0 bit. 0: Does not assert int_hsif_err when the RHSIFnSTST.STE0 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSTST.STE0 bit is 1.
1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	STCE	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSTST.STC bit. 0: Does not assert int_hsif_str when the RHSIFnSTST.STC bit is 1. 1: Asserts int_hsif_str when the RHSIFnSTST.STC bit is 1.

(13) RHSIFnSTSA — Stream Tx Source Address RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STSA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STSA[15:4]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 27.31 RHSIFnSTSA Register Contents

Bit Position	Bit Name	Function
31 to 4	STSA	These bits specify the start address of the area storing the data to be transferred when L2, operating as an initiator node, starts the DMAC for Stream Command transmission. When the data payload of the Stream Command to be transmitted is 256 bits (when RHSIFnSTMD.STPS = 0), the STSA[4] bit must always be 0. When RHSIFnSTCT.STDE = 1, the value of these bits (STSA) must not be changed. The settings for RHSIFnSTSA and RHSIFnSTBC which causes wrap around of address is not allowed.
3 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

(14) RHSIFnSTBC — Stream Tx Byte Count RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	STBC[24:16]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STBC[15:4]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Table 27.32 RHSIFnSTBC Register Contents

Bit Position	Bit Name	Function
31 to 25	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
24 to 4	STBC	These bits specify the total number of bytes of the data to be transferred when L2, operating as an initiator node, starts the DMAC for Stream Command transmission. When the data payload of the Stream Command to be transmitted is 256 bits (when RHSIFnSTMD.STPS = 0), the STBC[4] bit must always be 0. When 0 is set in these bits, 2 ²⁵ bytes (32 MBytes) are transferred. The value read from these bits indicates the number of remaining bytes of the data. When RHSIFnSTCT.STDE = 1, the value of these bits (STBC) must not be changed. The setting of RHSIFnSTSA and RHSIFnSTBC that causes wrap around of address is not allowed.
3 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

CAUTION: Write to unused Channel Register should not be executed**NOTE**

Note that when using a bit-manipulation instruction for writing to this register, the read-modify-write operation may inadvertently set the STBC bits to 1.

(15) RHSIFnSRMD — Stream Rx Mode RegisterValue after reset: 0000 0010_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	SRMC	—	—	—	SRPS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.33 RHSIFnSRMD Register Contents

Bit Position	Bit Name	Function
31 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	SRMC	This bit specifies the operation to be done if an error occurs when L2, operating as a target node, receives a Stream Command. 0: Automatically clears the RHSIFnSRCT.SRDE bit automatically when an error occurs. 1: Does not automatically clear the RHSIFnSRCT.SRDE bit automatically when an error occurs. For the conditions of errors, see the description of the RHSIFnSRCT.SRDE bit. Note that this bit can be changed only at the time of initial setting. You cannot change the setting dynamically during system operation.
19 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	SRPS	This bit specifies the frame data payload size that is expected when L2, operating as a target node, receives a Stream Command. 0: 256 bits 1: 128 bits Note that this bit can be changed only at the time of initial setting. You cannot change the setting dynamically during system operation.
15 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

(16) RHSIFnSRCT — Stream Rx Control RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SRDE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 27.34 RHSIFnSRCT Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SRDE	<p>This bit specifies whether to enable the DMAC. Write 1 to this bit to start the DMAC when L2, operating as a target node, receives a Stream Command.</p> <p>0: Disables the DMAC (ignores the Stream Command that is received without any response).</p> <p>1: Enables the DMAC (automatically transfers the Stream Command that is received to a storage area).</p> <p>When RHSIFnCMD2.TNME2 = 0, you must not set this bit to 1.</p> <p>When RHSIFnSRMD.SRMC = 0, this bit is cleared to 0 automatically when one of the following events occur:</p> <ul style="list-style-type: none"> • An error occurs when Stream Command data is transferred to the H-Bus. • When a Stream Command is received, Stream Command data is discarded because the data storage area is full. • An error is detected in a Stream Command that is received. • In channel 2, a Command causing a CRC error is detected. • H-Bus Write request's address does not hit any memory window.

NOTE

Note that when using a bit-manipulation instruction for writing to this register, the read-modify-write operation may inadvertently set the SRDE bit to 1.

(17) RHSIFnSRST — Stream Rx Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SRA	—	—	—	—	—	—	—	SRE2	—	SRE1	SRE0	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRTA	—	—	—	—	—	—	—	—	—	—	—	—	—	SRC1	SRC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.35 RHSIFnSRST Register Contents

Bit Position	Bit Name	Function
31	SRA	This bit is set to 1 if RHSIFnSRCT.SRDE is cleared automatically because L2, operating as a target node, detects errors related to the Stream Command that is received. For the conditions of errors, see the description of the RHSIFnSRCT.SRDE bit.
30 to 24	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
23	SRE2	This bit is set to 1 if a memory window mis-hit occurs when RHSIF, operating as a target node, transfers Stream Command data that is received to the H-Bus.
22	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
21	SRE1	This bit is set to 1 if Stream Command data is discarded because the data storage area is full when L2, operating as a target node, receives a Stream Command.
20	SRE0	This bit is set to 1 if an error occurs when L2, operating as a target node, transfers Stream Command data that is received to the H-Bus.
19 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15	SRTA	This bit indicates the processing status of the received Stream command when L2 is operating as the target node. 0: No Command is being processed. 1: Command(s) is being processed.
14 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SRC1	This bit is set to 1 when the size of unprocessed data (in the received Stream Command L2 is operating as a target node) is not less than the value set in RHSIFnSRBC.
0	SRC0	This bit is set to 1 when the size of unprocessed data (in the received Stream Command L2 is operating as a target node) is not 0.

(18) RHSIFnSRSC — Stream Rx Status Clear RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SRAC	—	—	—	—	—	—	—	SREC2	—	SREC1	SREC0	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	R	R	R	R	R	R	R	W	R	W	W	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.36 RHSIFnSRSC Register Contents

Bit Position	Bit Name	Function
31	SRAC	Writing 1 to this bit clears the RHSIFnSRST.SRA bit. Writing 0 to this bit is ignored.
30 to 24	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
23	SREC2	Writing 1 to this bit clears the RHSIFnSRST.SRE2 bit. Writing 0 to this bit is ignored.
22	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
21	SREC1	Writing 1 to this bit clears the RHSIFnSRST.SRE1 bit. Writing 0 to this bit is ignored.
20	SREC0	Writing 1 to this bit clears the RHSIFnSRST.SRE0 bit. Writing 0 to this bit is ignored.
19 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

(19) RHSIFnSRIE — Stream Rx Interrupt Enable RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SRAE	—	—	—	—	—	—	—	SREE2	—	SREE1	SREE0	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SRCE1	SRCE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 27.37 RHSIFnSRIE Register Contents

Bit Position	Bit Name	Function
31	SRAE	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSRST.SRA bit. 0: Does not assert int_hsif_err when the RHSIFnSRST.SRA bit is 1. 1: Asserts int_hsif_err when the RHSIFnSRST.SRA bit is 1.
30 to 24	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
23	SREE2	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSRST.SRE2 bit. 0: Does not assert int_hsif_err when the RHSIFnSRST.SRE2 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSRST.SRE2 bit is 1.
22	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
21	SREE1	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSRST.SRE1 bit. 0: Does not assert int_hsif_err when the RHSIFnSRST.SRE1 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSRST.SRE1 bit is 1.
20	SREE0	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSRST.SRE0 bit. 0: Does not assert int_hsif_err when the RHSIFnSRST.SRE0 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSRST.SRE0 bit is 1.
19 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SRCE1	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSRST.SRC1 bit. 0: Does not assert int_hsif_str when the RHSIFnSRST.SRC1 bit is 1. 1: Asserts int_hsif_str when the RHSIFnSRST.SRC1 bit is 1.
0	SRCE0	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSRST.SRC0 bit. 0: Does not assert int_hsif_str when the RHSIFnSRST.SRC0 bit is 1. 1: Asserts int_hsif_str when the RHSIFnSRST.SRC0 bit is 1.

(20) RHSIFnSRDA — Stream Rx Destination Area Start Address Register

Value after reset: 0000 0000_H

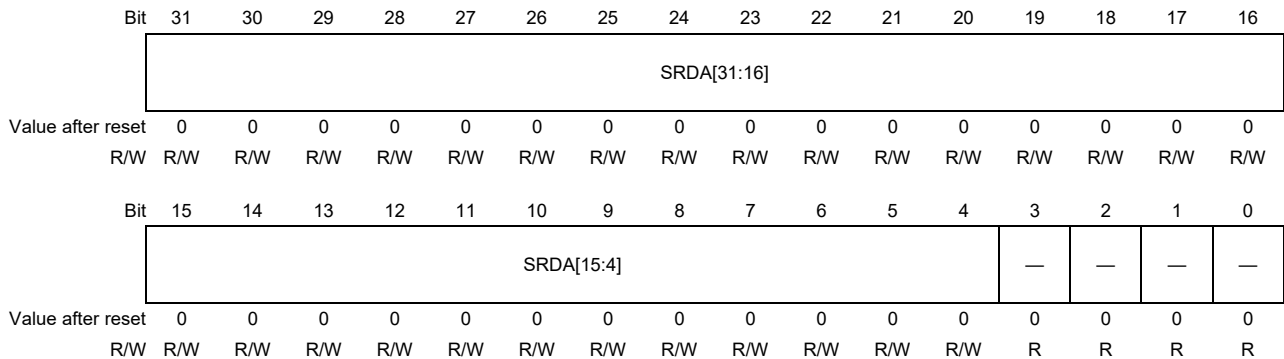


Table 27.38 RHSIFnSRDA Register Contents

Bit Position	Bit Name	Function
31 to 4	SRDA	These bits specify the start address of the area to store data when L2, operating as a target node, receives a Stream Command. Write 0 to all the bits lower than those corresponding to the valid size value set in RHSIFnSRDS. Example: When the valid size is 4 kilobytes, set the SRDA[11:4] bits to 0000_0000_00b. When the data payload of the Stream Command to be received is 256 bits (when RHSIFnSRMD.SRPS = 0), the SRDA[4] bit must always be 0. When RHSIFnSRCT.SRDE = 1, the value of these bits (SRDA) must not be changed.
3 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

(21) RHSIFnSRDS — Stream Rx Destination Area Size RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	—	—	—	—	—	—	—	SRDS[24:16]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	SRDS[15:4]												—	—	—	—		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	

Table 27.39 RHSIFnSRDS Register Contents

Bit Position	Bit Name	Function
31 to 25	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
24 to 4	SRDS	These bits specify the valid size of the area to store data when L2, operating as a target node, receives a Stream Command. Write 1 to all the bits lower than those corresponding to the valid size value, and write 0 to all the bits higher than those corresponding to the valid size value. Examples of settings are as follows: 0_0000_0000_0000_0000_0000b: 16 bytes 0_0000_0000_0000_0000_0001b: 32 bytes ... 0_1111_1111_1111_1111_1111b: 16 Mbytes 1_1111_1111_1111_1111_1111b: 32 Mbytes When the data payload of the Stream Command to be received is 256 bits (when RHSIFnSRMD.SRPS = 0), the valid size must not be set to 16 bytes. When RHSIFnSRCT.SRDE = 1, the value of these bits (SRDS) must not be changed.
3 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

(22) RHSIFnSRBC — Stream Rx Byte Count RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	—	—	—	—	SRBC[24:16]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	SRBC[15:4]												—	—	—	—	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 27.40 RHSIFnSRBC Register Contents

Bit Position	Bit Name	Function
31 to 25	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
24 to 4	SRBC	These bits specify the number of bytes that triggers a notification when L2, operating as a target node, receives a Stream Command. When the size of unprocessed data reaches or exceeds the value specified in these bits, the RHSIFnSRST.SRC1 bit is set to 1. If the data payload of the Stream Command to be received is 256 bits (when RHSIFnSRMD.SRPS = 0), the SRBC[4] bit must always be 0. If 0 is set in these bits, a notification is triggered when the size of unprocessed data reaches 2 ²⁵ bytes (32 Mbytes). When RHSIFnSRCT.SRDE = 1, the value of these bits (SRBC) must not be changed.
3 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

(23) RHSIFnSRWP — Stream Rx Write Pointer RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	SRWP[24:16]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRWP[15:4]												—	—	—	SRWT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.41 RHSIFnSRWP Register Contents

Bit Position	Bit Name	Function
31 to 25	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
24 to 4	SRWP	These bits indicate the value of the write pointer (offset value from start address) to the area to store received data when L2, operating as a target node, receives a Stream Command. The number of valid bits of the pointer corresponds to the value set in RHSIFnSRDS. All the bits higher than valid bits show 0. These bits are initialized when the RHSIFnSRCT.SRDE bit changes from 0 to 1. For details, see Section 27.4.3.5, Stream Command Reception by Target Node .
3 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SRWT	This bit is a toggle bit for the SRWP bits. This bit is toggled each time the SRWP bits wrap around. This bit is initialized when the RHSIFnSRCT.SRDE bit changes from 0 to 1.

(24) RHSIFnSRRP — Stream Rx Read Pointer RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	—	—	—	—	SRRP[24:16]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	SRRP[15:4]												—	—	—	SRRT	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W

Table 27.42 RHSIFnSRRP Register Contents

Bit Position	Bit Name	Function
31 to 25	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
24 to 4	SRRP	These bits indicate the value of the read pointer (offset value from start address) to the area to store received data when L2, operating as a target node, receives a Stream Command. The number of valid bits of the pointer corresponds to the value set in RHSIFnSRDS. All the bits higher than valid bits must be 0. These bits are initialized when the RHSIFnSRCT.SRDE bit changes from 0 to 1. For details, see Section 27.4.3.5, Stream Command Reception by Target Node .
3 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SRRT	This bit is a toggle bit for the SRRP bits. This bit is toggled each time the SRRP bits wrap around. This bit is initialized when the RHSIFnSRCT.SRDE bit changes from 0 to 1.

NOTE

Note that when using a bit-manipulation instruction for writing to this register, the read-modify-write operation may inadvertently set the SRRP bits or the SRRT bit to 1.

(25) RHSIFnAID (0, 1, 2, 3, 4, 5, 6, 7) — Authentication ID Data Register (0, 1, 2, 3, 4, 5, 6, 7)Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AID(M)[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AID(M)[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Note 1. This bit is readable/writable (R/W) or read-only (R). For details, see the description of the bit.

Table 27.43 RHSIFnAID (0, 1, 2, 3, 4, 5, 6, 7) Register Contents

Bit Position	Bit Name	Function
31 to 0	AID(M)	<p>These bits store the ID data for the first authentication (ID-based authentication).</p> <p>HSIFnAID0.AID0[31:0]: ID[31:0] HSIFnAID1.AID1[31:0]: ID[63:32] HSIFnAID2.AID2[31:0]: ID[95:64] HSIFnAID3.AID3[31:0]: ID[127:96] HSIFnAID4.AID4[31:0]: ID[159:128] HSIFnAID5.AID5[31:0]: ID[191:160] HSIFnAID6.AID6[31:0]: ID[223:192] HSIFnAID7.AID7[31:0]: ID[255:224]</p> <p>Data can be read from or written to these bits only via the master port of L2. When another master device such as a CPU accesses these bits, these bits have the read-only (R) attribute.</p>

(26) RHSIFnAAD (0, 1, 2, 3) — Authentication Answer Data Register (0, 1, 2, 3)Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AAD(N)[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AAD(N)[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Note 1. This bit is readable/writable (R/W) or read-only (R). For details, see the description of the bit.

Table 27.44 RHSIFnAAD (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31 to 0	AAD(N)	<p>These bits store the answer data (calculated from challenge data by the ICUM) for the second authentication (challenge-and-response authentication).</p> <p>RHSIFnAAD0.AAD0[31:0]: Answer Data[31:0] RHSIFnAAD1.AAD1[31:0]: Answer Data[63:32] RHSIFnAAD2.AAD2[31:0]: Answer Data[95:64] RHSIFnAAD3.AAD3[31:0]: Answer Data[127:96]</p> <p>Data can be read from or written to these bits only from ICUM. When another master device such as a CPU accesses these bits, these bits have the read-only (R) attribute.</p>

(27) RHSIFnARD (0, 1, 2, 3) — Authentication Response Data Register (0, 1, 2, 3)Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ARD(N)[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ARD(N)[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Note 1. This bit is readable/writable (R/W) or read-only (R). For details, see the description of the bit.

Table 27.45 RHSIFnARD (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31 to 0	ARD(N)	<p>These bits store the response data for the second authentication (challenge-and-response authentication).</p> <p>RHSIFnARD0.ARD0[31:0]: Response Data[31:0] RHSIFnARD1.ARD1[31:0]: Response Data[63:32] RHSIFnARD2.ARD2[31:0]: Response Data[95:64] RHSIFnARD3.ARD3[31:0]: Response Data[127:96]</p> <p>Data can be read from or written to these bits only via the master port of L2. When another master device such as a CPU accesses these bits, these bits have the read-only (R) attribute.</p>

(28) RHSIFnAEST — Authentication Error Status RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	WES1	WES0	WEW3	WER3	WEW2	WER2	WEW1	WER1	WEW0	WER0	—	—	—	AES
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEW3	AER3	AEI3	AEE3	AEW2	AER2	AEI2	AEE2	AEW1	AER1	AEI1	AEE1	AEW0	AER0	AEI0	AEE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.46 RHSIFnAEST Register Contents

Bit Position	Bit Name	Function
31 to 30	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
29	WES1	This bit is set to 1 when RHSIF, as a target node, tries to issue H-Bus Write for Stream Command that does not hit any memory window.
28	WES0	This bit is set to 1 when RHSIF, as an initiator node, tries to issue H-Bus Read for Stream Command that does not hit any memory window.
27,25,23,21	WEW(N)	This bit is set to 1 when RHSIF, as a target node, receives Write Command on Channel (N) whose targeted address does not hit any memory window.
26,24,22,20	WER(N)	This bit is set to 1 when RHSIF, as a target node, receives Read Command on Channel (N) whose targeted address does not hit any memory window.
19 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	AES	This bit is set to 1 when RHSIF, as a target node, receives Stream Command on Channel (N) before authentication is completed.
15,11,7,3	AEW(N)	This bit is set to 1 when RHSIF, as a target node, receives Write Command on Channel (N) before authentication is completed.
14,10,6,2	AER(N)	This bit is set to 1 when RHSIF, as a target node, receives Read Command on Channel (N) before authentication is completed.
13,9,5,1	AEI(N)	This bit is set to 1 when RHSIF, as a target node, receives ID Command on Channel (N) before authentication is completed.
12,8,4,0	AEE(N)	This bit is set to 1 when RHSIF, as a target node, receives Event Command on Channel (N) before authentication is completed.

(29) RHSIFnAESC — Authentication Error Status Clear RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	WESC1	WESC0	WEWC ₃	WERC3	WEWC ₂	WERC2	WEWC ₁	WERC1	WEWC ₀	WERC0	—	—	—	AESC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	W	W	W	W	W	W	W	W	W	W	R	R	R	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEWC3	AERC3	AEIC3	AEEC3	AEWC2	AERC2	AEIC2	AEEC2	AEWC1	AERC1	AEIC1	AEEC1	AEWC0	AERC0	AEIC0	AEEC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 27.47 RHSIFnAESC Register Contents

Bit Position	Bit Name	Function
31 to 30	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
29	WESC1	Writing 1 to this bit clears the RHSIFnAESC.WES1 bit. Writing 0 to this bit is ignored.
28	WESC0	Writing 1 to this bit clears the RHSIFnAESC.WES0 bit. Writing 0 to this bit is ignored.
27,25,23,21	WEWC(N)	Writing 1 to this bit clears the RHSIFnAESC.WEW(N) bit. Writing 0 to this bit is ignored.
26,24,22,20	WERC(N)	Writing 1 to this bit clears the RHSIFnAESC.WER(N) bit. Writing 0 to this bit is ignored.
19 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	AESC	Writing 1 to this bit clears the RHSIFnAESC.AES bit. Writing 0 to this bit is ignored.
15,11,7,3	AEWC(N)	Writing 1 to this bit clears the RHSIFnAESC.AEW(N) bit. Writing 0 to this bit is ignored.
14,10,6,2	AERC(N)	Writing 1 to this bit clears the RHSIFnAESC.AER(N) bit. Writing 0 to this bit is ignored.
13,9,5,1	AEIC(N)	Writing 1 to this bit clears the RHSIFnAESC.AEI(N) bit. Writing 0 to this bit is ignored.
12,8,4,0	AEEC(N)	Writing 1 to this bit clears the RHSIFnAESC.AEE(N) bit. Writing 0 to this bit is ignored.

(30) RHSIFnAEIE — Authentication Error Interrupt Enable RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	WESE1	WESE0	WEWE ₃	WERE3	WEWE ₂	WERE2	WEWE ₁	WERE1	WEWE ₀	WERE0	—	—	—	AESE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AWEWE3	AERE3	AEIE3	AEEE3	AWEWE2	AERE2	AEIE2	AEEE2	AWEWE1	AERE1	AEIE1	AEEE1	AWEWE0	AERE0	AEIE0	AEEE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.48 RHSIFnAESC Register Contents

Bit Position	Bit Name	Function
31 to 30	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
29	WESE1	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.WES1 bit. 0: Does not assert int_hsif_sec when the RSIFnAEST.WES1 bit is 1. 1: Asserts int_hsif_sec when the RSIFnAEST.WES1 bit is 1.
28	WESE0	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.WES0 bit. 0: Does not assert int_hsif_sec when the RSIFnAEST.WES0 bit is 1. 1: Asserts int_hsif_sec when the RSIFnAEST.WES0 bit is 1.
27,25,23,21	WEWE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.WEW(N) bit. 0: Does not assert int_hsif_sec when the RSIFnAEST.WEW(N) bit is 1. 1: Asserts int_hsif_sec when the RSIFnAEST.WEW(N) bit is 1.
26,24,22,20	WERE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.WER(N) bit. 0: Does not assert int_hsif_sec when the RSIFnAEST.WER(N) bit is 1. 1: Asserts int_hsif_sec when the RSIFnAEST.WER(N) bit is 1.
19 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15,11,7,3	AWEWE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.AEW(N) bit. 0: Does not assert int_hsif_sec when the RSIFnAEST.AEW(N) bit is 1. 1: Asserts int_hsif_sec when the RSIFnAEST.AEW(N) bit is 1.
14,10,6,2	AEREx(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.AER(N) bit. 0: Does not assert int_hsif_sec when the RSIFnAEST.AER(N) bit is 1. 1: Asserts int_hsif_sec when the RSIFnAEST.AER(N) bit is 1.
13,9,5,1	AEIE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.AEI(N) bit. 0: Does not assert int_hsif_sec when the RSIFnAEST.AEI(N) bit is 1. 1: Asserts int_hsif_sec when the RSIFnAEST.AEI(N) bit is 1.
12,8,4,0	AEEE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.AEE(N) bit. 0: Does not assert int_hsif_sec when the RSIFnAEST.AEE(N) bit is 1. 1: Asserts int_hsif_sec when the RSIFnAEST.AEE(N) bit is 1.

(31) RHSIFnCMD (0, 1, 2, 3) — Channel (0, 1, 2, 3) Mode RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TNME (N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INME (N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 27.49 RHSIFnCMD (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	TNME(N)	This bit specifies whether to enable the target node function. 0: Disables reception of Read, Write, Event, ID, and Stream Commands (ignores these Commands without any response). 1: Enables reception of Read, Write, Event, ID, and Stream Commands. Even when this bit is 1, whether Read, Write, and Stream Commands can be received depends on other individual settings. For details, see Sections 27.4.3.3, Non-Stream Command Reception by Target Node and Section 27.4.3.5, Stream Command Reception by Target Node .
15 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	INME(N)	This bit specifies whether to enable the initiator node function. 0: Disables transmission of Read, Write, Event, ID, and Stream Commands. 1: Enables transmission of Read, Write, Event, ID, and Stream Commands. This bit must not be cleared to 0 while L2 is transmitting data or waiting for a response by using the initiator node function.

(32) RHSIFnCCCT (0, 1, 2, 3) — Channel (0, 1, 2, 3) Control RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CTYW(N)	—	—	CTY(N)[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	R	R	W	W	W	W	W

Table 27.50 RHSIFnCCCT (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	CTYW(N)	“1” must be written to this bit at the same time as writing data to the CTY(N) bits when L2, operating as an initiator node, transmits a request Command other than a Stream Command. When this bit is 0, the data written to the CTY(N) bits is ignored.
6, 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	CTY(N)	A Command type must be set in these bits when L2, operating as an initiator node, transmits a request Command other than a Stream Command. When L2 transmits a Command, 1 must be written to the CTYW(N) bit at the same time as writing data to these bits. 00000: Read Command (8 bits) 00001: Read Command (16 bits) 00010: Read Command (32 bits) 00100: Write Command (8 bits) 00101: Write Command (16 bits) 00110: Write Command (32 bits) 01100: Event Command 10010: ID Command Only the above Command types can be set.

(33) RHSIFnCST (0, 1, 2, 3) — Channel (0, 1, 2, 3) Status RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRE(N)	—	—	—	—	—	—	—	—	—	—	BRE(N)	—	—	—	TER(N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDY(N)	—	—	—	—	—	—	—	AOE(N)	—	IDE(N)	TOE(N)	—	AKE(N)	AKR(N)	RAR(N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.51 RHSIFnCST (0, 1, 2, 3) Register Contents (1/2)

Bit Position	Bit Name	Function
31	CRE(N)	This bit is set to 1 when a Command causing a CRC error is detected in the channel corresponding to the bit number. For details, see Section 27.4.3.6, CRC Error Handling .
30 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	BRE(N)	This bit is set to 1 when an error has occurred during data transfer in the H-Bus by a Read or Write Command received by L2 operating as a target node.
19 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	TER(N)	This bit is set to 1 when L2, operating as a target mode, receives an Event Command. Note that L2 ignores an event if another Event Command is received when this bit is already set to 1.
15	RDY(N)	This bit indicates whether L2, operating as an initiator node, can transmit request Commands other than Stream Commands. 0: Request Commands cannot be transmitted (the initiator node function is disabled or L2 is waiting for a reply Command). 1: Request Commands can be transmitted. Transmission of request Commands (writing 1 to the RHSIFnCCT(N).CTYW(N) bit) must be done only when this bit is 1.
14 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 27.51 RHSIFnCST (0, 1, 2, 3) Register Contents (2/2)

Bit Position	Bit Name	Function
7	AOE(N)	This bit is set to 1 when an error that cannot be classified as the one indicated by the AKE or IDE bit is detected after L2, operating as an initiator node, has transmitted a Read, Write, Event, or ID Command. In particular, the following event corresponds to this error: <ul style="list-style-type: none"> • A Read Command transmitted by RHSIF has met the “Any Other” condition in <i>Table 18 of the HSSL / SIPI Interprocessor Bus Protocol Specification</i>.
6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	IDE(N)	This bit is set to 1 when a reply Command (ACK, NACK, or Read Answer) causing a transaction ID error is detected after L2, operating as an initiator node, has transmitted a Read, Write, Event or ID Command.
4	TOE(N)	This bit is set to 1 when L2 cannot receive a response within the time set in RHSIFnMRT after L2, operating as an initiator node, has transmitted a Read, Write, Event, or ID Command.
3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	AKE(N)	This bit is set to 1 when L2 receives a NACK Command after L2, operating as an initiator node, has transmitted a Read, Write, Event, or ID Command.
1	AKR(N)	This bit is set to 1 when L2 receives an ACK Command after L2, operating as an initiator node, has transmitted a Write or Event Command.
0	RAR(N)	This bit is set to 1 when L2 receives a Read Answer Command after L2, operating as an initiator node, has transmitted a Read or ID Command.

(34) RHSIFnCSC (0, 1, 2, 3) — Channel (0, 1, 2, 3) Status Clear RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CREC(N)	—	—	—	—	—	—	—	—	—	—	BREC(N)	—	—	—	TERC(N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	R	R	R	R	R	R	R	R	R	R	W	R	R	R	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	AOEC(N)	—	IDEC(N)	TOEC(N)	—	AKEC(N)	AKRC(N)	RARC(N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	R	W	W	R	W	W	W

Table 27.52 RHSIFnCSC (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31	CREC(N)	Writing 1 to this bit clears the RHSIFnCST(N).CRE(N) bit. Writing 0 to this bit is ignored.
30 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	BREC(N)	Writing 1 to this bit clears the RHSIFnCST(N).BRE(N) bit. Writing 0 to this bit is ignored.
19 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	TERC(N)	Writing 1 to this bit clears the RHSIFnCST(N).TER(N) bit. Writing 0 to this bit is ignored.
15 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	AOEC(N)	Writing 1 to this bit clears the RHSIFnCST(N).AOE(N) bit. Writing 0 to this bit is ignored.
6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	IDEC(N)	Writing 1 to this bit clears the RHSIFnCST(N).IDE(N) bit.
4	TOEC(N)	Writing 1 to this bit clears the RHSIFnCST(N).TOE(N) bit. Writing 0 to this bit is ignored.
3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	AKEC(N)	Writing 1 to this bit clears the RHSIFnCST(N).AKE(N) bit. Writing 0 to this bit is ignored.
1	AKRC(N)	Writing 1 to this bit clears the RHSIFnCST(N).AKR(N) bit. Writing 0 to this bit is ignored.
0	RARC(N)	Writing 1 to this bit clears the RHSIFnCST(N).RAR(N) bit. Writing 0 to this bit is ignored.

(35) RHSIFnCIE (0, 1, 2, 3) — Channel (0, 1, 2, 3) Interrupt Enable RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CREE(N)	—	—	—	—	—	—	—	—	—	—	BREE(N)	—	—	—	TERE(N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	AOEE(N)	—	IDEE(N)	TOEE(N)	—	AKEE(N)	AKRE(N)	RARE(N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R/W	R/W	R/W

Table 27.53 RHSIFnCIE (0, 1, 2, 3) Register Contents (1/2)

Bit Position	Bit Name	Function
31	CREE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).CRE(N) bit. 0: Does not assert int_hsif_err when the RHSIFnCST(N).CRE(N) bit is 1. 1: Asserts int_hsif_err when the RHSIFnCST(N).CRE(N) bit is 1.
30 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	BREE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).BRE(N) bit. 0: Does not assert int_hsif_err when the RHSIFnCST(N).BRE(N) bit is 1. 1: Asserts int_hsif_err when the RHSIFnCST(N).BRE(N) bit is 1.
19 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	TERE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).TER(N) bit. 0: Does not assert int_hsif_ch(N) when the RHSIFnCST(N).TER(N) bit is 1. 1: Asserts int_hsif_ch(N) when the RHSIFnCST(N).TER(N) bit is 1.
15 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	AOEE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).AOE(N) bit. 0: Does not assert int_hsif_err when the RHSIFnCST(N).AOE(N) bit is 1. 1: Asserts int_hsif_err when the RHSIFnCST(N).AOE(N) bit is 1.
6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	IDEE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).IDE(N) bit. 0: Does not assert int_hsif_err when the RHSIFnCST(N).IDE(N) bit is 1. 1: Asserts int_hsif_err when the RHSIFnCST(N).IDE(N) bit is 1.
4	TOEE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).TOE(N) bit. 0: Does not assert int_hsif_err when the RHSIFnCST(N).TOE(N) bit is 1. 1: Asserts int_hsif_err when the RHSIFnCST(N).TOE(N) bit is 1.
3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 27.53 RHSIFnCIE (0, 1, 2, 3) Register Contents (2/2)

Bit Position	Bit Name	Function
2	AKEE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).AKE(N) bit. 0: Does not assert int_hsif_err when the RHSIFnCST(N).AKE(N) bit is 1. 1: Asserts int_hsif_err when the RHSIFnCST(N).AKE(N) bit is 1.
1	AKRE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).AKR(N) bit. 0: Does not assert int_hsif_ch(N) when the RHSIFnCST(N).AKR(N) bit is 1. 1: Asserts int_hsif_ch(N) when the RHSIFnCST(N).AKR(N) bit is 1.
0	RARE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).RAR(N) bit. 0: Does not assert int_hsif_ch(N) when the RHSIFnCST(N).RAR(N) bit is 1. 1: Asserts int_hsif_ch(N) when the RHSIFnCST(N).RAR(N) bit is 1.

(36) RHSIFnCAR (0, 1, 2, 3) — Channel (0, 1, 2, 3) Read/Write Address RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CAR(N)[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CAR(N)[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.54 RHSIFnCAR (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31 to 0	CAR(N)	These bits specify the address of the area that is the target of a Read or Write Command when L2, operating as an initiator node, transmits the Read or Write Command. When the request size is 16 bits, the CAR(N)[0] bit must always be 0. When the request size is 32 bits, the CAR(N)[1:0] bits must always be 00.

(37) RHSIFnCWD (0, 1, 2, 3) — Channel (0, 1, 2, 3) Write Data RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CWD(N)[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CWD(N)[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.55 RHSIFnCWD (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31 to 0	CWD(N)	<p>These bits are used to set the data to be transmitted when L2, operating as an initiator node, transmits a Write Command.</p> <p>When the request size is 8 bits, set the data in the CDW(N)[7:0] bits.</p> <p>When the request size is 16 bits, set the data in the CDW(N)[15:0] bits.</p> <p>When the request size is 32 bits, set the data in the CDW(N)[31:0] bits.</p>

(38) RHSIFnCRD (0, 1, 2, 3) — Channel (0, 1, 2, 3) Read Data RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRD(N)[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRD(N)[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.56 RHSIFnCRD (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31 to 0	CRD(N)	<p>These bits contain the data that is received from a target node after L2, operating as an initiator node, has transmitted a Read or ID Command to the target node.</p> <p>When the request size is 8 bits, reference the CRD(N)[7:0] bits.</p> <p>When the request size is 16 bits, reference the CRD(N)[15:0] bits.</p> <p>When the request size is 32 bits, reference the CRD(N)[31:0] bits.</p> <p>These bits are valid only when RHSIFnCST(N).RAR(N) = 1.</p>

27.4.3 Operation

27.4.3.1 Initial Settings

The procedure for the initial settings of L2 is as follows:

- (1) Set the memory windows
- (2) Set the channels
- (3) Set the link partner authentication (only for the systems that use the target node function and require authentication)*¹
- (4) Set the transmission stream (only for the systems that transmit Stream Commands by using the initiator node function)
- (5) Set the reception stream (only for the systems that receive Stream Commands by using the target node function)

Note 1. To complete link partner authentication, you need to at least specify the settings below before step (3).

At step (1): Open memory window for the IP's register space.

for first authentication: RHSIFnAID0-7

for second authentication: RHSIFnARD0-3

At step (2): Enable target node function on at least one channel.

The following sections describe how to specify settings with individual procedural steps.

(1) Setting Memory Windows

Figure 27.6 shows the flow to set memory windows in L2.

When the target node function of L2 is enabled, link partners can access (read and write) information in the product with L2 installed. If, however, unspecified link partners are permitted to access L2 without restriction, security problems can occur. Therefore, L2 has a function to enable you to specify the space (called a “window”) that is permitted to be accessed.

You can set up to four windows independently of channels. For each window, specify a start address and a size to prevent window areas from overlapping. You can also set read and write permissions separately for individual windows.

This window setting is valid for H-Bus requests that are issued by RHSIF as master. For details, refer to **Section 27.4.3.3, Non-Stream Command Reception by Target Node**, **Section 27.4.3.4, Stream Command Transmission by Initiator Node** and **Section 27.4.3.5, Stream Command Reception by Target Node**.

If you set windows at a time other than during the initial settings after a reset, you must disable the initiator node function on all channels (clear the RHSIFnCMD(N).INME(N) bit to 0), and the target node function on all channels (clear the RHSIFnCMD(N).TNME(N) bit to 0) before setting windows.

In normal mode, RHSIFnMCT, RHSIFnMW(X)A, and RHSIFnMW(X)S cannot be written to RHSIF’s master port. Reading data from these registers is possible from all master modules.

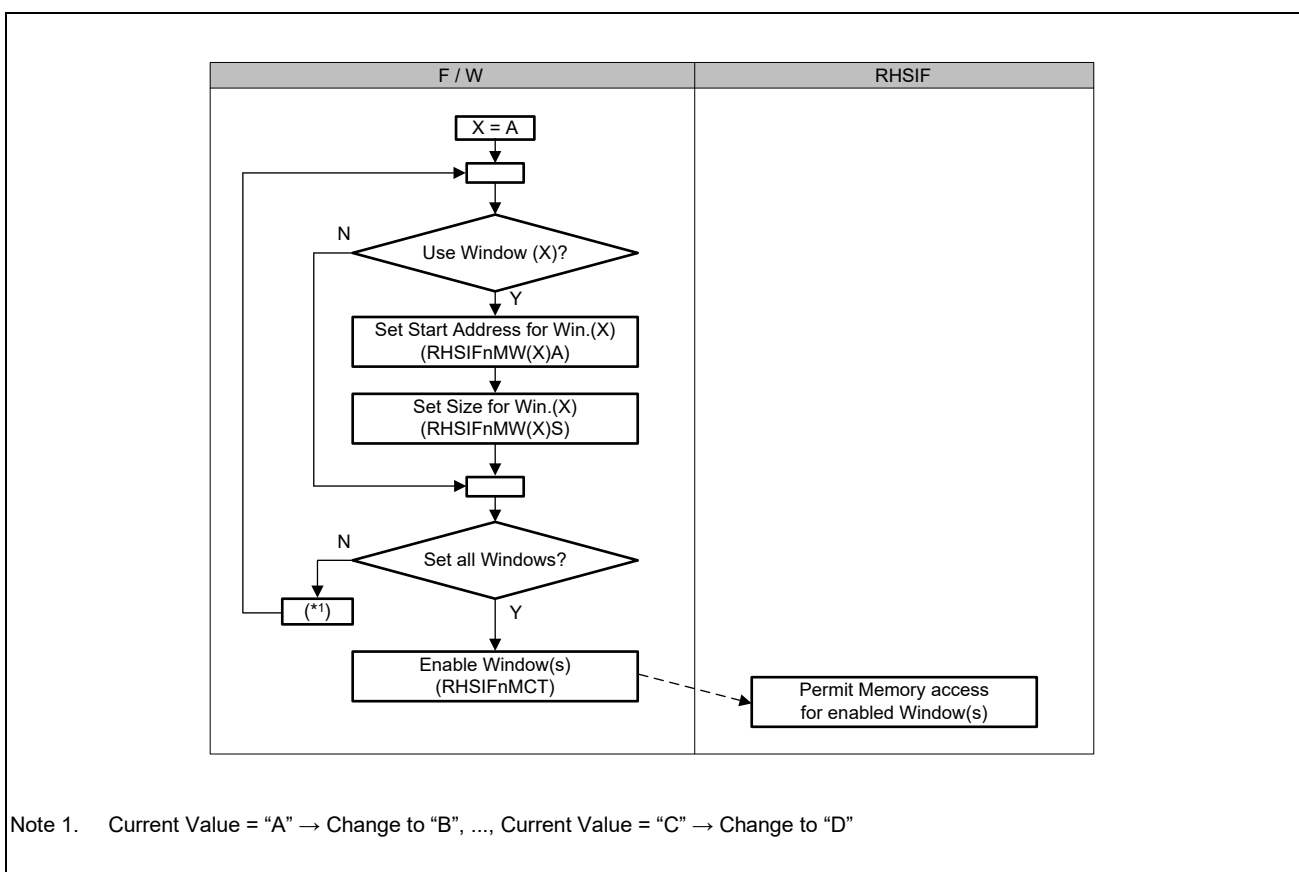


Figure 27.6 Setting Memory Windows

Figure 27.7 shows an example of setting memory windows.

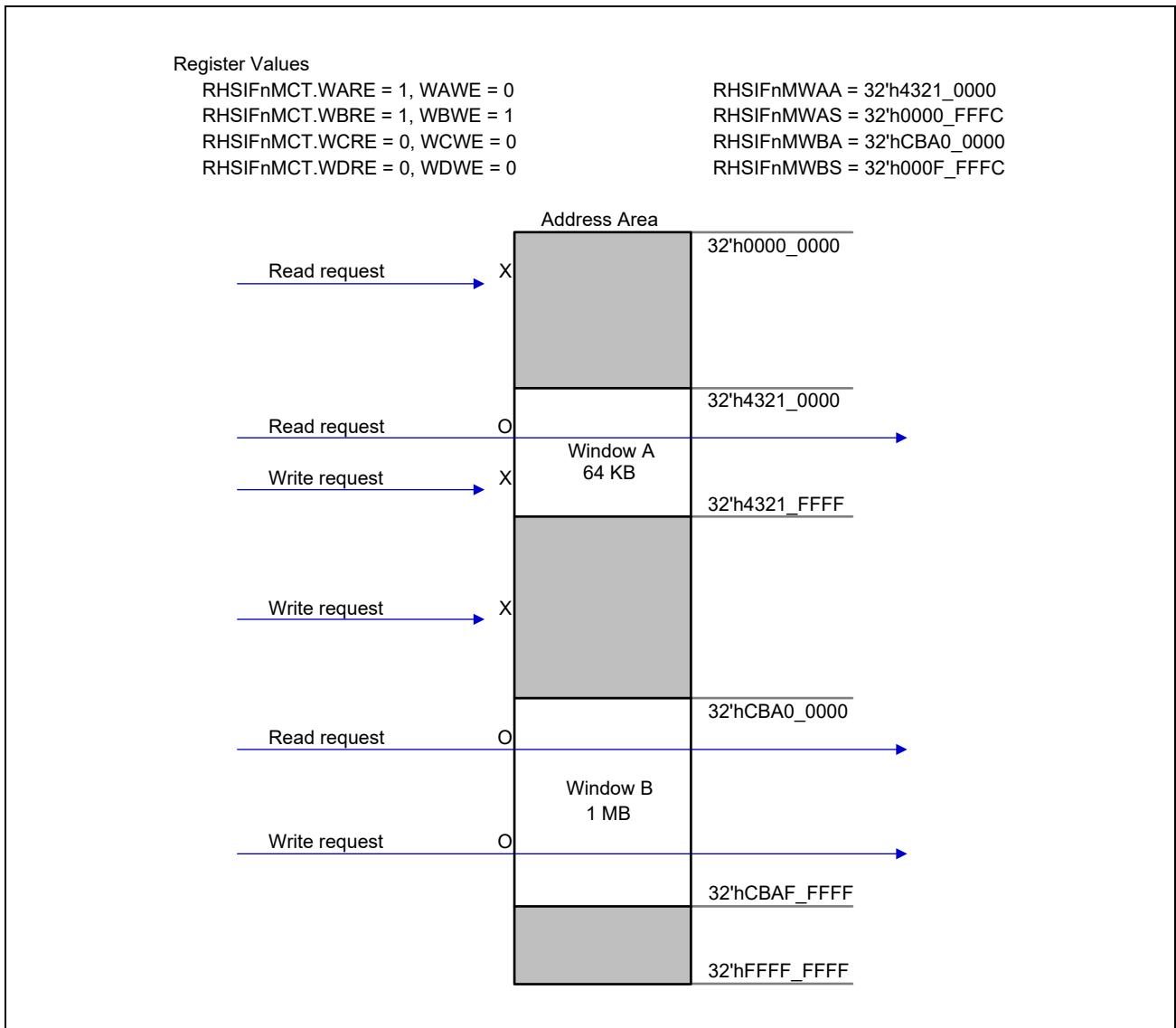


Figure 27.7 Example of Setting Memory Windows

(2) Setting Channels

Figure 27.8 shows the flow of setting channels for L2. Enable the functions to be used for individual channels.

When the RHSIFnCMD(N).INME(N) bit is set to 1 for a channel, L2 is enabled to transmit non-Stream Commands (Read, Write, Event, and ID Commands) through the channel. For details, see **Section 27.4.3.2, Non-Stream Command Transmission by Initiator Node**.

Also, specify a reply timeout time according to the system. The reply timer starts when L2 (Layer 2) finishes to transmit last symbol of a request Command. The reply timer stops when one of the following events occurs:

- L2 detects an expected reply Command.
- For Read Command, L2 detects reply Command with error (except CRC error).
- Reply timeout occurs

The reply timeout time that is specified applies to all channels.

When the RHSIFnCMD(N).TNME(N) bit is set to 1 for a channel, L2 is enabled to receive non-Stream Commands (Read, Write, Event, and ID Commands) through the channel. For details, see **Section 27.4.3.3, Non-Stream Command Reception by Target Node**.

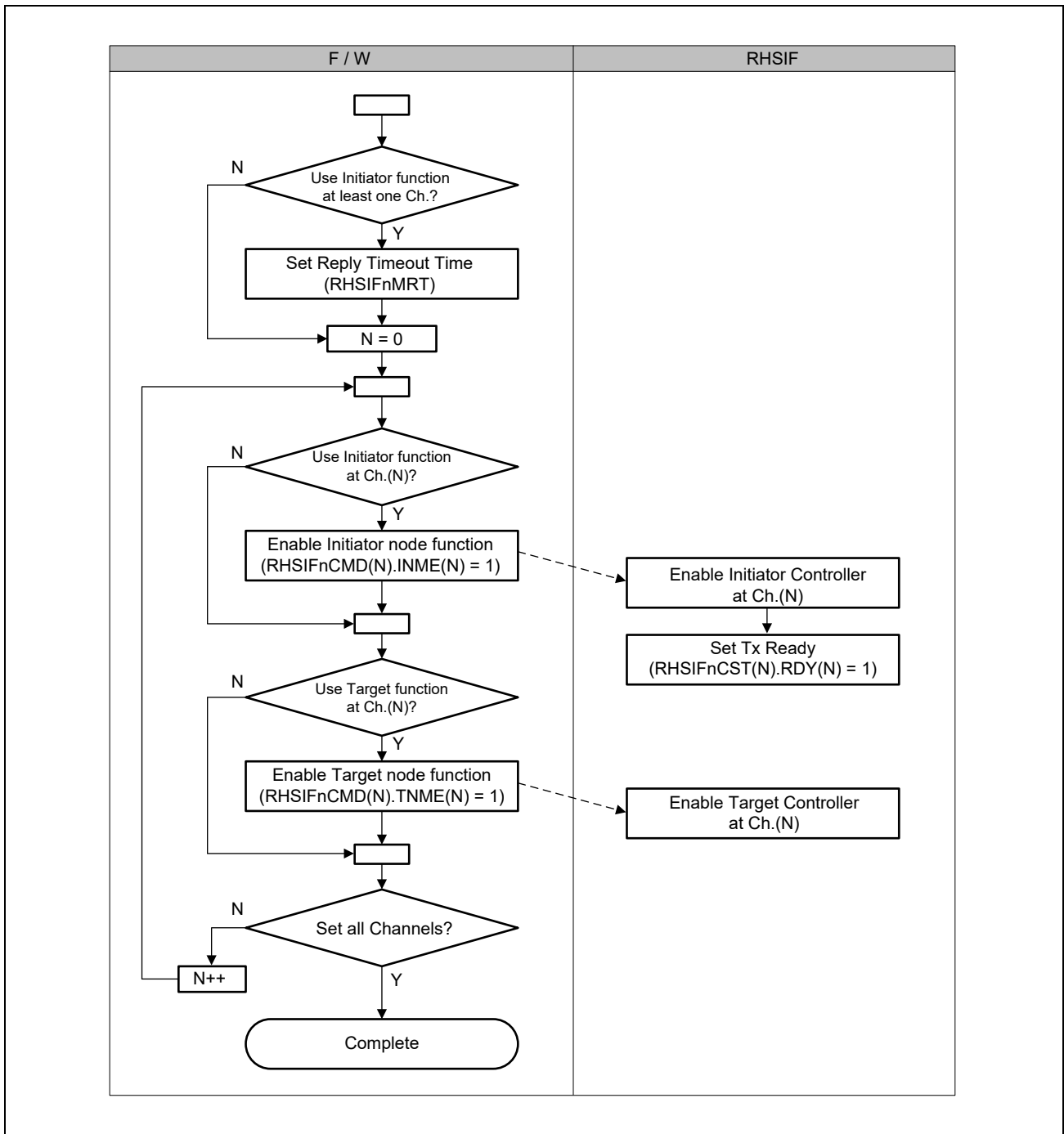


Figure 27.8 Setting Channels

(3) Setting Link Partner Authentication

When the target node function of L2 is enabled, link partners can access (read and write) information in the product with L2 installed. If, however, unspecified link partners are permitted to access L2, security problems can occur. Therefore, L2 has a function to authenticate link partners.

Link partner authentication is performed by using the following two steps:

1. Authentication based on a 256-bit key ID*¹
2. Challenge-and-response authentication*²

You can enable or disable each type of authentication by using FLASH option byte. When link partner authentication is disabled, link partners can access the internal space of the product with L2 installed without performing the authentication procedure described in this section. Specify settings of the relevant FLASH option byte for RHSIF according to the security level required of the product with L2 installed.

Note 1. First Authentication (ID-based Authentication) needs setting.

Product shipping setting is invalid.

For the values of the FLASH option byte, refer to **Section 42, Basic Hardware Protection (BHP)**. Refer to **Table 42.18** for details.

Note 2. Second Authentication (Challenge and Response Authentication) needs setting.

Product shipping setting is invalid.

For the values of the FLASH option byte, refer to the RH850/E2x ICUMD User's Manual.

(a) First Authentication (ID-Based)

Figure 27.9 shows a concept of the first authentication (ID-based authentication), and the procedure for the authentication is described below.

- (1) The link partner issues a Write Command (32 bits) for the address corresponding to the RHSIFnAID0 register in L2. Write data is the [31:0] bits of key ID (256 bits).
- (2) Upon receiving the Write Command, the target controller of L2 checks the address in the Command.
 - A) When the address matches any of RHSIFnAID0 to RHSIFnAID7, the target controller determines the Write Command to be valid, and then issues a master write request to the H-Bus.
 - B) When the address does not match any of RHSIFnAID0 to RHSIFnAID7, the target controller ignores the Write Command without returning any reply Command.
- (3) Upon receiving the H-Bus write request, the register unit of L2 checks the master ID in the request.
 - A) When the master ID matches the ID of L2, L2 determines the request to be valid, and then writes data to RHSIFnAID0.
 - B) When the master ID does not match, L2 does not execute the Write Command to RHSIFnAID0, and then returns an OKAY response.
- (4) In a manner similar to steps (1) to (3), data is set in the RHSIFnAID1 to RHSIFnAID7 registers.
- (5) When data is set in RHSIFnAID7, L2 compares the data (32 bits x 8 = 256 bits) in RHSIFnAID0 to RHSIFnAID7 with the data of FLASH option byte.

Refer to **Section 42, Basic Hardware Protection (BHP)** and **Table 42.25** for details.

- A) If the data matches each other, L2 determines that the link partner passes the authentication. When only the first authentication is required, L2 determines whether the Read or Write Command from the link partner is valid according to the window setting described in **Section 27.4.3.1(1), Setting Memory Windows**.

When the second authentication is also required, the procedure described in **Section 27.4.3.1(2), Setting Channels** is performed.

- B) If the data does not match, L2 determines that the link partner fails the authentication. In such a case, the authentication procedure must be repeated beginning from step (1). If the link partner fails the authentication three consecutive times, L2 determines that the link partner is unauthorized, and will not let the link partner pass the authentication until a reset signal is asserted.

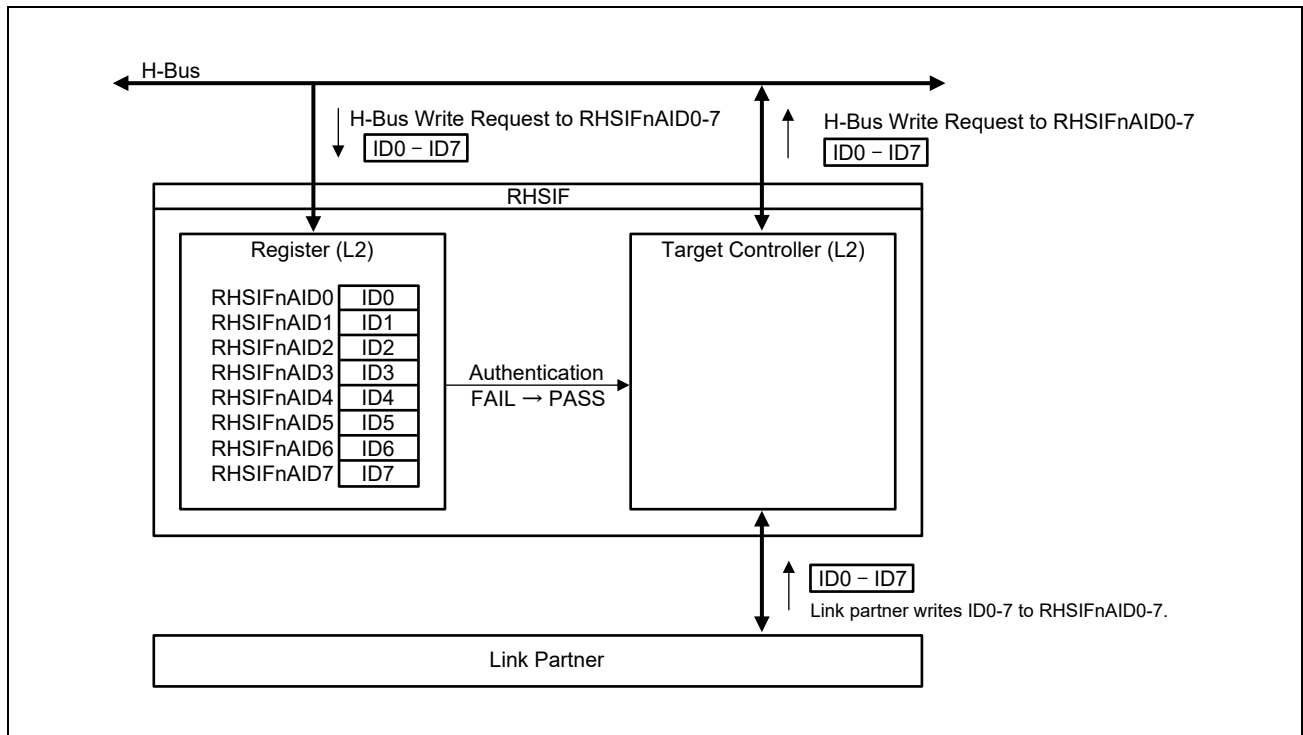


Figure 27.9 Concept of ID-Based Authentication

(b) Second Authentication (Challenge and Response)

Figure 27.10 and **Figure 27.11** show a concept of the second authentication (challenge-and-response authentication), and the procedure for the authentication is described below. Note that this manual describes only how to set the data to be used for challenge and response. For the procedure of challenge and response, see the manual for the product in which L2 is installed.

Step 1 Preparation of the challenge data (Figure 27.10)

- (1) Prepare the challenge data, and send it to the Link Partner. The way to generate the challenge data can be selected. In the example in this figure, challenge data is generated by ICUM and send it via RHSIF to the Link Partner.

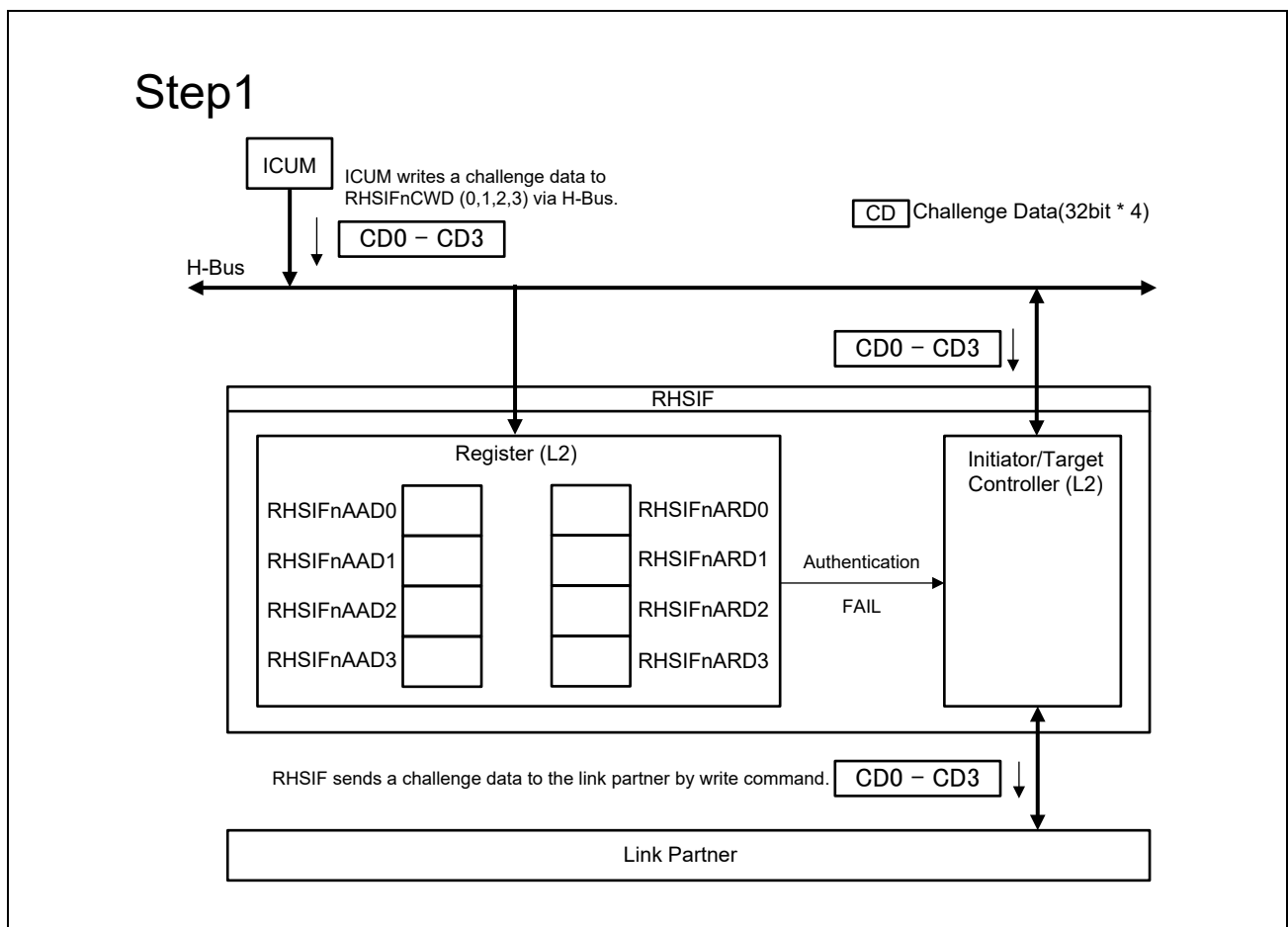


Figure 27.10 Concept of Challenge and Response Authentication (Preparation of the challenge data)

Step 2 Calculate the response data and compare it (Figure 27.11)

- (1) The security module (ICUM) connected to the H-Bus calculates Answer (Response) data from Challenge data, and then sets the [31:0] bits of the Response data in the RHSIFnAAD0 register in L2.
- (2) Upon receiving the H-Bus write request, the register unit of L2 checks the master ID in the request.
 - A) When the master ID matches the one in the ICUM, L2 determines the request to be valid, and then writes data to RHSIFnAAD0.
 - B) When the master ID does not match, L2 does not execute the Write Command to RHSIFnAAD0, and then returns an OKAY response.
- (3) In a manner similar to steps (1) and (2), data is set in the RHSIFnAAD1 to RHSIFnAAD3 registers.
- (4) The link partner issues a Write Command (32 bits) for the address corresponding to the RHSIFnARD0 register in L2. Write data is the [31:0] bits of Response data.
- (5) Upon receiving the Write Command, the target controller of L2 checks the address in the Command.
 - A) When the address matches any of RHSIFnARD0 to RHSIFnARD3, the target controller determines the Write Command to be valid, and then issues a master write request to the H-Bus.
 - B) When the address does not match any of RHSIFnARD0 to RHSIFnARD3, the target controller ignores the Write Command without returning any reply Command.
- (6) Upon receiving the H-Bus write request, the register unit of L2 checks the master ID in the request.
 - A) When the master ID matches the ID of L2, L2 determines the request to be valid, and then writes data to RHSIFnARD0.
 - B) When the master ID does not match, L2 does not execute the Write Command to RHSIFnARD0, and then returns an OKAY response.
- (7) In a manner similar to steps (4) to (6), data is set in the RHSIFnARD1 to RHSIFnARD3 registers.
- (8) When data is set in RHSIFnARD3, L2 compares the data (32 bits x 4 = 128 bits) in RHSIFnARD0 to RHSIFnARD3 with the data (32 bits x 4 = 128 bits) preset in RHSIFnAAD0 to RHSIFnAAD3.
 - A) If the data matches each other, L2 determines that the link partner passes the authentication. After that, L2 determines whether the read or Write Command from the link partner is valid according to the window setting described in **Section 27.4.3.1(1), Setting Memory Windows**.
 - B) If the data does not match, L2 determines that the link partner fails the authentication. In such a case, the authentication procedure must be repeated beginning from step (1). If the link partner fails the authentication three consecutive times, L2 determines that the link partner is unauthorized, and will not let the link partner pass the authentication until a reset signal is asserted.

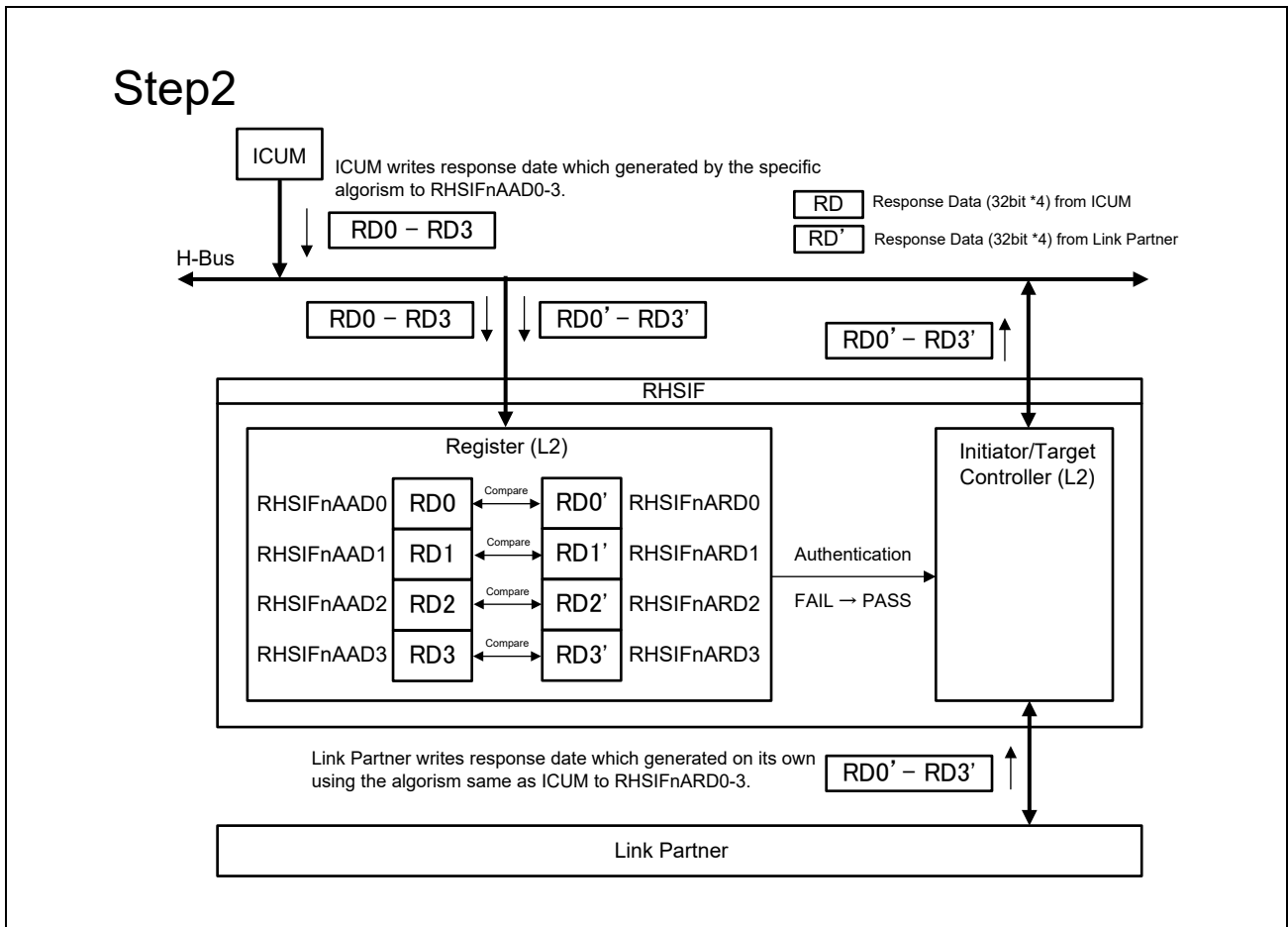


Figure 27.11 Concept of Challenge and Response Authentication (Calculates and Compares Response Data)

(4) Setting Tx Stream for Initiator Node

Figure 27.12 shows the flow of the initial settings of Stream Command transmission by L2.

RHSIF's protocol defines the data payload size of a Stream Command as 256 bits. You, however, can also set the data payload size to 128 bits for L2. Note that the data payload size can be changed only at the time of the initial settings. You cannot change this setting dynamically during system operation. Specify an appropriate value in the relevant parameter during the initial settings according to the system.

There are two modes for receiving NACK Command.

- RHSIFnSTMD.STNK = 0: L2 regards reception of the NACK Command as occurrence of an error.
- RHSIFnSTMD.STNK = 1: L2 regards reception of the NACK Command as reception of an ACK Command.

Select an appropriate mode according to the application program.

For details about the flow of Stream Command transmission, see **Section 27.4.3.4, Stream Command Transmission by Initiator Node**.

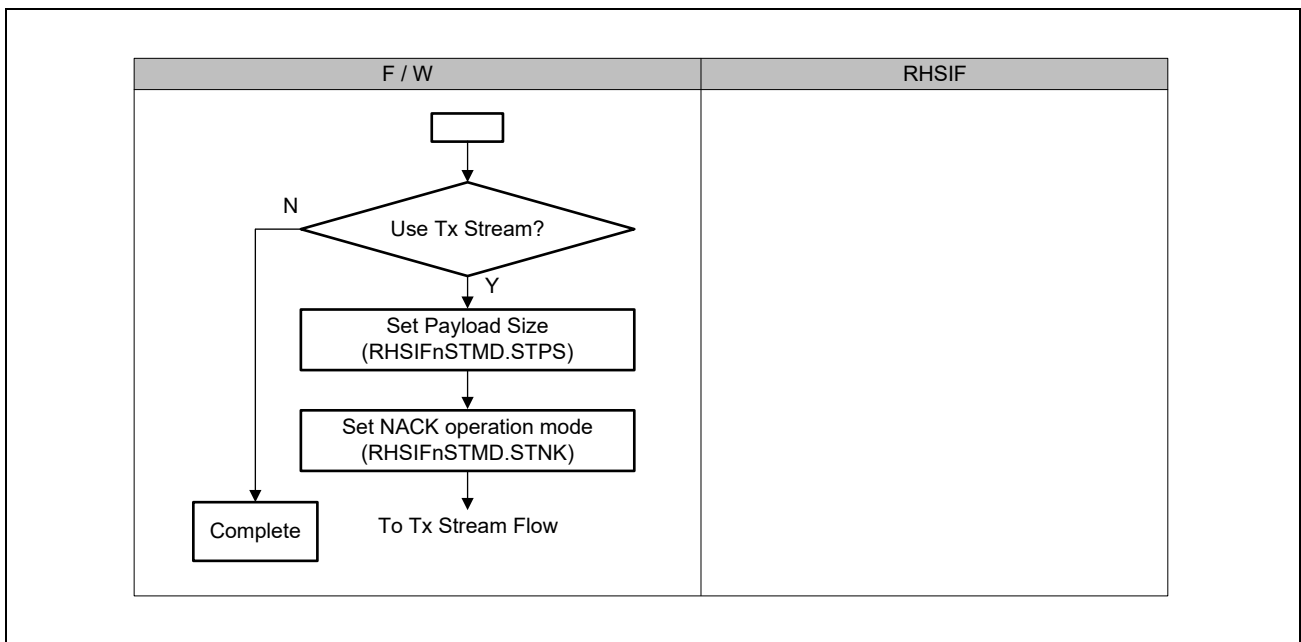


Figure 27.12 Initial Setting of Tx Stream

(5) Setting Rx Stream for Target Node

Figure 27.13 shows the flow of the initial settings of Stream Command reception by L2.

RHSIF defines the data payload size of a Stream Command as 256 bits. You, however, can also set the data payload size to 128 bits for L2. Note that the data payload size can be changed only at the time of the initial settings. You cannot change this setting dynamically during system operation. Specify an appropriate value in the relevant parameter during the initial settings according to the system.

How to handle an error occurring in a command that is received or during data transfer depends on the application program. L2 allows you to select the operation that is necessary when an error occurs in the following two modes:

- **RHSIFnSRMD.SRMC = 0:** The RHSIFnSRCT.SRDE bit is cleared automatically when an error occurs.
Select this mode when you want to stop data storage immediately after the error occurs.
Note that, because L2 can receive two Stream Commands concurrently, if an H-Bus error occurs, L2 might store the data up to the data causing the error.
- **RHSIFnSRMD.SRMC = 1:** The RHSIFnSRCT.SRDE bit is not cleared automatically when an error occurs.
Select this mode when you want to let the firmware determine the processing that is necessary if an error occurs.
To stop data storage after the processing is determined, clear the RHSIFnSRCT.SRDE bit by the firmware.

Select an appropriate mode according to the application program.

For details about the flow of Stream Command reception, see **Section 27.4.3.5, Stream Command Reception by Target Node**.

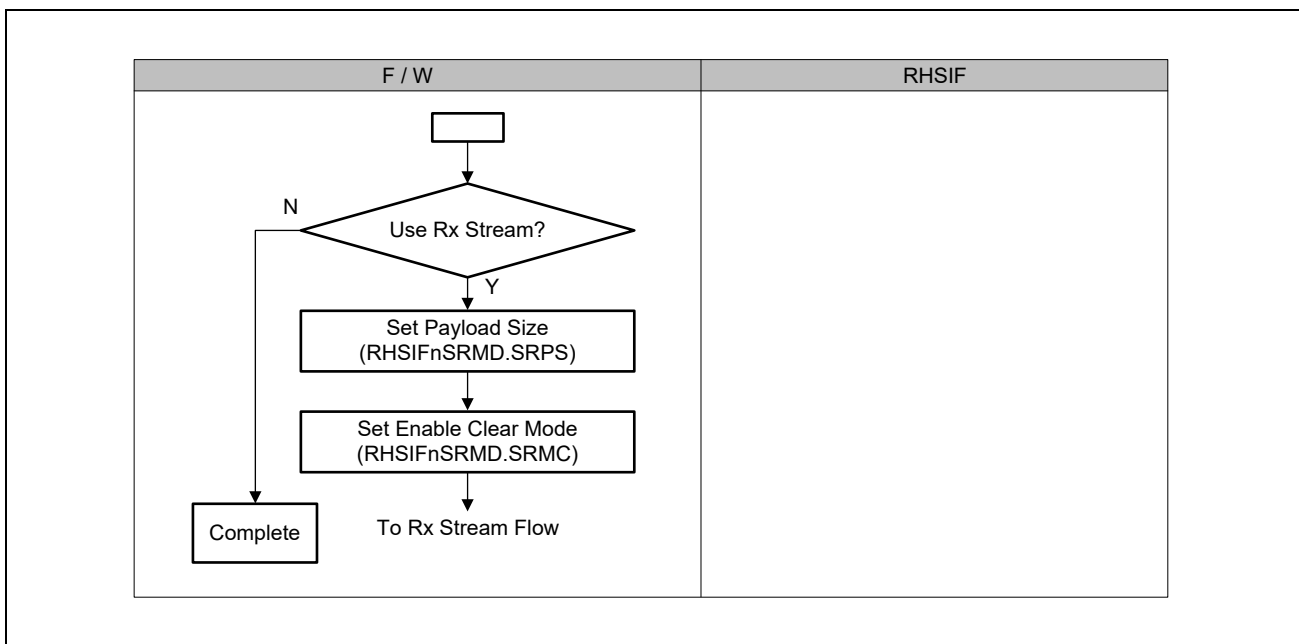


Figure 27.13 Initial Setting of Rx Stream

27.4.3.2 Non-Stream Command Transmission by Initiator Node

Figure 27.14 to **Figure 27.17** show the flows for transmitting non-stream requests (Read, Write, Event, and ID Commands) by using the initiator node function of L2.

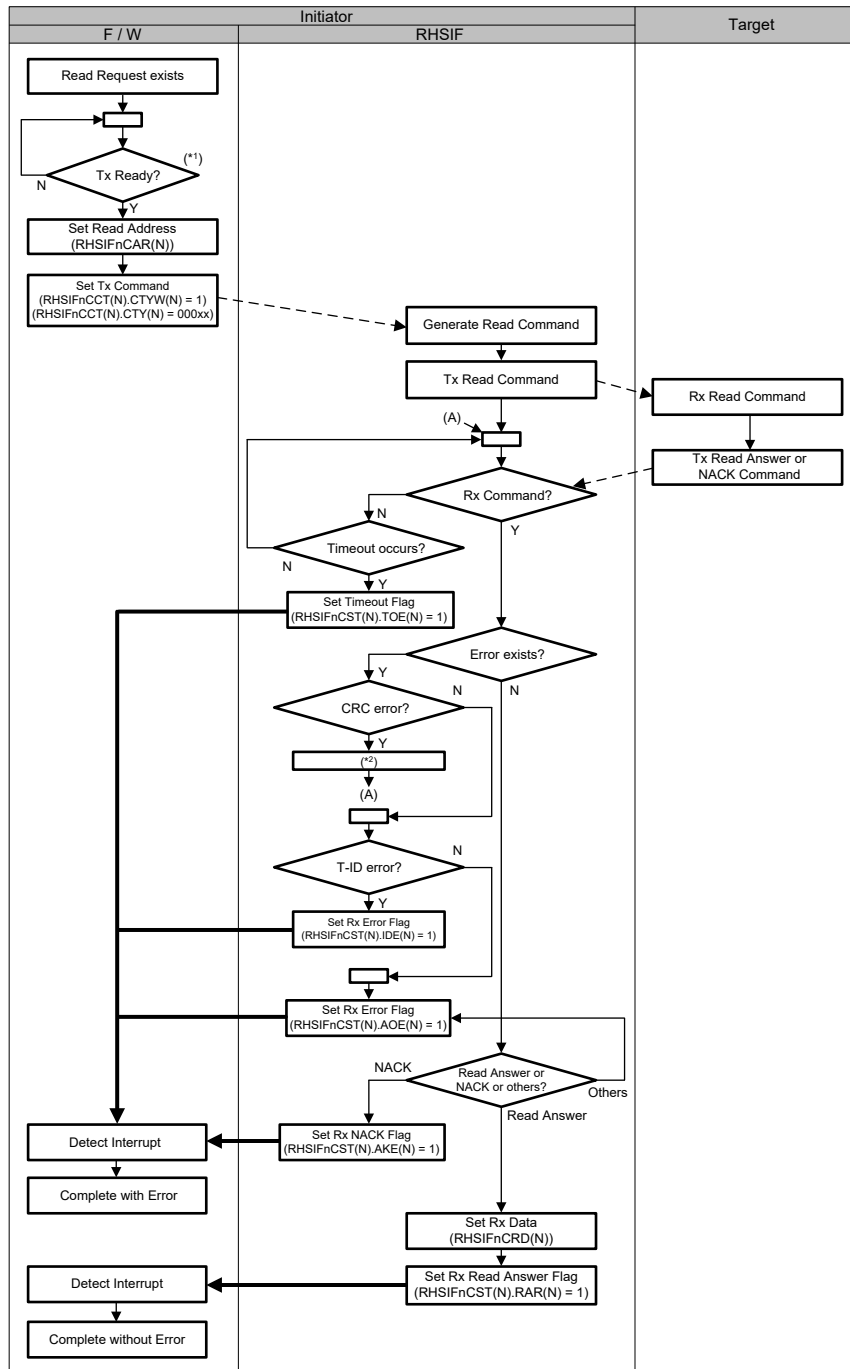
Transmission of non-stream requests is performed under control of the firmware. A request is transmitted by accessing a register, and an interrupt is used as the trigger for checking response.

The following describes the transmission operation, using Read Command transmission (**Figure 27.14**) for example.

When a Command transmission request is found, whether transmission is enabled must be checked first. Transmission is enabled when the RHSIFnCST(N).RDY(N) bit is 1. Then, a target address must be set in the RHSIFnCAR(N) register, and a trigger for transmission must be issued by using the RHSIFnCCT(N) register. When the trigger is issued, L2 generates and transmits the Command.

After transmitting the Command, L2 waits for a reply Command from the link partner. L2 subsequently performs one of the following four operations depending on the response of the link partner and whether a frame error occurs:

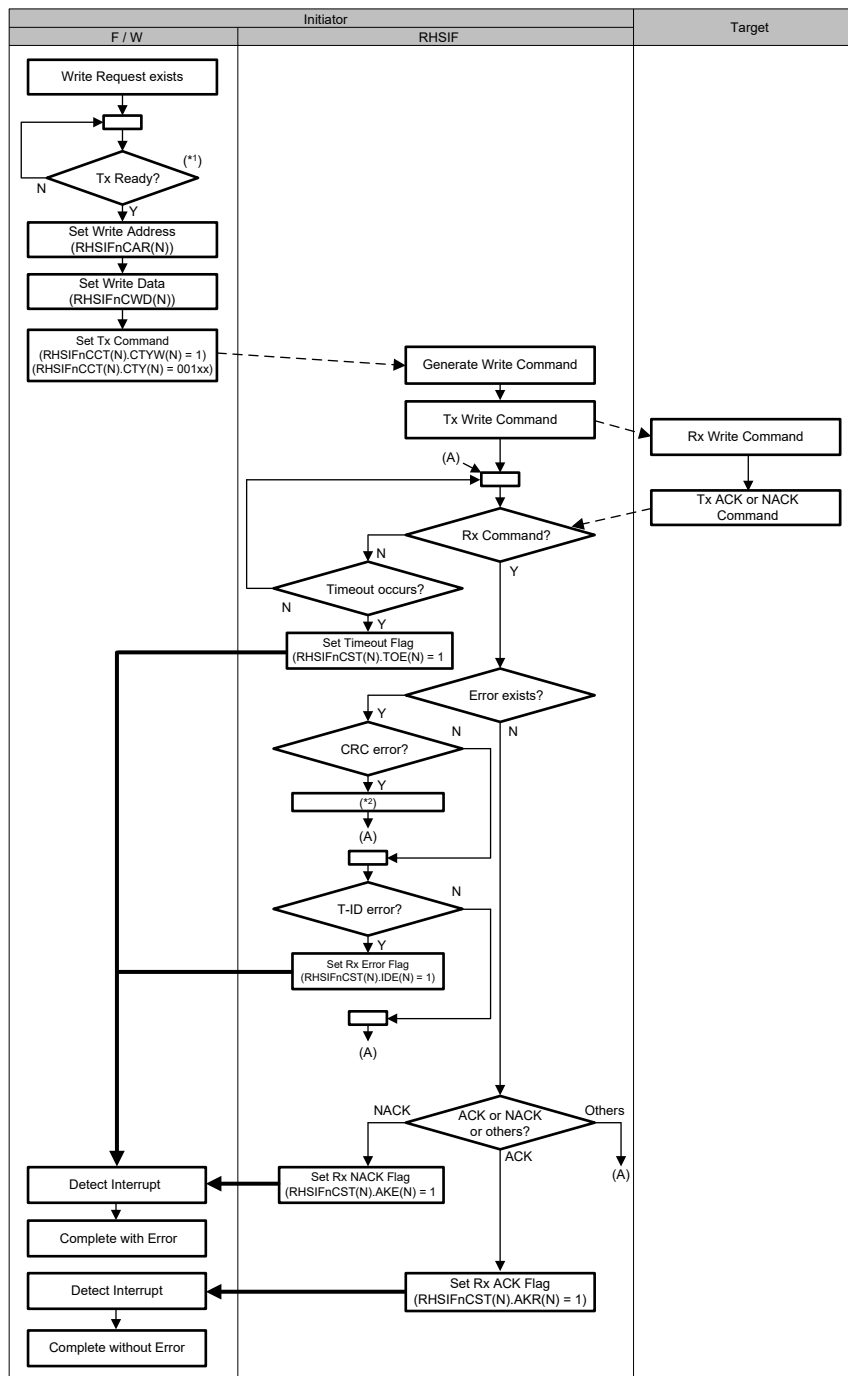
- (1) When the link partner returns a Read Answer Command (normal case)
L2 sets the RHSIFnCST(N).RAR(N) bit, and (when interrupt is enabled) generates an int_hsif_ch(N) interrupt. Checking of read data in the RHSIFnCRD(N) register and other necessary processing must be performed.
- (2) When the link partner returns a NACK Command
L2 sets the RHSIFnCST(N).AKE(N) bit, and (when interrupt is enabled) generates an int_hsif_err interrupt. Retransmission of the Command and other necessary processing must be performed.
- (3) When the reply Command from the link partner involves an error
L2 sets the RHSIFnCST(N).IDE(N) or RHSIFnCST(N).AOE(N) bit according to the type of the error, and (when interrupt is enabled) generates an int_hsif_err interrupt.
Retransmission of the Command and other necessary processing must be performed.
In case reply Command has CRC error, L2 performs special action. For details, see **Section 27.4.3.6, CRC Error Handling**.
- (4) When the link partner does not respond (reply timeout occurs)
If the reply Command from the link partner cannot be detected within the time set in the RHSIFnMRT register after Command transmission, L2 sets the RHSIFnCST(N).TOE(N) bit, and (when interrupt is enabled) generates an int_hsif_err interrupt.
Retransmission of the Command and other necessary processing must be performed.



Note 1. RHSIFnCST(N).RDY(N) = 1

Note 2. In case Rx Command has CRC error, RHSIF performs special action. For details, see **Section 27.4.3.6, CRC Error Handling**.

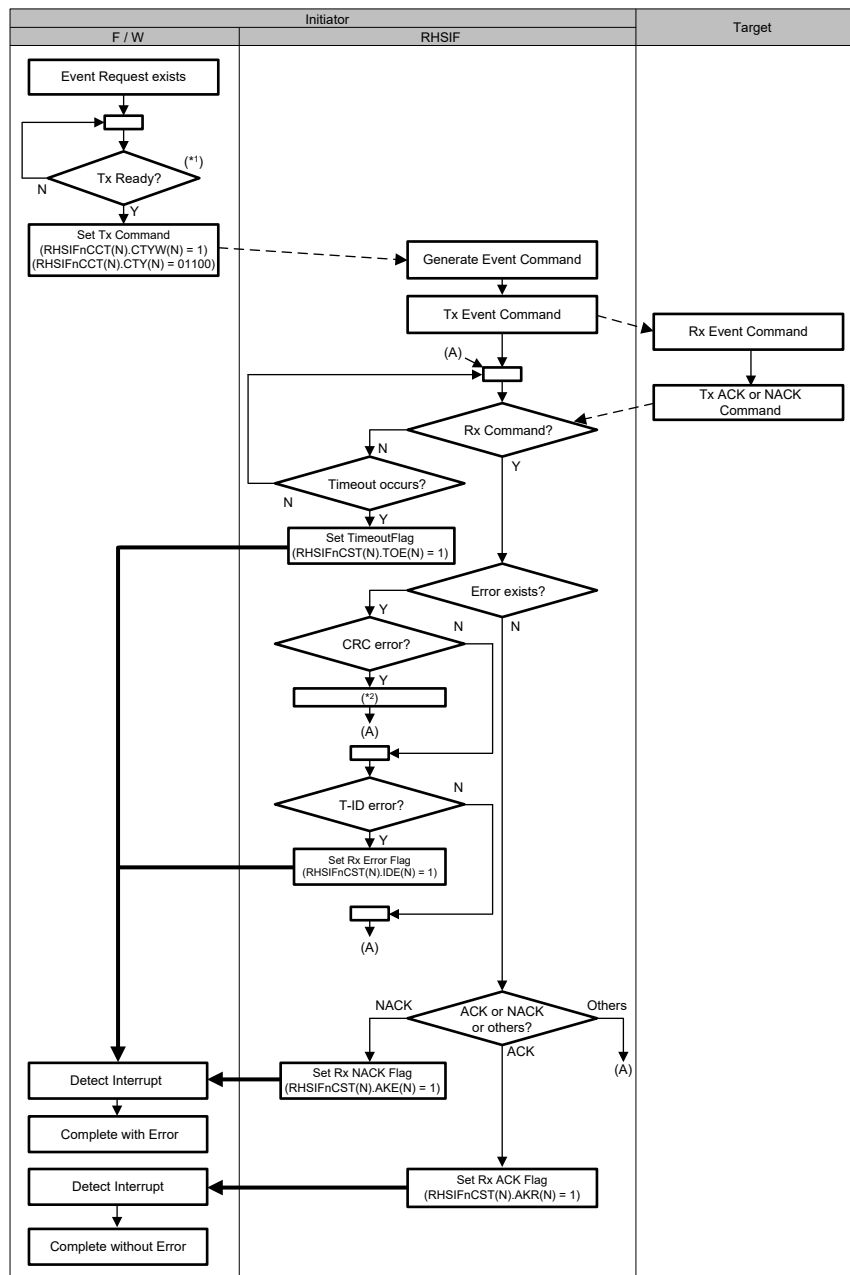
Figure 27.14 Tx Read Command Operation



Note 1. RHSIFnCST(N).RDY(N) = 1

Note 2. In case Rx Command has CRC error, RHSIF performs special action. For details, see **Section 27.4.3.6, CRC Error Handling**.

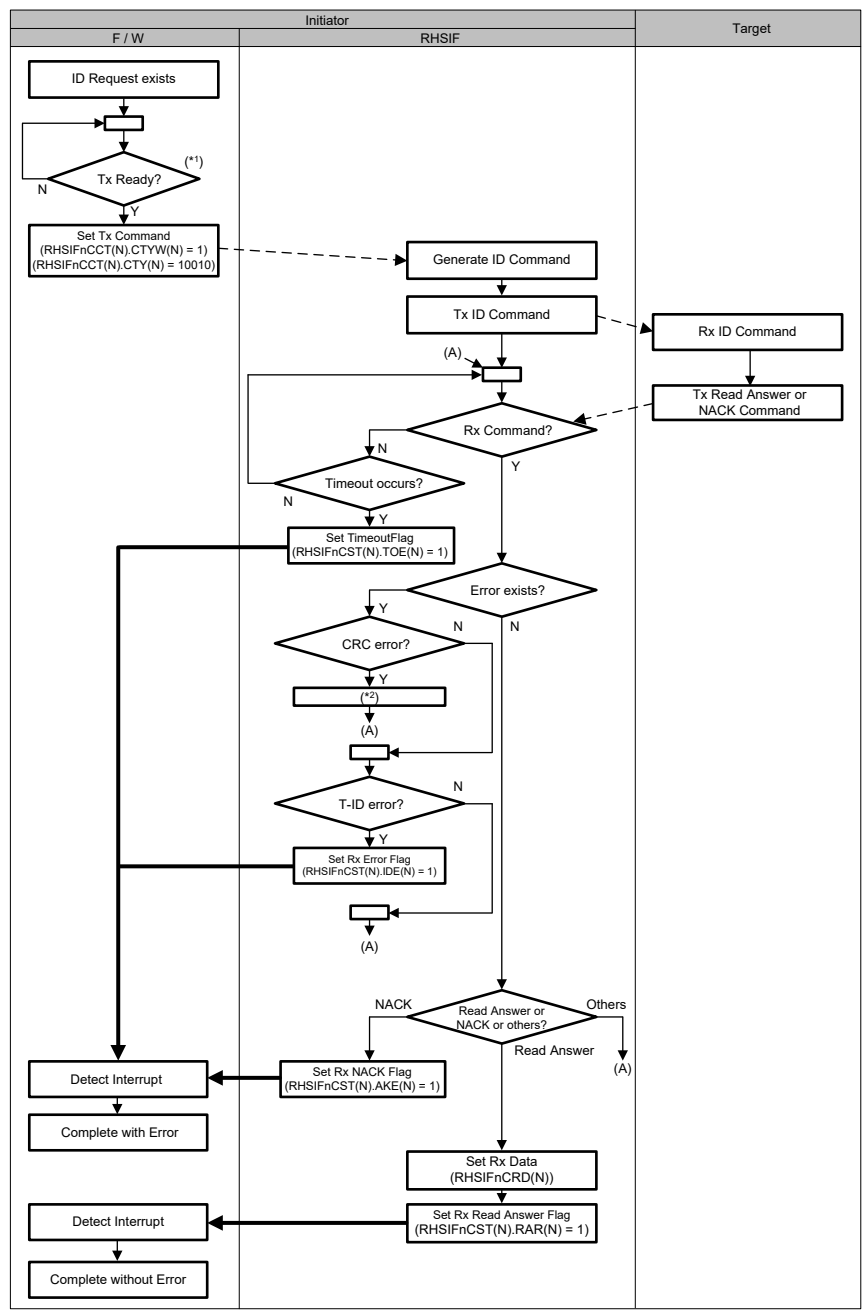
Figure 27.15 Tx Write Command Operation



Note 1. RHSIFnCST(N).RDY(N) = 1

Note 2. In case Rx Command has CRC error, RHSIF performs special action. For details, see **Section 27.4.3.6, CRC Error Handling**.

Figure 27.16 Tx Event Command Operation



Note 1. RHSIFnCST(N).RDY(N) = 1

Note 2. In case Rx Command has CRC error, RHSIF performs special action. For details, see **Section 27.4.3.6, CRC Error Handling.**

Figure 27.17 Tx ID Command Operation

27.4.3.3 Non-Stream Command Reception by Target Node

Figure 27.18 to **Figure 27.21** show the flows for receiving non-stream requests (Read, Write, Event, and ID Commands) by using the target node function of L2.

Because the hardware of L2 automatically processes Commands that are received, the firmware of L2 does not need to perform any processing.

The following describes the reception operation, using the Read Command reception (**Figure 27.18**) for example.

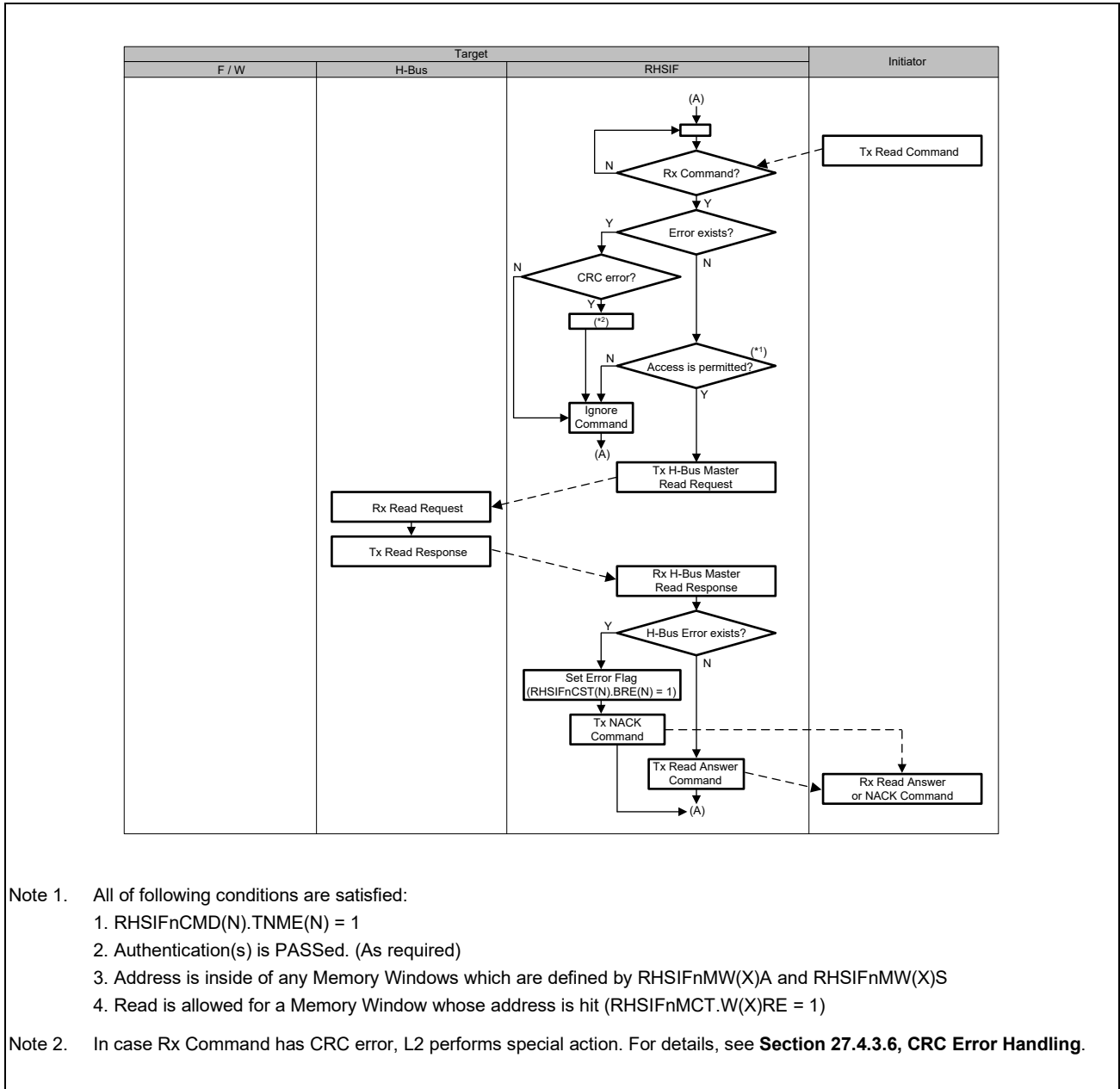
Upon receiving a Command, L2 first checks whether an error has occurred. When an error is detected, L2 discards the command that is received without returning any reply Command. If the command that is received has a CRC error, L2 performs special action. For details, see **Section 27.4.3.6, CRC Error Handling**.

When no error is detected, L2 checks the following items to determine whether read access is enabled:

- (1) Whether the target node function has been enabled for the relevant channel
- (2) Whether authentication has been performed (when authentication is required)
- (3) Whether the target address corresponds to a valid memory window
- (4) Whether reading of the window described in (3) above is permitted

If the result of any of the above checks is negative, L2 discards the command that is received without returning any reply Command.

If the results of all of the above checks are positive, L2 issues a Master Read request to the H-Bus. After receiving a response from the H-Bus, L2 transmits a Read Answer Command (when no H-Bus response error is detected) or an NACK Command (when an H-Bus response error is detected) to the link partner.

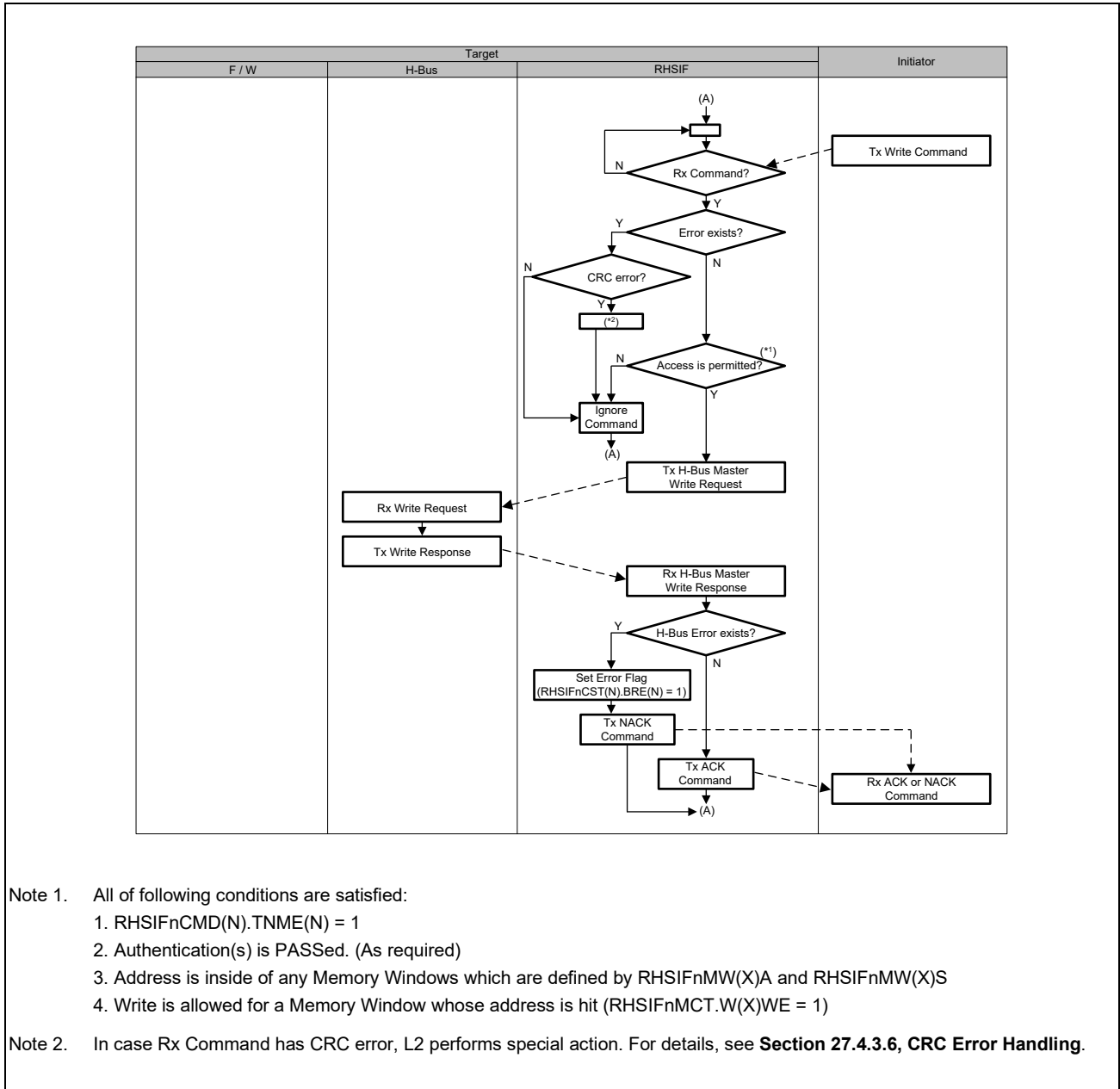


Note 1. All of following conditions are satisfied:

1. RHSIFnCMD(N).TNME(N) = 1
2. Authentication(s) is PASSEd. (As required)
3. Address is inside of any Memory Windows which are defined by RHSIFnMW(X)A and RHSIFnMW(X)S
4. Read is allowed for a Memory Window whose address is hit (RHSIFnMCT.W(X)RE = 1)

Note 2. In case Rx Command has CRC error, L2 performs special action. For details, see **Section 27.4.3.6, CRC Error Handling.**

Figure 27.18 Rx Read Command Operation

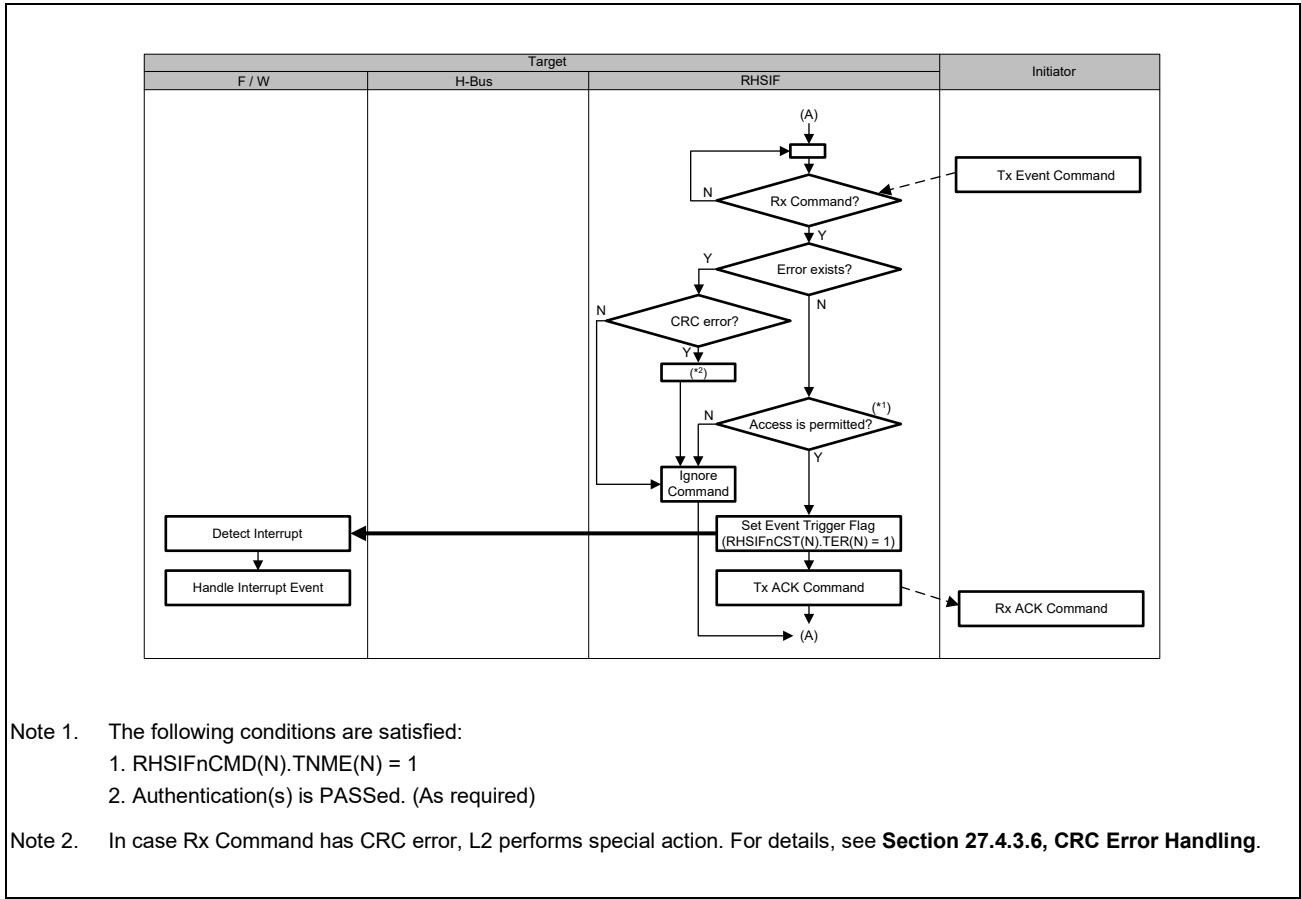


Note 1. All of following conditions are satisfied:

1. $RHSIFnCMD(N).TNME(N) = 1$
2. Authentication(s) is PASSEd. (As required)
3. Address is inside of any Memory Windows which are defined by $RHSIFnMW(X)A$ and $RHSIFnMW(X)S$
4. Write is allowed for a Memory Window whose address is hit ($RHSIFnMCT.W(X)WE = 1$)

Note 2. In case Rx Command has CRC error, L2 performs special action. For details, see **Section 27.4.3.6, CRC Error Handling**.

Figure 27.19 Rx Write Command Operation



Note 1. The following conditions are satisfied:
 1. RHSIFnCMD(N).TNME(N) = 1
 2. Authentication(s) is PASSED. (As required)

Note 2. In case Rx Command has CRC error, L2 performs special action. For details, see **Section 27.4.3.6, CRC Error Handling.**

Figure 27.20 Rx Event Command Operation

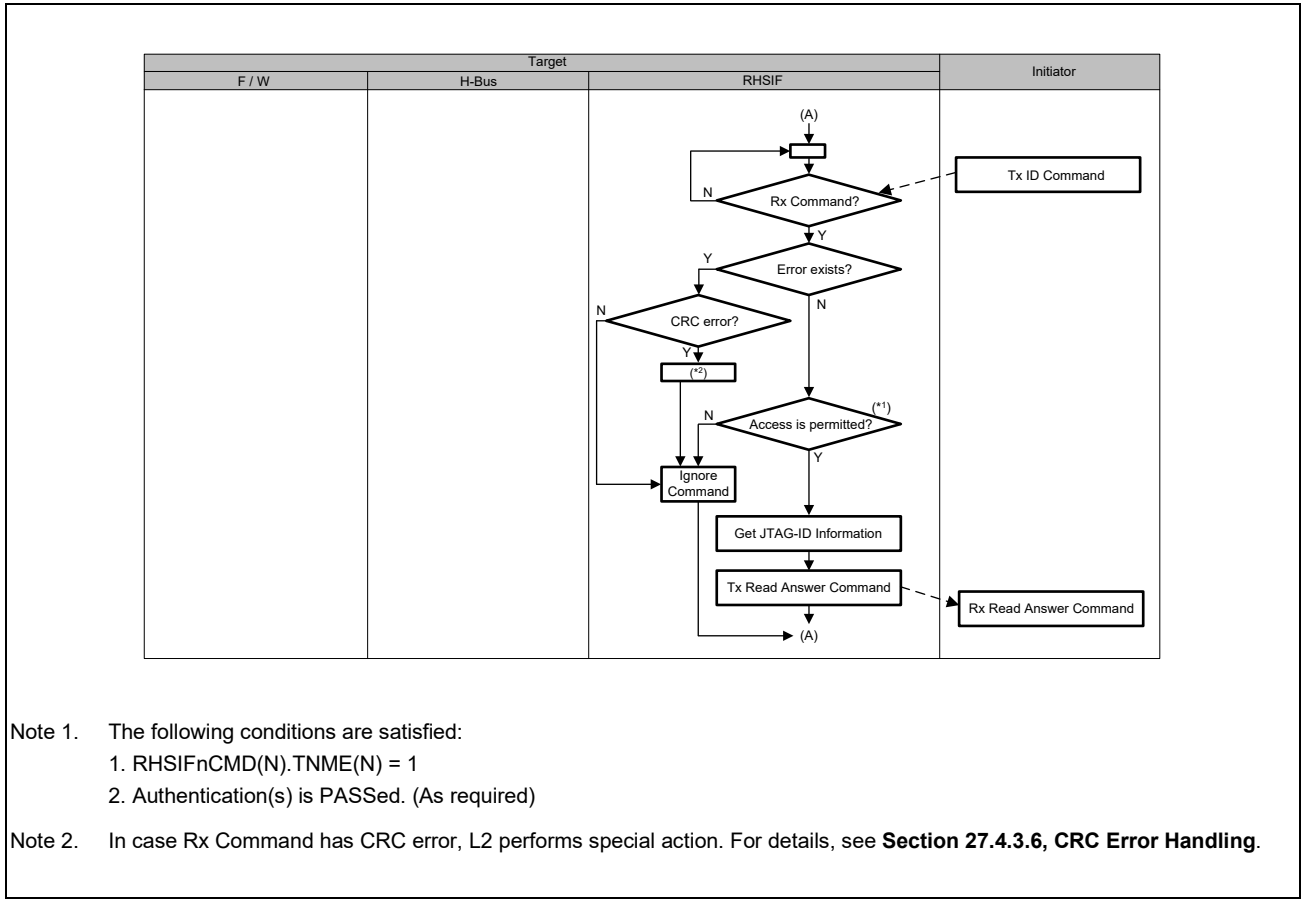


Figure 27.21 Rx ID Command Operation

27.4.3.4 Stream Command Transmission by Initiator Node

Figure 27.22 and **Figure 27.23** show the flow for transmitting Stream Commands by using the initiator node function and internal DMAC of L2.

After preparing stream data, the firmware must set the start address of the area containing the data and the number of bytes to be transferred, and then enable the DMAC.

L2 internally has a 512-bit data buffer, and obtains data by issuing a Master Read request to the H-Bus with the maximum size is 256-bit. For L2, the maximum number of outstanding requests to the H-Bus is set to 2.

If source address does not hit any window or read operation is not allowed for a memory window whose address is hit RHSIF does not issue H-Bus read request. When RHSIF collects all the responses to transmitted H-Bus requests and transmitted request commands, it sets the RHSIFnSTST.STE4 bit, and then stops the DMAC.

If an error is detected in the response from the H-Bus, L2 collects all the responses to transmitted H-Bus requests and transmitted request commands, sets the RHSIFnSTST.STE3 bit, and then stops the DMAC.

If no error is detected in the response from the H-Bus, L2 generates and transmits a Stream Command in units of the size specified by the RHSIFnSTMD.STPS bit. RHSIF allows up to two Stream Commands to be transmitted to a link partner before the link partner responds. Compliant with this specification, L2 can transmit a maximum of two outstanding requests to a link partner.

The Stream Command data to be transmitted is realigned in L2 so that the data alignment meets the specification. For the data alignment, see **Figure 27.24**.

After normally transmitting all data (specified number of bytes to be transferred), L2 sets the RHSIFnSTST.STC bit, and (when interrupt is enabled) generates an `int_hsic_str` interrupt.

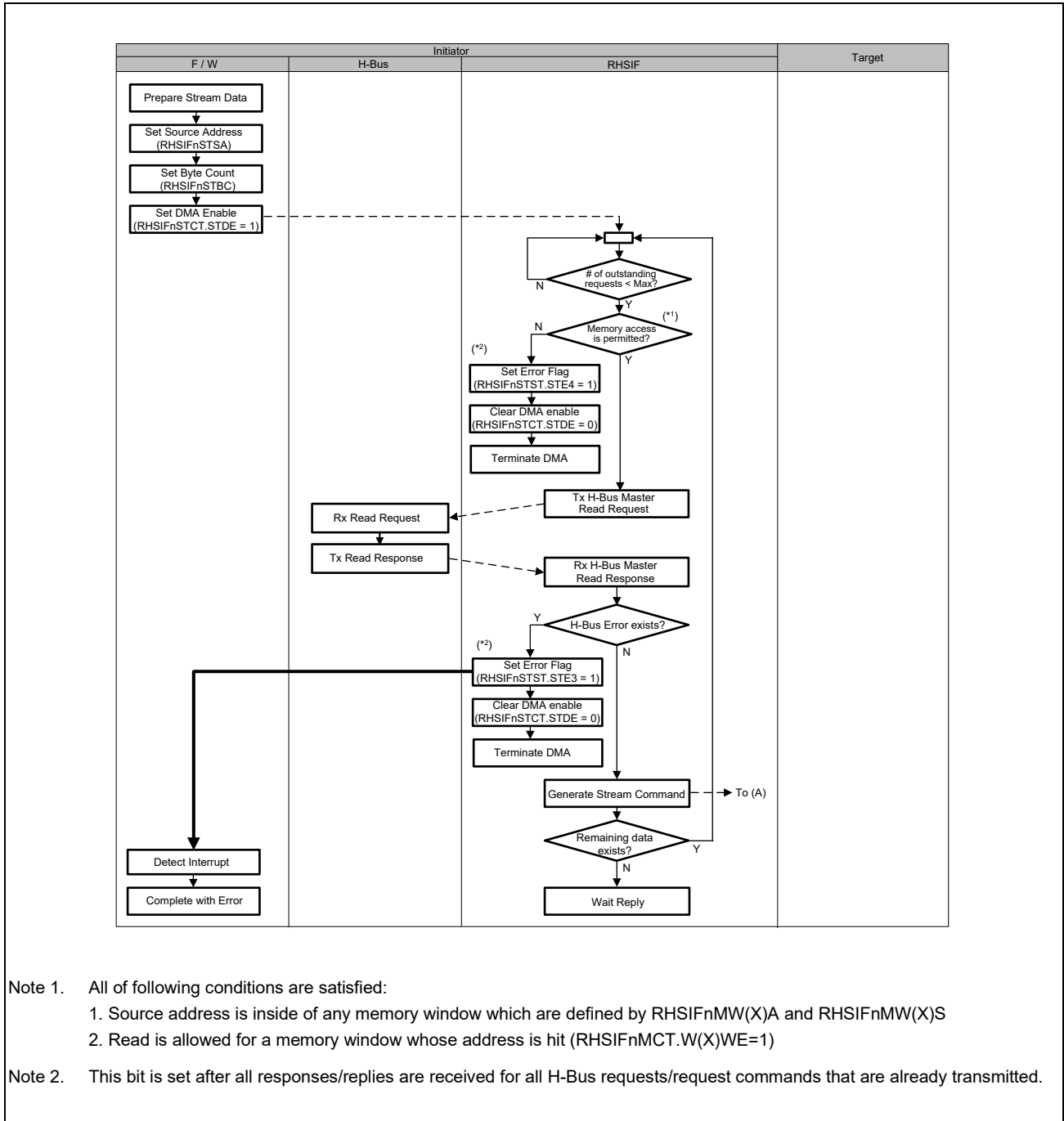
If an error (including timeout) is caused by a reply command during transmission, L2 collects all the responses to transmitted H-Bus requests and transmitted request commands, sets the RHSIFnSTST.STE1 or RHSIFnSTST.STE2 bit, and then stops the DMAC.

Note that the operation L2 performs when it receives a NACK Command depends on the value of the RHSIFnSTMD.STNK bit. When the RHSIFnSTMD.STNK bit is 0, L2 regards reception of the NACK Command as occurrence of an error. It sets the RHSIFnSTST.STE0 bit, and then stops the DMA transfer. When the RHSIFnSTMD.STNK bit is 1, L2 regards reception of the NACK Command as reception of an ACK Command, and continues the DMA transfer. Specify an appropriate value in the RHSIFnSTMD.STNK bit according to the usage of the data to be transferred.

If more than one error (receives NACK, detects T-ID error, or reply timeout) occurs during a DMA operation, only the first error is set to RHSIFnSTST.STE0, STE1, or STE2.

In case reply Command has CRC error, L2 performs special action. For details, see **Section 27.4.3.6, CRC Error Handling**.

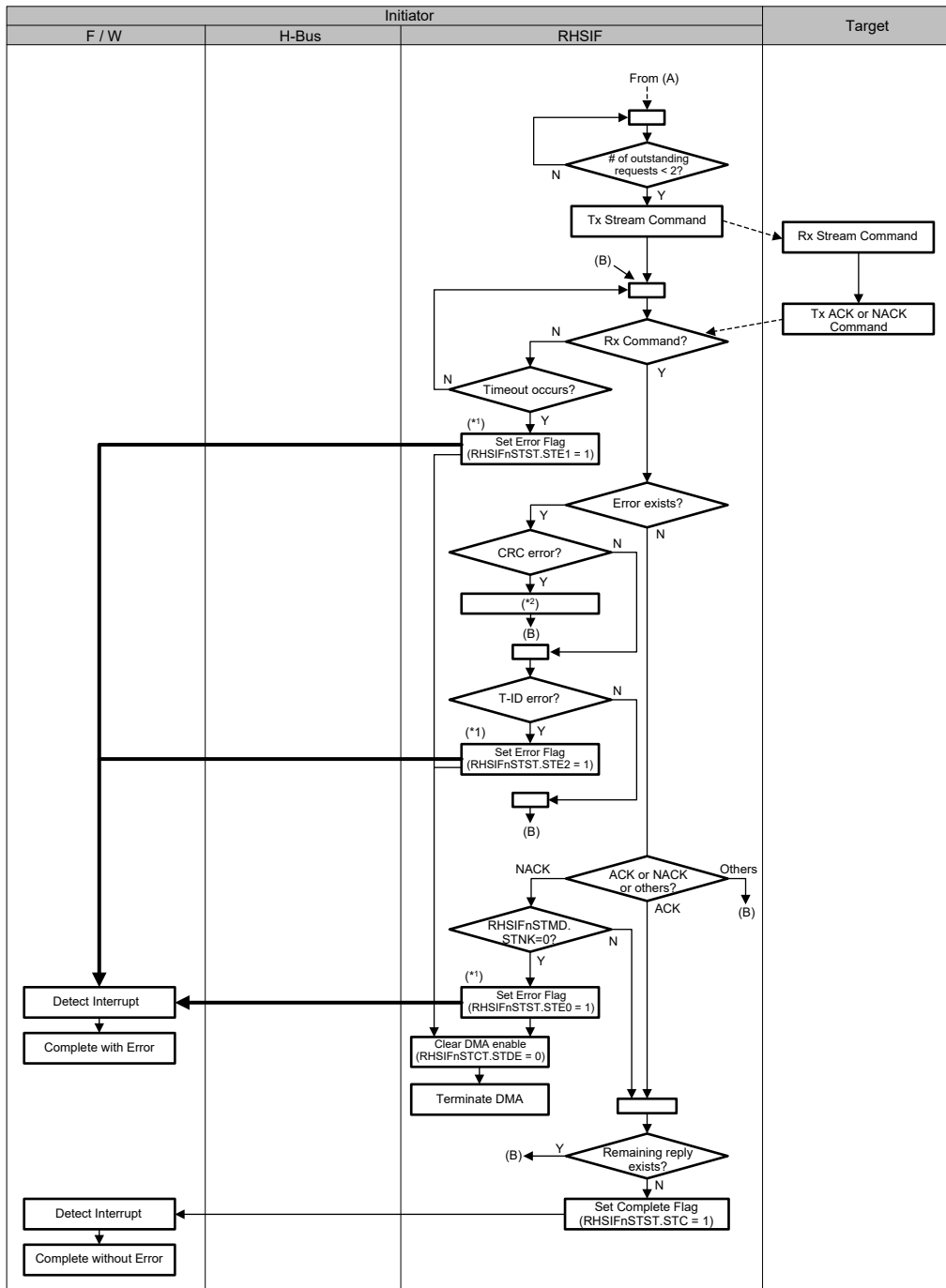
Note that Stream Commands must not be issued together with a non-Stream Command (Read, Write, Event, or ID Command). The firmware must be controlled so that non-Stream Commands are not transmitted while a Stream Command is being transmitted.



Note 1. All of following conditions are satisfied:
 1. Source address is inside of any memory window which are defined by RHSIFnMW(X)A and RHSIFnMW(X)S
 2. Read is allowed for a memory window whose address is hit (RHSIFnMCT.W(X)WE=1)

Note 2. This bit is set after all responses/replies are received for all H-Bus requests/request commands that are already transmitted.

Figure 27.22 Tx Stream Command Operation (1)



Note 1. This bit is set after all responses/replies are received for all H-Bus requests/request commands that are already transmitted.

Note 2. In case Rx Command has CRC error, L2 performs special action. For details, see Section 27.4.3.6, CRC Error Handling.

Figure 27.23 Tx Stream Command Operation (2)

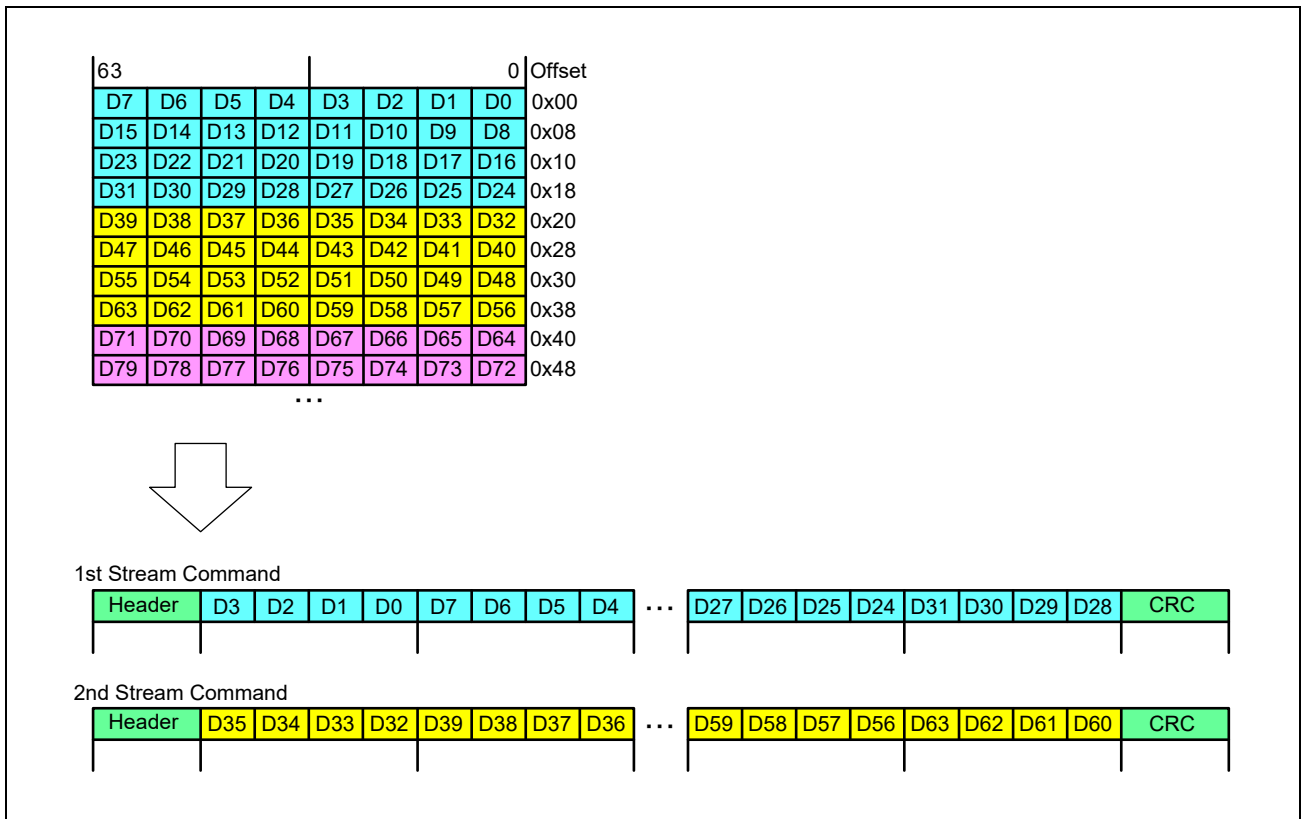


Figure 27.24 Tx Stream Data Alignment

27.4.3.5 Stream Command Reception by Target Node

Figure 27.25 and **Figure 27.26** show the flow for receiving Stream Commands by using the target node function and internal DMAC of L2.

First, the type of interrupt must be selected. L2 provides the following two status types:

- (1) Status indicating that the size of unprocessed data is not 0 (RHSIFnSRST.SRC0)
- (2) Status indicating that the size of unprocessed data is at least the specified size (RHSIFnSRST.SRC1)

An appropriate type must be selected according to the system.

Upon receiving a Stream Command, L2 first checks whether an error has occurred. When an error is detected, L2 discards the received Command without returning any reply Command. If the command that is received has CRC error, L2 performs special action. For details, see **Section 27.4.3.6, CRC Error Handling**.

When the RHSIFnSRMD.SRMC bit is 0, L2 automatically clears the RHSIFnSRCT.SRDE bit to 0 and sets the RHSIFnSRST.SRA to prevent extra data from being stored subsequently.

When no error is detected, L2 checks the following items to determine whether stream transfer is enabled:

- (1) Whether the target node function has been enabled for channel 2
- (2) Whether authentication has been performed (when authentication is required)
- (3) Whether DMA for stream reception has been enabled

If the result of any of the above checks is negative, L2 discards the command that is received without returning any reply Command.

If destination address does not hit any window or write operation is not allowed for a memory window whose address is hit, RHSIF discards the command that is received without returning any reply Command, and sets the RHSIFnSRST.STE2 bit to 1. When the RHSIFnSRMD.SRMC bit is 0, RHSIF automatically clears the RHSIFnSRCT.SRDE bit to 0 and sets the RHSIFnSRST.SRA to prevent extra data from being stored subsequently.

If the results of all the above checks are positive, L2 checks the size of the free space in the stream storage buffer. If the buffer does not have enough free space to store the data that is received, L2 discards the command that is received, returns a NACK Command, and sets the RHSIFnSRST.SRE1 bit to 1. When the RHSIFnSRMD.SRMC bit is 0, L2 automatically clears the RHSIFnSRCT.SRDE bit to 0 and sets the RHSIFnSRST.SRA to prevent extra data from being stored subsequently.

If the buffer has enough free space to store received data, L2 issues a Master Write request to the H-Bus. RHSIF allows up to two Stream Commands to be transmitted to a link partner before the link partner responds. Compliant with this specification, L2 can receive up to two Stream Commands concurrently. The maximum number of outstanding requests to the H-Bus is 2.

The data alignment of the Stream Commands that are received complies with the specifications. Therefore, the Stream Command data to be transmitted is realigned in L2 before transmission to the H-Bus. For the data alignment, see **Figure 27.24**.

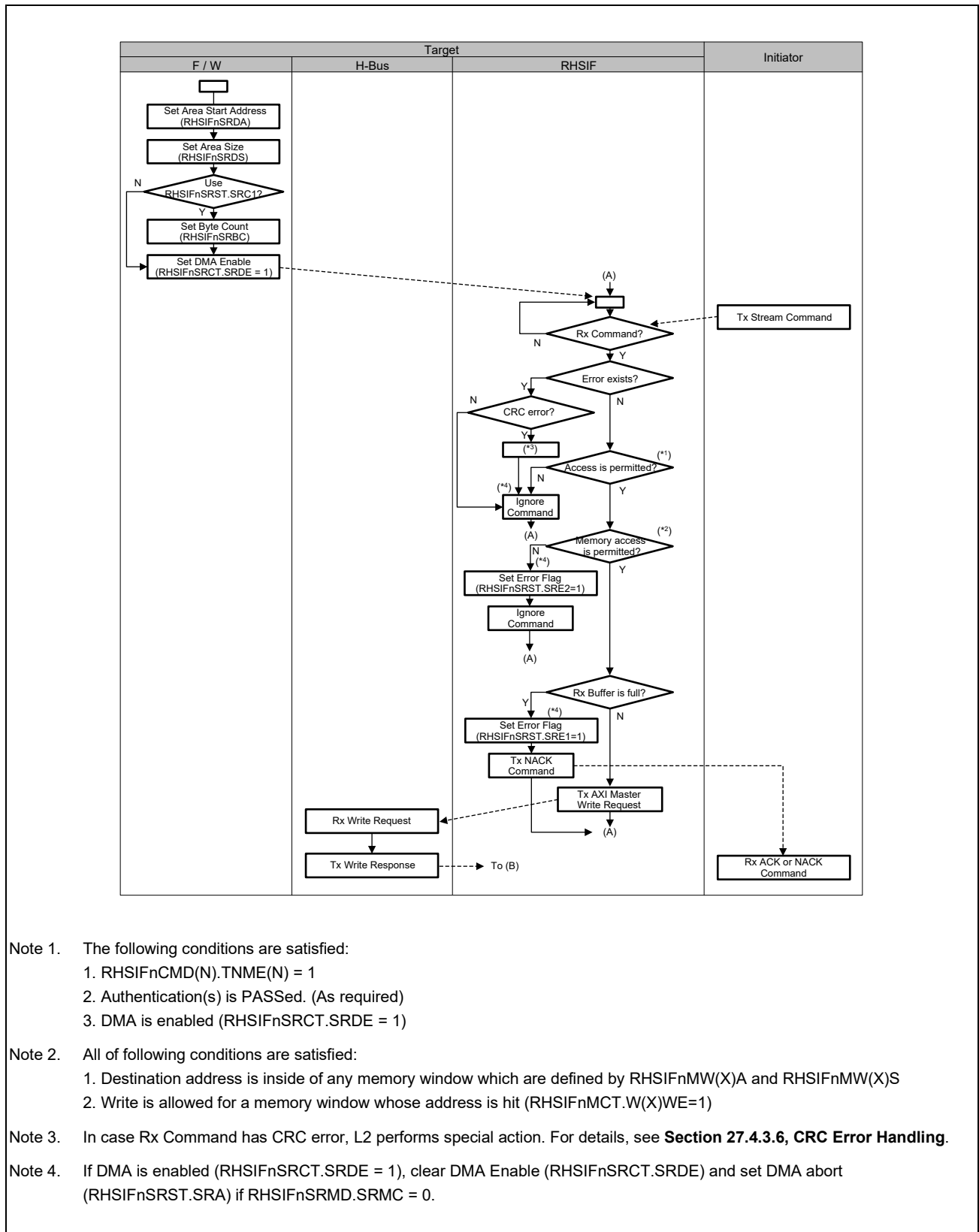
After receiving a response from the H-Bus, L2 subsequently performs one of the following operations depending on whether an error occurs in the response from the H-Bus:

- (1) When no H-Bus response error is detected
L2 updates the write pointer (RHSIFnSRWP) for the data payload size of the command that is received, and returns an ACK Command to the link partner.
- (2) When an H-Bus response error is detected
L2 updates the write pointer (RHSIFnSRWP) for the data payload size of the command that is received, returns a NACK Command to the link partner, and sets the RHSIFnSRST.SRE0 bit to 0. When the RHSIFnSRMD.SRMC

bit is 0, L2 automatically clears the RHSIFnSRCT.SRDE bit to 0 and sets the RHSIFnSRST.SRA to prevent extra data from being stored subsequently.

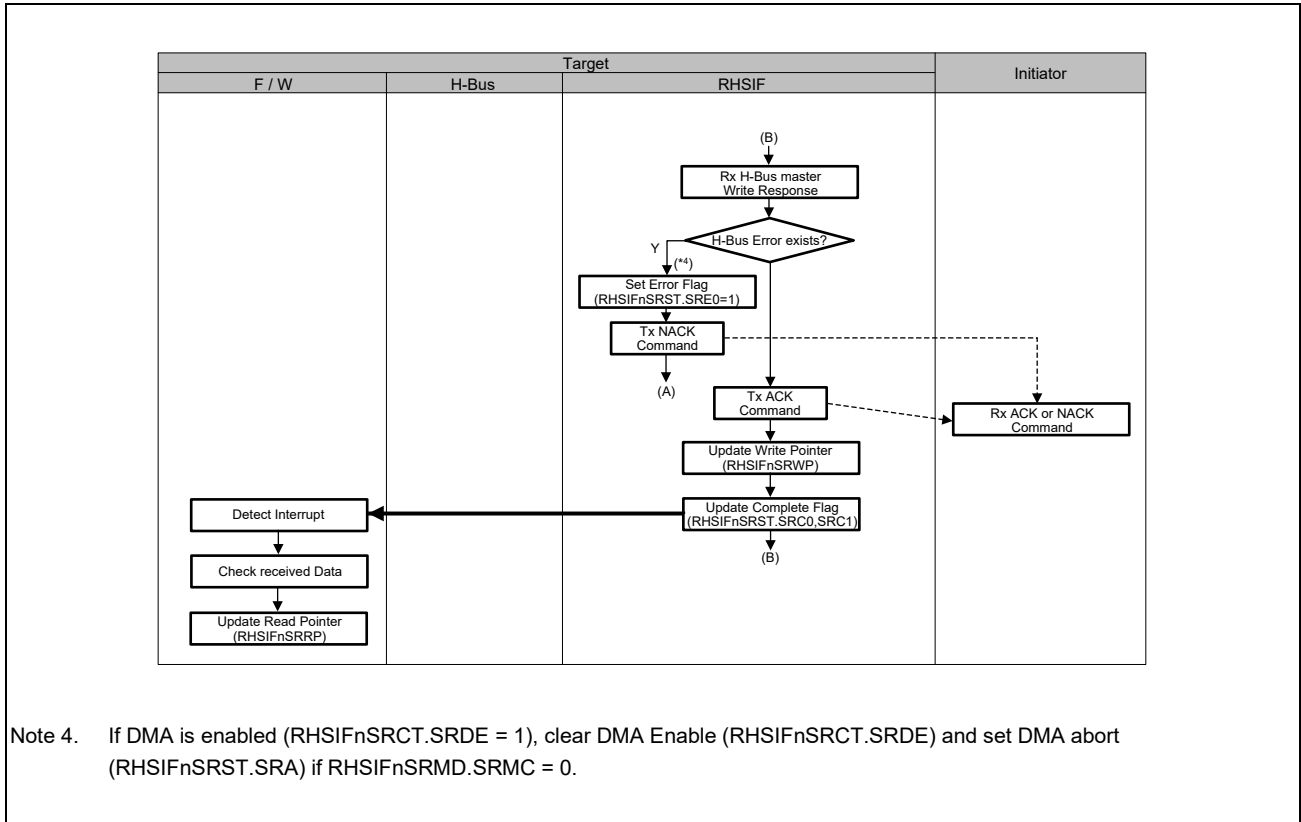
If the values of write pointer and read pointer differ from each other, an `int_hsif_str` interrupt is generated (when the RHSIFnSRIE.SRCE0 bit is 1). Also, if the size of unprocessed data reaches or exceeds the size specified in the RHSIFnSRBC register, an `int_hsif_str` interrupt is generated (when the RHSIFnSRIE.SRCE1 bit is 1).

The firmware must check the data size with the write pointer as needed, and process the data in the data storage area. After data processing ends, the firmware must update the read pointer for the size of processed data. Updating of the read pointer must always be performed in units of the size specified by the RHSIFnSRMD.SRPS bit. **Figure 27.27** shows an example of updating the pointer.



- Note 1. The following conditions are satisfied:
1. RHSIFnCMD(N).TNME(N) = 1
 2. Authentication(s) is PASSEd. (As required)
 3. DMA is enabled (RHSIFnSRCT.SRDE = 1)
- Note 2. All of following conditions are satisfied:
1. Destination address is inside of any memory window which are defined by RHSIFnMW(X)A and RHSIFnMW(X)S
 2. Write is allowed for a memory window whose address is hit (RHSIFnMCT.W(X)WE=1)
- Note 3. In case Rx Command has CRC error, L2 performs special action. For details, see **Section 27.4.3.6, CRC Error Handling**.
- Note 4. If DMA is enabled (RHSIFnSRCT.SRDE = 1), clear DMA Enable (RHSIFnSRCT.SRDE) and set DMA abort (RHSIFnSRMD.SRMC = 0).

Figure 27.25 Rx Stream Command Operation (1)



Note 4. If DMA is enabled (RHSIFnSRCT.SRDE = 1), clear DMA Enable (RHSIFnSRCT.SRDE) and set DMA abort (RHSIFnSRST.SRA) if RHSIFnSRMD.SRMC = 0.

Figure 27.26 Rx Stream Command Operation (2)

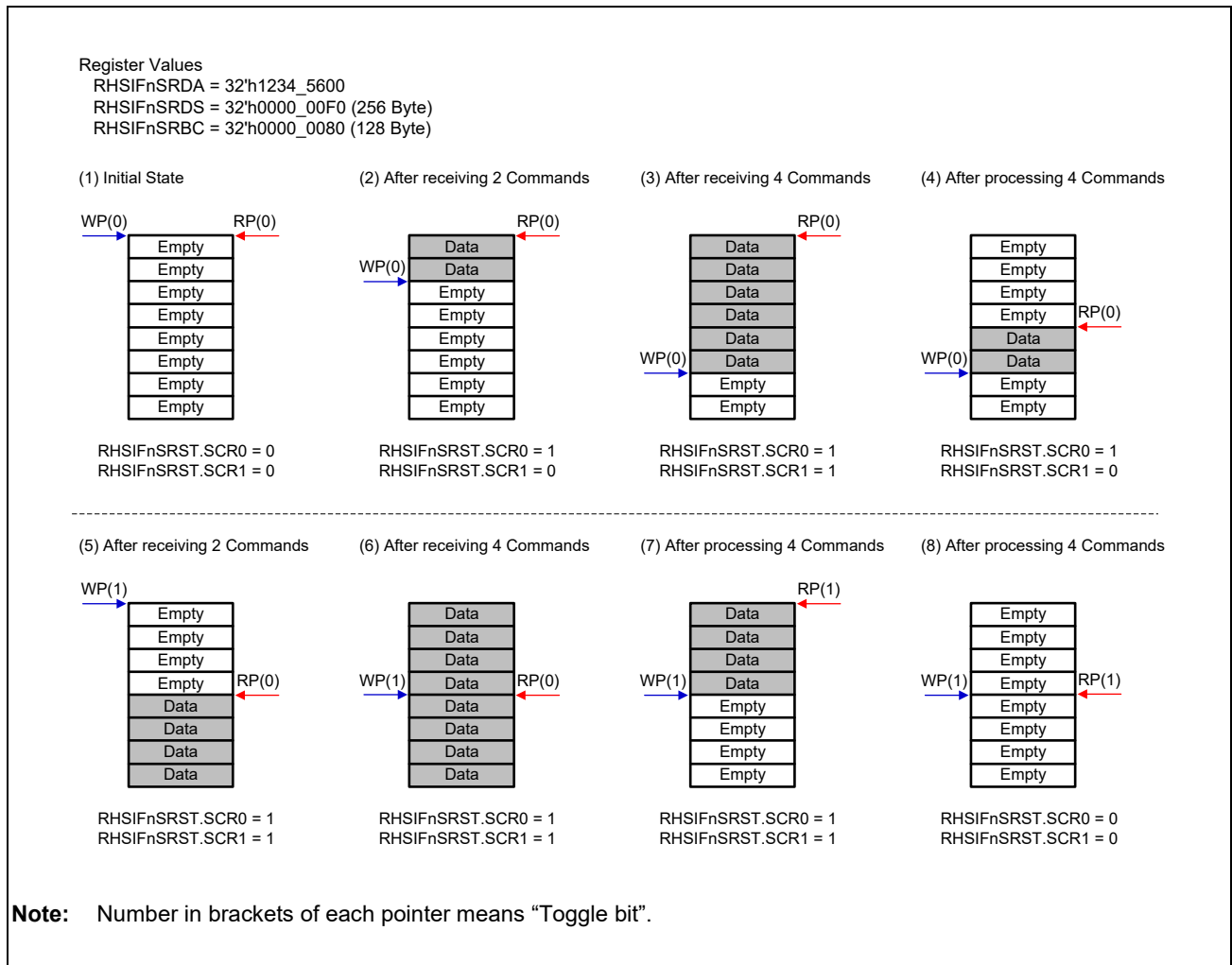


Figure 27.27 Rx Stream Pointer Operation Example

If an error occurs during Stream Command reception when the RHSIFnSRMD.SRMC bit is 0, L2 automatically clears the RHSIFnSRCT.SRDE bit and stops receiving any more Commands. (For detailed conditions, see the description of the RHSIFnSRCT.SRDE bit.)

If reception of Stream Commands is to be resumed in this status, the following procedure must be followed:

- (1) Detect interrupt by the RHSIFnSRST.SRA bit. (If RHSIFnSRIE.SRAE bit is 1.)
- (2) Check that the RHSIFnSRCT.SRDE bit is 0.
- (3) Wait until L2 can check that the RHSIFnSRST.SRTA bit is 0.
- (4) Reset the RHSIFnSRCT.SRDE bit to 1.

This procedure also applies when the firmware has cleared the RHSIFnSRCT.SRDE bit to 0 because of the system. (In this case, RHSIFnSRST.SRA is not set.)

27.4.3.6 CRC Error Handling

If a CRC error is detected in a command that is received, the header information also cannot be trusted, and, therefore, the command that is received cannot be determined to be a request or a reply. For this reason, the operation of L2 differs from those of other errors.

Upon receiving a Command involving a CRC error, L2 performs the following operations:

- Sets the RHSIFnCST(N).CRE(N) bit
- Discards the command that is received
- Clears the RHSIFnSRCT.SRDE bit when the error has occurred in channel 2 and the RHSIFnSRMD.SRMC bit is 0

Based on the above operations, necessary processing must be executed when a CRC error is detected.

27.4.4 Interrupt Specification

27.4.4.1 Interrupt Signals and Corresponding Interrupt Sources

Table 27.57 shows a list of interrupt signals, interrupt status register fields, interrupt source register fields and interrupt enable register fields. The interrupt signal is output as long as the corresponding interrupt enable bit is set to 1 and the corresponding status/flag bit is set to 1.

All interrupt signals are synchronized to ACLK, level signal, and active high.

Table 27.57 Interrupt Signal List (1/2)

Signal	Status	Source	Enable	Description
int_hsif_ch0	RHSIFnMIST.RARS0	RHSIFnCST0.RAR0	RHSIFnCIE0.RARE0	Read Answer received at Ch.0
	RHSIFnMIST.AKRS0	RHSIFnCST0.AKR0	RHSIFnCIE0.AKRE0	ACK received at Ch.0
	RHSIFnMIST.TERS0	RHSIFnCST0.TER0	RHSIFnCIE0.TERE0	Event received at Ch.0
int_hsif_ch1	RHSIFnMIST.RARS1	RHSIFnCST1.RAR1	RHSIFnCIE1.RARE1	Read Answer received at Ch.1
	RHSIFnMIST.AKRS1	RHSIFnCST1.AKR1	RHSIFnCIE1.AKRE1	ACK received at Ch.1
	RHSIFnMIST.TERS1	RHSIFnCST1.TER1	RHSIFnCIE1.TERE1	Event received at Ch.1
int_hsif_ch2	RHSIFnMIST.RARS2	RHSIFnCST2.RAR2	RHSIFnCIE2.RARE2	Read Answer received at Ch.2
	RHSIFnMIST.AKRS2	RHSIFnCST2.AKR2	RHSIFnCIE2.AKRE2	ACK received at Ch.2
	RHSIFnMIST.TERS2	RHSIFnCST2.TER2	RHSIFnCIE2.TERE2	Event received at Ch.2
int_hsif_ch3	RHSIFnMIST.RARS3	RHSIFnCST3.RAR3	RHSIFnCIE3.RARE3	Read Answer received at Ch.3
	RHSIFnMIST.AKRS3	RHSIFnCST3.AKR3	RHSIFnCIE3.AKRE3	ACK received at Ch.3
	RHSIFnMIST.TERS3	RHSIFnCST3.TER3	RHSIFnCIE3.TERE3	Event received at Ch.3
int_hsif_str	RHSIFnMIST.STCS	RHSIFnSTST.STC	RHSIFnSTIE.STCE	Tx stream completed
	RHSIFnMIST.SRCS0	RHSIFnSRST.SRC0	RHSIFnSRIE.SRCE0	Rx stream completed (more than 1 Stream Command data size)
	RHSIFnMIST.SRCS1	RHSIFnSRST.SRC1	RHSIFnSRIE.SRCE1	Rx stream completed (more than RHSIFnSRBC)
int_hsif_err	RHSIFnMIST.CERS(N)	RHSIFnCST0.AKE(N)	RHSIFnCIE0.AKEE(N)	NACK received at Ch.(N)
		RHSIFnCST0.TOE(N)	RHSIFnCIE0.TOEE(N)	Reply timeout detected at Ch.(N)
		RHSIFnCST0.IDE(N)	RHSIFnCIE0.IDEE(N)	Transaction-ID error detected at Ch.(N)
		RHSIFnCST0.AOE(N)	RHSIFnCIE0.AOEE(N)	Any other error detected at Ch.(N)
		RHSIFnCST0.BRE(N)	RHSIFnCIE0.BREE(N)	Bus error detected at Ch.(N)
		RHSIFnCST0.CRE(N)	RHSIFnCIE0.CREE(N)	CRC error detected at Ch.(N)
	RHSIFnMIST.STES	RHSIFnSTST.STE0	RHSIFnSTIE.STEE0	NACK received at Tx stream Ch.
		RHSIFnSTST.STE1	RHSIFnSTIE.STEE1	Reply timeout detected at Tx stream Ch
		RHSIFnSTST.STE2	RHSIFnSTIE.STEE2	Transaction-ID error detected at Tx stream Ch.
		RHSIFnSTST.STE3	RHSIFnSTIE.STEE3	Bus error detected at Tx stream Ch.
		RHSIFnSTST.STE4	RHSIFnSTIE.STEE4	Window mis-hit detected at Tx stream Ch.
	RHSIFnMIST.SRES	RHSIFnSRST.SRE0	RHSIFnSRIE.SREE0	Bus error detected at Rx stream Ch.
		RHSIFnSRST.SRE1	RHSIFnSRIE.SREE1	Buffer overflow detected at Rx stream Ch.
		RHSIFnSRST.SRE2	RHSIFnSRIE.SREE2	Window mis-hit detected at Rx stream Ch.
		RHSIFnSRST.SRA	RHSIFnSRIE.SRAE	Rx stream aborted by error.

Table 27.57 Interrupt Signal List (2/2)

Signal	Status	Source	Enable	Description
int_hsif_sec	RHSIFnMIST.AESS	RHSIFnAEST.AEE(N)	RHSIFnAEIE.AEEE(N)	Event received at Ch.(N) before authentication is completed.
		RHSIFnAEST.AEI(N)	RHSIFnAEIE.AEIE(N)	ID received at Ch.(N) before authentication is completed.
		RHSIFnAEST.AER(N)	RHSIFnAEIE.AERE(N)	Read received at Ch.(N) before authentication is completed.
		RHSIFnAEST.AEW(N)	RHSIFnAEIE.AEWE(N)	Write received at Ch.(N) before authentication is completed.
		RHSIFnAEST.AES	RHSIFnAEIE.AESE	Stream received at Ch.2 before authentication is completed.
		RHSIFnAEST.WER(N)	RHSIFnAEIE.WERE(N)	Received Read at Ch.(N) does not hit any memory window.
		RHSIFnAEST.WEW(N)	RHSIFnAEIE.WEWE(N)	Received Write at Ch.(N) does not hit any memory window.
		RHSIFnAEST.WES0	RHSIFnAEIE.WESE0	H-Bus Read for Stream does not hit any window.
		RHSIFnAEST.WES1	RHSIFnAEIE.WESE1	H-Bus Write for Stream does not hit any window.

Figure 27.28 shows the logic for interrupt generation. The corresponding bit in the source register/field is ANDed with the corresponding enable bit in the interrupt enable register/field. Moreover, respective factors are ORed to an interrupt signal.

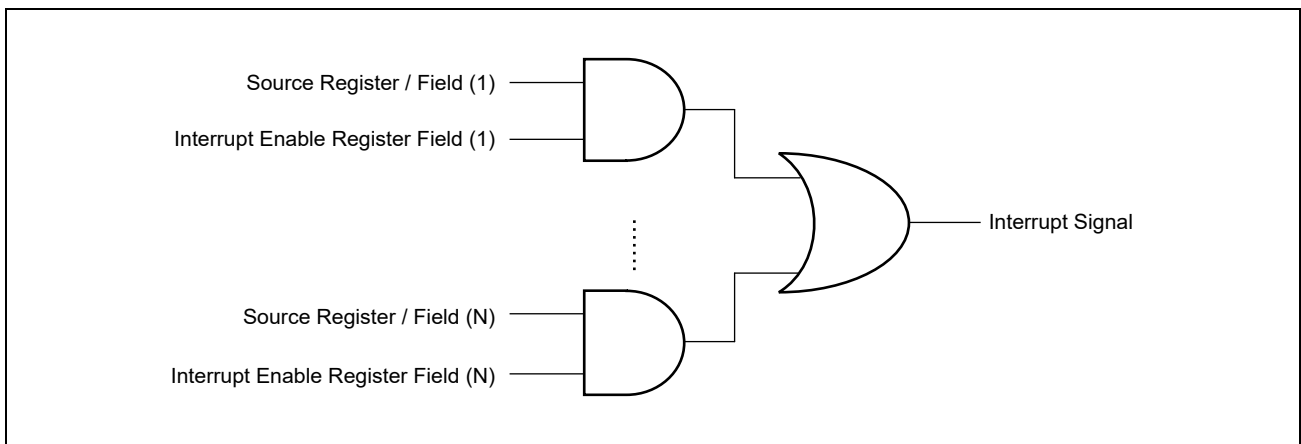


Figure 27.28 Interrupt Signal Generation Logic

27.4.4.2 Interrupt Signal Status Register

L2 has a register to monitor the interrupt factor state. This register (RHSIFnMIST) indicates the current state of the interrupt factors. Whether an interrupt is active can be checked by a single register read.

27.5 Datalink Layer (L1)

27.5.1 Overview

Regarding the basic functions and specifications, refer to the protocol specifications.

27.5.1.1 Functional Overview

The Renesas High-speed Serial Interface Datalink Layer IP (henceforth, called “L1”) is compliant with the following specifications.

(1) L1

- Addition of L1 header information to the L2 frame for data transmission
- Removal of the L1 header from the receive data and transmission to L2
- Generation and transmission of Interface Control Logical Channel (hereafter, ICLC) commands
- Reception and execution of ICLC commands
- Generation of interrupt sources for transmission completion, transmission error, reception completion, reception error, and ICLC command reception
- Generation and transmission of CTS commands
- Frame arbitration: Fixed (Higher) PingAnswer → CTS Command → ICLC Command → L2 Frame (lower)
- Execution of flow control
- Execution of Loopback test mode
- Execution of toggle pattern test mode

(2) Physical Interface

This layer consists of LVDS differential interface (input and output) and the reference clock. Regarding the basic functions and specifications, refer to the protocol specification.

Table 27.58 Physical Interface

External Port	Specification Item	Specification
Clock (RHSIFnREFCLK)	Electrical Characteristics	TTL / CMOS
	Frequency	20 MHz, 10 MHz
Data (RHSIFnTXDP, RHSIFnTXDN, RHSIFnRXDP, RHSIFnRXDN)	Electrical Characteristics	LVDS (based on IEEE1596.3-1996 reducedrangelink)
	BaudRate (FastMode) Using PLL	160 M Baud, 80 M Baud selectable
	BaudRate (SlowMode) Using REFCLK	5 M Baud

27.5.1.2 Block Diagram

(1) L1

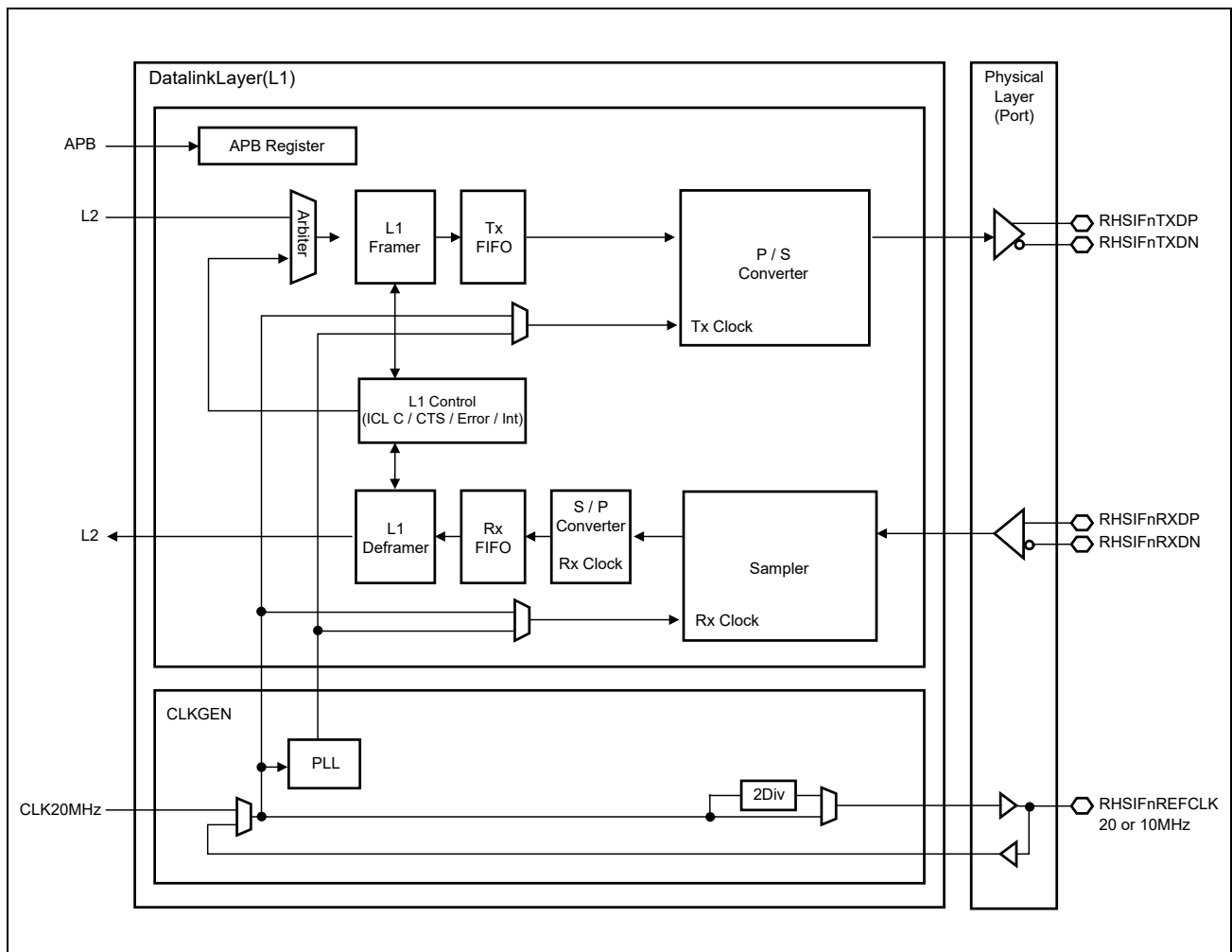


Figure 27.29 Block Diagram of L1 and Physical Interface

27.5.2 Registers

This section describes the L1 registers in detail.

In case of a concurrent update of the same register bit, L1 module write (hardware write) has higher priority than CPU write (software write).

27.5.2.1 List of Registers

The register map of an L1 module is shown in **Table 27.59**.

The L1 module remains in reset while RESET.

The values shown in **Table 27.59** in the L1 registers description under “Values after Reset” are related to this reset state.

For further information about the L1 operation modes, refer to **Section 27.5.3, Operation**.

The module specific base address of an L1 module needs to be added to each of the specified offset addresses.

Table 27.59 Registers Address Map

Module Name	Register Name	Symbol	Address	Access	Access Protection
RHSIFn	Mode Control Register	RHSIFnMDCR	<RHSIFn_L1_base> + 000 _H	32	-
	Sleep Mode Control Register	RHSIFnSMCR	<RHSIFn_L1_base> + 004 _H	32	-
	PLL Control Register	RHSIFnPCR	<RHSIFn_L1_base> + 008 _H	32	-
	Speed Control Register	RHSIFnSPCR	<RHSIFn_L1_base> + 00C _H	32	-
	Test Mode Control Register	RHSIFnTMDCR	<RHSIFn_L1_base> + 010 _H	32	-
	L1 Status Register	RHSIFnL1SR	<RHSIFn_L1_base> + 018 _H	32	-
	Last Tx Frame L1 Result Register	RHSIFnLTXFRMRL1	<RHSIFn_L1_base> + 030 _H	32	-
	Last Rx Frame L1 Result Register	RHSIFnLRXFRMRL1	<RHSIFn_L1_base> + 034 _H	32	-
	Last Tx Frame L2 Result Register	RHSIFnLTXFRMRL2	<RHSIFn_L1_base> + 038 _H	32	-
	Last Rx Frame L2 Result Register	RHSIFnLRXFRMRL2	<RHSIFn_L1_base> + 03C _H	32	-
	TX/RX Control Register	RHSIFnTXRXCR	<RHSIFn_L1_base> + 050 _H	32	-
	ICLC Command Control Register	RHSIFnCCR	<RHSIFn_L1_base> + 060 _H	32	-
	CTS Frame Control Register	RHSIFnCCR	<RHSIFn_L1_base> + 070 _H	32	-
	Tx Complete Status Register	RHSIFnTXCMPST	<RHSIFn_L1_base> + 080 _H	32	-
	Tx Complete Status Clear Register	RHSIFnTXCMPSC	<RHSIFn_L1_base> + 084 _H	32	-
	Tx Complete Interrupt Enable Register	RHSIFnTXCMPIE	<RHSIFn_L1_base> + 088 _H	32	-
	Tx Error Status Register	RHSIFnTXERRST	<RHSIFn_L1_base> + 090 _H	32	-
	Tx Error Status Clear Register	RHSIFnTXERRSC	<RHSIFn_L1_base> + 094 _H	32	-
	Tx Error Interrupt Enable Register	RHSIFnTXERRIE	<RHSIFn_L1_base> + 098 _H	32	-
	Rx Complete Status Register	RHSIFnRXCMPST	<RHSIFn_L1_base> + 0A0 _H	32	-
	Rx Complete Status Clear Register	RHSIFnRXCMPSC	<RHSIFn_L1_base> + 0A4 _H	32	-
	Rx Complete Interrupt Enable Register	RHSIFnRXCMPIE	<RHSIFn_L1_base> + 0A8 _H	32	-
	Rx Error Status Register	RHSIFnRXERRST	<RHSIFn_L1_base> + 0B0 _H	32	-
	Rx Error Status Clear Register	RHSIFnRXERRSC	<RHSIFn_L1_base> + 0B4 _H	32	-
	Rx Error Interrupt Enable Register	RHSIFnRXERRIE	<RHSIFn_L1_base> + 0B8 _H	32	-
	Rx ICLC Command Status Register	RHSIFnRXICST	<RHSIFn_L1_base> + 0C0 _H	32	-
	Rx ICLC Command Status Clear Register	RHSIFnRXICSC	<RHSIFn_L1_base> + 0C4 _H	32	-
	Rx ICLC Command Interrupt Enable Register	RHSIFnRXICIE	<RHSIFn_L1_base> + 0C8 _H	32	-

27.5.2.2 Legend

This section explains the module state dependent abbreviations used for the L1 registers description.

Conditions:

- R/W: Bit is readable and writable
- R: Read-only bit; the user cannot write to this bit
- W: Write-only bit; Read value is always “0”

Reserved bits:

- Always read as 0
- Writing to these bits has no effect (hardware protection)

27.5.2.3 Detailed Register Description

Detailed specifications of each register are described below.

(1) RHSIFnMDCR — Mode Control Register

Value after reset: 0001 0100_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CTSEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CTSV	—	—	—	—	—	—	CLKSEL	MST
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W

Table 27.60 RHSIFnMDCR Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	CTSEN	This bit specifies whether to enable the flow control. 0: Disable automatic flow control. 1: Enable automatic flow control. Set this bit to "1" to enable flow control and stop next frame transfer. See Section 27.5.3.5, CTS Commands .
15 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	CTSV	This bit specifies CTS Value of L1 header for all frames. 0: Frame unreceivable. 1: Frame receivable.
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CLKSEL	This bit specifies REFCLK frequency. 0: 20MHz. 1: 10MHz.
0	MST	This bit specifies whether Master or Slave. 0: Slave. 1: Master.

(2) RHSIFnSMCR — Sleep Mode Control RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SLP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 27.61 RHSIFnSMCR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SLP	Specify whether to set the sleep mode during data transmission or in the next transmission frame. 0: L1 End bit = 0 (Normal mode for the link partner) 1: L1 End bit = 1 (Transitioning to sleep mode for the link partner)

(3) RHSIFnPCR — PLL Control RegisterValue after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLSTBY
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 27.62 RHSIFnPCR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	PLLSTBY	This bit specifies whether PLL Standby or PLL Active. This bit is updated only when writing directly. It is not updated when ICLC (Slave interface PLL start, Slave interface PLL stop) command reception. 0: PLL Active. 1: PLL Standby.

(4) RHSIFnSPCR — Speed Control RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	FMBR[1:0]		—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TXSP	—	—	—	—	—	—	—	RXSP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 27.63 RHSIFnSPCR Register Contents

Bit Position	Bit Name	Function
31 to 26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25 to 24	FMBR1	These bits specify Fast Speed Mode Baud Rate. 00: 80MBaud 01: 160MBaud 10: Setting prohibited 11: Setting prohibited
23 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	TXSP*1	This bit specifies whether Slow Speed Mode or Fast Speed Mode for data transmission. This bit is updated only when writing directly. It is not updated when ICLC (Select Slow Speed mode for transfers from the Slave interface to the Master interface, Select Fast Speed mode for transfers from the Slave interface to the Master interface) command reception. 0: Slow Speed Mode. 1: Fast Speed Mode.
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	RXSP*1	This bit specifies whether Slow Speed Mode or Fast Speed Mode for data reception. This bit is updated only when writing directly. It is not updated when ICLC (Select Slow Speed mode for transfers from the Master interface to the Slave interface, Select Fast Speed mode for transfers from the Master interface to the Slave interface) command reception. 0: Slow Speed Mode. 1: Fast Speed Mode.

Note 1. In case of setting '1' to RHSIFnSPCR.TXSP or RHSIFnSPCR.RXSP, please set '0' to RHSIFnPCR.PLLSTBY for activating PLL.

(5) RHSIFnTMDCR — Test Mode Control Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CKTM	CKTMS EL	—	—	—	—	—	—	TXLPB K	RXLPB K
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 27.64 RHSIFnTMDCR Register Contents

Bit Position	Bit Name	Function
31 to 10	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
9	CKTM* ¹	This bit is specified for selecting clock test mode. This bit is updated when ICLC command is for “Turn on testmode” or “Turn off test mode” as well as writing to this bit directly. 0: Disable. 1: Clock test mode.
8	CKTMSSEL	Clock Selection of Clock testmode 0: RX BaudRate/2 Clock 1: TX BaudRate/2 Clock
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	TXLPBK* ^{1,*3}	This bit is specified for selecting TX-RX loopback testmode. 0: Disable. 1: TX-RX loopback testmode.
0	RXLPBK* ^{1,*2}	This bit is specified for selecting RX-TX loopback testmode. This bit is updated when ICLC command is for “Turn on payload loopback” or “Turn off test mode” as well as writing to this bit directly. 0: Disable. 1: RX-TX loopback testmode.

Note 1. Setting 2 bits or more to 1 at the same time is prohibited.

Note 2. In Rx-Tx loopback testmode, received ICLC Turn off command is output by looping back.

Note 3. In case of setting '1' to RHSIFnTMDCR.TXLPBK, please set the same value to RHSIFnSPCR.TXSP and RHSIFnSPCR.RXSP. Unless the same BaudRate is set to both transmitter/receiver, the communication will not be done normally.

(6) RHSIFnL1SR — L1 Status RegisterValue after reset: 0001 0101_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TFFUL	TFEMP	—	—	—	—	—	—	RFFUL	RFEMP
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.65 RHSIFnL1SR Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	RCTS	This bit indicates the CTS value that is received. 0: Linkpartner is not receivable 1: Linkpartner is receivable
15 to 10	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
9	TFFUL	This bit indicates TX FIFO full status. 0: Not full. 1: Full.
8	TFEMP	This bit indicates TX FIFO empty status. 0: Not empty. 1: Empty.
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	RFFUL	This bit indicates RX FIFO full status. 0: Not full. 1: Full.
0	RFEMP	This bit indicates RX FIFO empty status. 0: Not empty. 1: Empty.

(7) RHSIFnLTXFRMRL1 — Last Tx Frame L1 Result Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TL1E
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TL1P[7:0]							TL1H[7:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.66 RHSIFnLTXFRMRL1 Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	TL1E	This bit indicates last Tx Endbit.
15 to 8	TL1P[7:0]	These bits indicate Last Tx L1 Payload.
7 to 0	TL1H[7:0]	These bits indicate Last Tx L1 Header.

(8) RHSIFnLRXFRMRL1 — Last Rx Frame L1 Result RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RL1E
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RL1P[7:0]							RL1H[7:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.67 RHSIFnLRXFRMRL1 Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	RL1E	This bit indicates last Rx Endbit.
15 to 8	RL1P[7:0]	These bits indicate Last Rx L1 Payload.
7 to 0	RL1H[7:0]	These bits indicate Last Rx L1 Header.

(9) RHSIFnLTXFRMRL2 — Last Tx Frame L2 Result RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	T2L2H[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	T2L1H[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.68 RHSIFnLTXFRMRL2 Register Contents

Bit Position	Bit Name	Function
31 to 16	T2L2H[15:0]	These bits indicate Last Tx L2 Header of L2 frame.
15 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	T2L1H[7:0]	These bits indicate Last Tx L1 Header of L2 frame.

(10) RHSIFnLRXFRMRL2 — Last Rx Frame L2 Result RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R2L2H[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	R2L1H[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.69 RHSIFnLRXFRMRL2 Register Contents

Bit Position	Bit Name	Function
31 to 16	R2L2H[15:0]	These bits indicate Last Rx L2 Header of L2 frame.
15 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	R2L1H[7:0]	These bits indicate Last Rx L1 Header of L2 frame.

(11) RHSIFnTXRXCR — TX/RX Control RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 27.70 RHSIFnTXRXCR Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	TXEN	This bit specifies whether to enable transmission. This bit is updated when the ICLC command is for “Enable Slave interface transmitter” or “Disable Slave interface transmitter” as well as writing to this bit directly. 0: Disable. 1: Enable.
15 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	RXEN	This bit specifies whether to enable reception. 0: Disable. 1: Enable.

(12) RHSIFnICCR — ICLC Command Control RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ITRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PLD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.71 RHSIFnICCR Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	ITRG	ICLC Command Tx Trigger. This bit is cleared automatically after ICLC command transmission is completed. 0: Disable. 1: Trigger.
15 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	PLD[7:0] ^{*1}	ICLC Command (Payload)

Note 1. Please do not specify any commands except the commands described in **Table 27.12**.

(13) RHSIFnCCR — CTS Frame Control RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CTS	PLD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.72 RHSIFnCCR Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	CTRG	CTS Frame Tx Trigger. This bit is cleared automatically after CTS Frame transmission is completed. 0: Disable. 1: Trigger.
15 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	CTS	CTS Value
7 to 0	PLD[7:0]	CTS Frame (Payload)

(14) RHSIFnTXCMPST — Tx Complete Status RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TCL2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCCT	—	—	—	—	—	—	—	TCIC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.73 RHSIFnTXCMPST Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	TCL2	This bit indicates the status of L2 frame transmission. 0: L2 frame transmission not completed. 1: L2 frame transmission completed.
15 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	TCCT	This bit indicates the status of L1 CTS frame transmission. 0: L1 CTS frame transmission not completed. 1: L1 CTS frame transmission completed.
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	TCIC	This bit indicates the status of L1 ICLC command transmission. 0: L1 ICLC command transmission not completed. 1: L1 ICLC command transmission completed.

(15) RHSIFnTXCMPSC — Tx Complete Status Clear RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TCCL2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCCCT	—	—	—	—	—	—	—	TCCIC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	R	R	R	R	W

Table 27.74 RHSIFnTXCMPSC Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	TCCL2	Writing 1 to this bit clears the RHSIFnTXCMPST.TCL2 bit. Writing 0 to this bit is ignored.
15 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	TCCCT	Writing 1 to this bit clears the RHSIFnTXCMPST.TCCT bit. Writing 0 to this bit is ignored.
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	TCCIC	Writing 1 to this bit clears the RHSIFnTXCMPST.TCIC bit. Writing 0 to this bit is ignored.

(16) RHSIFnTXCMPIE — Tx Complete Interrupt Enable RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TCEL2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCECT	—	—	—	—	—	—	—	TCEIC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 27.75 RHSIFnTXCMPIE Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	TCEL2	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXCMPST.TCL2 bit.
15 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	TCECT	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXCMPST.TCCT bit.
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	TCEIC	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXCMPST.TCIC bit.

(17) RHSIFnTXERRST — Tx Error Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TERSZ	—	—	—	—	—	—	—	—	—	—	TERSZI5	TERSZI4	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TERCTF	TERCTE	TERCTD	TERCTC	TERCTB	TERCTA	TERCT9	TERCT8	—	—	—	—	—	TERCT2	TERCT1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.76 RHSIFnTXERRST Register Contents (1/2)

Bit Position	Bit Name	Function
31	TERSZ	This bit indicates the status of payload size error of transmission. 0: Payload size error has not occurred. 1: Payload size error occurred.
30 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	TERSZI5	This bit indicates that the payload size of transmission is 3'b101. 0: Payload size index is not 3'b101. 1: Payload size index is 3'b101.
19	TERSZI4	This bit indicates the payload size of transmission is 3'b100. 0: Payload size index is not 3'b100. 1: Payload size index is 3'b100.
18 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15	TERCTF	This bit indicates that the logical channel type of transmission is 4'b1111. 0: Logical channel type of transmission is not 4'b1111. 1: Logical channel type of transmission is 4'b1111.
14	TERCTE	This bit indicates that the logical channel type of transmission is 4'b1110. 0: Logical channel type of transmission is not 4'b1110. 1: Logical channel type of transmission is 4'b1110.
13	TERCTD	This bit indicates that the logical channel type of transmission is 4'b1101. 0: Logical channel type of transmission is not 4'b1101. 1: Logical channel type of transmission is 4'b1101.
12	TERCTC	This bit indicates that the logical channel type of transmission is 4'b1100. 0: Logical channel type of transmission is not 4'b1100. 1: Logical channel type of transmission is 4'b1100.
11	TERCTB	This bit indicates that the logical channel type of transmission is 4'b1011. 0: Logical channel type of transmission is not 4'b1011. 1: Logical channel type of transmission is 4'b1011.
10	TERCTA	This bit indicates that the logical channel type of transmission is 4'b1010. 0: Logical channel type of transmission is not 4'b1010. 1: Logical channel type of transmission is 4'b1010.
9	TERCT9	This bit indicates that the logical channel type of transmission is 4'b1001. 0: Logical channel type of transmission is not 4'b1001. 1: Logical channel type of transmission is 4'b1001.

Table 27.76 RHSIFnTXERRST Register Contents (2/2)

Bit Position	Bit Name	Function
8	TERCT8	This bit indicates that the logical channel type of transmission is 4'b1000. 0: Logical channel type of transmission is not 4'b1000. 1: Logical channel type of transmission is 4'b1000.
7 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	TERCT2	This bit indicates that the logical channel type of transmission is 4'b0010. 0: Logical channel type of transmission is not 4'b0010. 1: Logical channel type of transmission is 4'b0010.
1	TERCT1	This bit indicates that the logical channel type of transmission is 4'b0001. 0: Logical channel type of transmission is not 4'b0001. 1: Logical channel type of transmission is 4'b0001.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

(18) RHSIFnTXERRSC — Tx Error Status Clear Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TERCSZ	—	—	—	—	—	—	—	—	—	—	TERCSZI5	TERCSZI4	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	R	R	R	R	R	R	R	R	R	R	W	W	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TERCCTF	TERCCTE	TERCCTD	TERCCTC	TERCCTB	TERCCTA	TERCCT9	TERCCT8	—	—	—	—	—	TERCCT2	TERCCT1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	R	R	R	R	R	W	W	R

Table 27.77 RHSIFnTXERRSC Register Contents

Bit Position	Bit Name	Function
31	TERCSZ	Writing 1 to this bit clears the RHSIFnTXERRST.TERSZ bit. Writing 0 to this bit is ignored.
30 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	TERCSZI5	Writing 1 to this bit clears the RHSIFnTXERRST.TERSZI5 bit. Writing 0 to this bit is ignored.
19	TERCSZI4	Writing 1 to this bit clears the RHSIFnTXERRST.TERSZI4 bit. Writing 0 to this bit is ignored.
18 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15	TERCCTF	Writing 1 to this bit clears the RHSIFnTXERRST.TERCCTF bit. Writing 0 to this bit is ignored.
14	TERCCTE	Writing 1 to this bit clears the RHSIFnTXERRST.TERCCTE bit. Writing 0 to this bit is ignored.
13	TERCCTD	Writing 1 to this bit clears the RHSIFnTXERRST.TERCCTD bit. Writing 0 to this bit is ignored.
12	TERCCTC	Writing 1 to this bit clears the RHSIFnTXERRST.TERCCTC bit. Writing 0 to this bit is ignored.
11	TERCCTB	Writing 1 to this bit clears the RHSIFnTXERRST.TERCCTB bit. Writing 0 to this bit is ignored.
10	TERCCTA	Writing 1 to this bit clears the RHSIFnTXERRST.TERCCTA bit. Writing 0 to this bit is ignored.
9	TERCCT9	Writing 1 to this bit clears the RHSIFnTXERRST.TERCCT9 bit. Writing 0 to this bit is ignored.
8	TERCCT8	Writing 1 to this bit clears the RHSIFnTXERRST.TERCCT8 bit. Writing 0 to this bit is ignored.
7 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	TERCCT2	Writing 1 to this bit clears the RHSIFnTXERRST.TERCCT2 bit. Writing 0 to this bit is ignored.
1	TERCCT1	Writing 1 to this bit clears the RHSIFnTXERRST.TERCCT1 bit. Writing 0 to this bit is ignored.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

(19) RHSIFnTXERRIE — Tx Error Interrupt Enable RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TERESZ	—	—	—	—	—	—	—	—	—	—	TERESZ15	TERESZ14	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TERECTF	TERECTE	TERECTD	TERECTC	TERECTB	TERECTA	TERECT9	TERECT8	—	—	—	—	—	TERECT2	TERECT1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R

Table 27.78 RHSIFnTXERRIE Register Contents (1/2)

Bit Position	Bit Name	Function
31	TERESZ	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERSZ bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERSZ bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERSZ bit is 1.
30 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	TERESZ15	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERSZ15 bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERSZ15 bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERSZ15 bit is 1.
19	TERESZ14	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERSZ14 bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERSZ14 bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERSZ14 bit is 1.
18 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15	TERECTF	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCTF bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTF bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTF bit is 1.
14	TERECTE	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCTE bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTE bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTE bit is 1.
13	TERECTD	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCTD bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTD bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTD bit is 1.
12	TERECTC	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCTC bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTC bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTC bit is 1.
11	TERECTB	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCTB bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTB bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTB bit is 1.

Table 27.78 RHSIFnTXERRIE Register Contents (2/2)

Bit Position	Bit Name	Function
10	TERECTA	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCTA bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTA bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTA bit is 1.
9	TERECT9	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCT9 bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT9 bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT9 bit is 1.
8	TERECT8	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCT8 bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT8 bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT8 bit is 1.
7 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	TERECT2	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCT2 bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT2 bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT2 bit is 1.
1	TERECT1	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCT1 bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT1 bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT1 bit is 1.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

(20) RHSIFnRXCMPST — Rx Complete Status RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCL2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RCCT	—	—	—	—	—	—	—	RCIC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.79 RHSIFnRXCMPST Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	RCL2	This bit indicates the status of L2 frame reception. 0: L2 frame reception not completed. 1: L2 frame reception completed.
15 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	RCCT	This bit indicates the status of L1 CTS frame reception. 0: L1 CTS frame reception not completed. 1: L1 CTS frame reception completed.
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	RCIC	This bit indicates the status of L1 ICLC command reception. 0: L1 ICLC command reception not completed. 1: L1 ICLC command reception completed.

(21) RHSIFnRXCMPSC — Rx Complete Status Clear RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCCL2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RCCCT	—	—	—	—	—	—	—	RCCIC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	R	R	R	R	W

Table 27.80 RHSIFnRXCMPSC Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	RCCL2	Writing 1 to this bit clears the RHSIFnRXCMPST.RCL2 bit. Writing 0 to this bit is ignored.
15 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	RCCCT	Writing 1 to this bit clears the RHSIFnRXCMPST.RCCT bit. Writing 0 to this bit is ignored.
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	RCCIC	Writing 1 to this bit clears the RHSIFnRXCMPST.RCIC bit. Writing 0 to this bit is ignored.

(22) RHSIFnRXCMPIE — Rx Complete Interrupt Enable RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCEL2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RCECT	—	—	—	—	—	—	—	RCEIC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 27.81 RHSIFnRXCMPIE Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	RCEL2	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXCMPST.RCL2 bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXCMPST.RCL2 bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXCMPST.RCL2 bit is 1.
15 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	RCECT	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXCMPST.RCCT bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXCMPST.RCCT bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXCMPST.RCCT bit is 1.
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	RCEIC	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXCMPST.RCIC bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXCMPST.RCIC bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXCMPST.RCIC bit is 1.

(23) RHSIFnRXERRST — Rx Error Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RERSZ	—	—	—	—	—	—	—	—	—	—	RERSZI5	RERSZI4	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RERCTF	RERCTE	RERCTD	RERCTC	RERCTB	RERCTA	RERCT9	RERCT8	—	—	—	—	—	RERCT2	RERCT1	RERIPV
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.82 RHSIFnRXERRST Register Contents (1/2)

Bit Position	Bit Name	Function
31	RERSZ	This bit indicates the status of payload size error of reception. 0: Payload size error not occurred. 1: Payload size error occurred.
30 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	RERSZI5	This bit indicates that the payload size of reception is 3'b101. 0: Payload size index is not 3'b101. 1: Payload size index is 3'b101.
19	RERSZI4	This bit indicates that the payload size of reception is 3'b100. 0: Payload size index is not 3'b100. 1: Payload size index is 3'b100.
18 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15	RERCTF	This bit indicates that the logical channel type of reception is 4'b1111. 0: Logical channel type of reception is not 4'b1111. 1: Logical channel type of reception is 4'b1111.
14	RERCTE	This bit indicates that the logical channel type of reception is 4'b1110. 0: Logical channel type of reception is not 4'b1110. 1: Logical channel type of reception is 4'b1110.
13	RERCTD	This bit indicates that the logical channel type of reception is 4'b1101. 0: Logical channel type of reception is not 4'b1101. 1: Logical channel type of reception is 4'b1101.
12	RERCTC	This bit indicates that the logical channel type of reception is 4'b1100. 0: Logical channel type of reception is not 4'b1100. 1: Logical channel type of reception is 4'b1100.
11	RERCTB	This bit indicates that the logical channel type of reception is 4'b1011. 0: Logical channel type of reception is not 4'b1011. 1: Logical channel type of reception is 4'b1011.
10	RERCTA	This bit indicates that the logical channel type of reception is 4'b1010. 0: Logical channel type of reception is not 4'b1010. 1: Logical channel type of reception is 4'b1010.
9	RERCT9	This bit indicates that the logical channel type of reception is 4'b1001. 0: Logical channel type of reception is not 4'b1001. 1: Logical channel type of reception is 4'b1001.

Table 27.82 RHSIFnRXERRST Register Contents (2/2)

Bit Position	Bit Name	Function
8	RERCT8	This bit indicates that the logical channel type of reception is 4'b1000. 0: Logical channel type of reception is not 4'b1000. 1: Logical channel type of reception is 4'b1000.
7 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	RERCT2	This bit indicates that the logical channel type of reception is 4'b0010. 0: Logical channel type of reception is not 4'b0010. 1: Logical channel type of reception is 4'b0010.
1	RERCT1	This bit indicates that the logical channel type of reception is 4'b0001. 0: Logical channel type of reception is not 4'b0001. 1: Logical channel type of reception is 4'b0001.
0	RERIPV	This bit indicates whether or not the command undefined in Table 27.12 is received. 0: Command defined in the table is received. 1: Command undefined in the table is received.

(24) RHSIFnRXERRSC — Rx Error Status Clear RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RERCSZ	—	—	—	—	—	—	—	—	—	—	RERCSZI5	RERCSZI4	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	R	R	R	R	R	R	R	R	R	R	W	W	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RERCCTF	RERCCTE	RERCCTD	RERCCTC	RERCCTB	RERCCTA	RERCCT9	RERCCT8	—	—	—	—	—	RERCCT2	RERCCT1	RERICPV
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	R	R	R	R	R	W	W	W

Table 27.83 RHSIFnRXERRSC Register Contents

Bit Position	Bit Name	Function
31	RERCSZ	Writing 1 to this bit clears the RHSIFnRXERRST.RERSZ bit. Writing 0 to this bit is ignored.
30 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	RERCSZI5	Writing 1 to this bit clears the RHSIFnRXERRST.RERSZI5 bit. Writing 0 to this bit is ignored.
19	RERCSZI4	Writing 1 to this bit clears the RHSIFnRXERRST.RERSZI4 bit. Writing 0 to this bit is ignored.
18 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15	RERCCTF	Writing 1 to this bit clears the RHSIFnRXERRST.RERCCTF bit. Writing 0 to this bit is ignored.
14	RERCCTE	Writing 1 to this bit clears the RHSIFnRXERRST.RERCCTE bit. Writing 0 to this bit is ignored.
13	RERCCTD	Writing 1 to this bit clears the RHSIFnRXERRST.RERCCTD bit. Writing 0 to this bit is ignored.
12	RERCCTC	Writing 1 to this bit clears the RHSIFnRXERRST.RERCCTC bit. Writing 0 to this bit is ignored.
11	RERCCTB	Writing 1 to this bit clears the RHSIFnRXERRST.RERCCTB bit. Writing 0 to this bit is ignored.
10	RERCCTA	Writing 1 to this bit clears the RHSIFnRXERRST.RERCCTA bit. Writing 0 to this bit is ignored.
9	RERCCT9	Writing 1 to this bit clears the RHSIFnRXERRST.RERCCT9 bit. Writing 0 to this bit is ignored.
8	RERCCT8	Writing 1 to this bit clears the RHSIFnRXERRST.RERCCT8 bit. Writing 0 to this bit is ignored.
7 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	RERCCT2	Writing 1 to this bit clears the RHSIFnRXERRST.RERCCT2 bit. Writing 0 to this bit is ignored.
1	RERCCT1	Writing 1 to this bit clears the RHSIFnRXERRST.RERCCT1 bit. Writing 0 to this bit is ignored.
0	RERICPV	Writing 1 to this bit clears the RHSIFnRXERRST.RERICPV bit. Writing 0 to this bit is ignored.

(25) RHSIFnRXERRIE — Rx Error Interrupt Enable Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RERESZ	—	—	—	—	—	—	—	—	—	—	RERESZI5	RERESZI4	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RERECTF	RERECTE	RERECTD	RERECTC	RERECTB	RERECTA	RERECT9	RERECT8	—	—	—	—	—	RERECT2	RERECT1	REREIPV
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 27.84 RHSIFnRXERRIE Register Contents (1/2)

Bit Position	Bit Name	Function
31	RERESZ	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERSZ bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERSZ bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERSZ bit is 1.
30 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	RERESZI5	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERSZI5 bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERSZI5 bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERSZI5 bit is 1.
19	RERESZI4	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERSZI4 bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERSZI4 bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERSZI4 bit is 1.
18 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15	RERECTF	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERCTF bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTF bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTF bit is 1.
14	RERECTE	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERCTE bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTE bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTE bit is 1.
13	RERECTD	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERCTD bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTD bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTD bit is 1.
12	RERECTC	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERCTC bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTC bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTC bit is 1.
11	RERECTB	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERCTB bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTB bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTB bit is 1.

Table 27.84 RHSIFnRXERRIE Register Contents (2/2)

Bit Position	Bit Name	Function
10	RRECTA	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RRECTA bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RRECTA bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RRECTA bit is 1.
9	RRECT9	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RRECT9 bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RRECT9 bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RRECT9 bit is 1.
8	RRECT8	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RRECT8 bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RRECT8 bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RRECT8 bit is 1.
7 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	RRECT2	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RRECT2 bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RRECT2 bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RRECT2 bit is 1.
1	RRECT1	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RRECT1 bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RRECT1 bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RRECT1 bit is 1.
0	RREIPV	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RREIPV bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RREIPV bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RREIPV bit is 1.

(26) RHSIFnRXICST — Rx ICLC Command Status Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RIPA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RITOL	RITOF	RITON	RIDT	RIET	RIFT	RIST	RIFR	RISR	RIPSTP	RIPSRT	RIPG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.85 RHSIFnRXICST Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	RIPA	This bit indicates the status of ICLC ping answer. 0: Ping answer not received. 1: Ping answer received.
15 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11	RITOL	This bit indicates the status of ICLC turn on payload loopback. 0: Turn on payload loopback not received. 1: Turn on payload loopback received.
10	RITOF	This bit indicates the status of ICLC turn off payload loopback. 0: Turn off payload loopback not received. 1: Turn off payload loopback received.
9	RITON	This bit indicates the status of ICLC turn on test mode. 0: Turn on test mode not received. 1: Turn on test mode received.
8	RIDT	This bit indicates the status of ICLC disable Tx port. 0: Disable Tx port not received. 1: Disable Tx port received.
7	RIET	This bit indicates the status of ICLC enable Tx port. 0: Enable Tx port not received. 1: Enable Tx port received.
6	RIFT	This bit indicates the status of ICLC select FastMode Tx port. 0: Select FastMode Tx port not received. 1: Select FastMode Tx port received.
5	RIST	This bit indicates the status of ICLC select SlowMode Tx port. 0: Select SlowMode Tx port not received. 1: Select SlowMode Tx port received.
4	RIFR	This bit indicates the status of ICLC select FastMode Rx port. 0: Select FastMode Rx port not received. 1: Select FastMode Rx port received.
3	RISR	This bit indicates the status of ICLC select SlowMode Rx port. 0: Select SlowMode Rx port not received. 1: Select SlowMode Rx port received.

Table 27.85 RHSIFnRXICST Register Contents (2/2)

Bit Position	Bit Name	Function
2	RIPSTP	This bit indicates the status of ICLC PLL Stop. 0: PLL Stop not received. 1: PLL Stop received.
1	RIPSRT	This bit indicates the status of ICLC PLL Start. 0: PLL Start not received. 1: PLL Start received.
0	RIPG	This bit indicates the status of ICLC Ping. 0: Ping not received. 1: Ping received.

(27) RHSIFnRXICSC — Rx ICLC Command Status Clear RegisterValue after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RICPA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RICTOL	RICTOF	RICTON	RICDT	RICET	RICFT	RICST	RICFR	RICSR	RICPSTP	RICPSRT	RICPG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W	W	W	W	W	W	W	W	W

Table 27.86 RHSIFnRXICSC Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	RICPA	Writing 1 to this bit clears the RHSIFnRXICST.RIPA bit. Writing 0 to this bit is ignored.
15 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11	RICTOL	Writing 1 to this bit clears the RHSIFnRXICST.RITOL bit. Writing 0 to this bit is ignored.
10	RICTOF	Writing 1 to this bit clears the RHSIFnRXICST.RITOF bit. Writing 0 to this bit is ignored.
9	RICTON	Writing 1 to this bit clears the RHSIFnRXICST.RITON bit. Writing 0 to this bit is ignored.
8	RICDT	Writing 1 to this bit clears the RHSIFnRXICST.RIDT bit. Writing 0 to this bit is ignored.
7	RICET	Writing 1 to this bit clears the RHSIFnRXICST.RIET bit. Writing 0 to this bit is ignored.
6	RICFT	Writing 1 to this bit clears the RHSIFnRXICST.RIRF bit. Writing 0 to this bit is ignored.
5	RICST	Writing 1 to this bit clears the RHSIFnRXICST.RIRS bit. Writing 0 to this bit is ignored.
4	RICFR	Writing 1 to this bit clears the RHSIFnRXICST.RITF bit. Writing 0 to this bit is ignored.
3	RICSR	Writing 1 to this bit clears the RHSIFnRXICST.RITS bit. Writing 0 to this bit is ignored.
2	RICPSTP	Writing 1 to this bit clears the RHSIFnRXICST.RIPSTP bit. Writing 0 to this bit is ignored.
1	RICPSRT	Writing 1 to this bit clears the RHSIFnRXICST.RIPSRT bit. Writing 0 to this bit is ignored.
0	RICPG	Writing 1 to this bit clears the RHSIFnRXICST.RIPG bit. Writing 0 to this bit is ignored.

(28) RHSIFnRXICIE — Rx ICLC Command Interrupt Enable Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RIEPA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RIETOL	RIETOF	RIETON	RIEDT	RIEET	RIEFT	RIEST	RIEFR	RIESR	RIEPSTP	RIEPSRT	RIEPG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.87 RHSIFnRXICIE Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	RIEPA	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIPA bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIPA bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIPA bit is 1.
15 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11	RIETOL	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RITOL bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RITOL bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RITOL bit is 1.
10	RIETOF	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RITOF bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RITOF bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RITOF bit is 1.
9	RIETON	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RITON bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RITON bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RITON bit is 1.
8	RIEDT	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIDT bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIDT bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIDT bit is 1.
7	RIEET	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIET bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIET bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIET bit is 1.
6	RIEFT	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIFT bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIFT bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIFT bit is 1.
5	RIEST	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIST bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIST bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIST bit is 1.
4	RIEFR	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIFR bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIFR bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIFR bit is 1.
3	RIESR	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RISR bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RISR bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RISR bit is 1.

Table 27.87 RHSIFnRXICIE Register Contents (2/2)

Bit Position	Bit Name	Function
2	RIEPSTP	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIPSTP bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIPSTP bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIPSTP bit is 1.
1	RIEPSRT	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIPSRT bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIPSRT bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIPSRT bit is 1.
0	RIEPG	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIPG bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIPG bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIPG bit is 1.

27.5.3 Operation

27.5.3.1 Initial Settings

(1) Clocks generated by CLKGEN and transfer mode

The CLKGEN generates the clock for data transmission/reception. **Table 27.88** below shows the correspondence between the clocks to be generated and the transfer mode. The switching between master mode and slave mode, the transfer mode, and the baud rate for fast transfer mode are specified by setting the registers.

During operation in slave mode, the transfer mode can be switched between fast and slow by the ICLC command transmitted from the master.

Table 27.88 Clocks Generated by CLKGEN

Data Transfer Modes	Operation Modes	Clock Sources	Sampling Clock	Baud Rate
			Frequency	
Fast Speed mode	Master/slave	Internal PLL	160, 80 MHz	160, 80 M Baud
Slow Speed mode	Master	System clock	20 MHz	5 M Baud
	Slave	External clock	10, 20 MHz	

(2) L1 settings

- (1) Master/slave setting
Specify the master or slave operation by setting the RHSIFnMDCR.MST register. When the master operation is specified, also specify the frequency of the clock to be output to communication device by setting the RHSIFnMDCR.CLKSEL.
- (2) Flow control setting
Specify the automatic flow control setting by setting the RHSIFnMDCR.CTSEN register.
- (3) RxD enable setting
Enable or disable RxD reception by setting the RHSIFnTXRXCR.RXEN register.
- (4) TxD enable setting
Enable or disable TxD transmission by setting the RHSIFnTXRXCR.TXEN.
- (5) Interrupt enable setting
Enable or disable interrupts by setting the RHSIFn*IE registers for L2 and L1, respectively.
- (6) Transmission/reception speed setting for fast transfer mode
Specify the transmission/reception speed (baud rate) by setting the RHSIFnSPCR.FMBR register.
- (7) Transfer mode setting for transmission/reception
Specify the transfer speed mode for transmission and reception by setting the RHSIFnSPCR.TXSP and RHSIFnSPCR.RXSP registers, respectively.
- (8) Procedure for releasing CLKGEN standby
After power-on reset release, the PLL in the CLKGEN is in the standby (OFF) state. It is necessary to release the PLL from the standby state (ON) to perform data transmission/reception in fast transfer mode. When switching from PLL OFF to PLL ON, be sure to release the standby state of the PLL by using the procedures of steps (a) to (e) described below.
If data transmission/reception is performed in fast transfer mode without following the procedure, the baud rate is unstable and normal operation may not be achieved.
 - (a) Insert waits by software so that the external reference clock (RHSIF0_REFCLK) becomes stable 1 μ s before the PLL is released from the standby state (indicated by step (d) below).
During master operation, however, no wait is required because the internal clock is used as the reference clock for PLL at this time.
 - (b) Set the following registers as required.
RHSIFnMDCR.MST
RHSIFnMDCR.CLKSEL
RHSIFnSPCR.FMBR
RHSIFnSPCR.TXSP
RHSIFnSPCR.RXSP
Be sure to update these registers when the PLL is in the standby state.
 - (c) Insert waits for at least 1 μ s by software.
 - (d) Clear the RHSIFnPCR.PLLSTBY bit to "0". The PLL released from the standby starts running.
 - (e) Insert waits for at least $t_{RHPLLCT}$ μ s by software until the PLL locks up. For details of PLL locked time, refer to *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

Note: When switching from PLL ON to PLL OFF, set the RHSIFnPCR.PLLSTBY bit to "1" first to set the PLL in standby state, and then change the registers described in step (b) above. If setting values of these registers are changed while the PLL is running, the PLL may not operate normally.

Figure 27.30 below shows a timing chart for the procedure of steps (a) to (e) above.

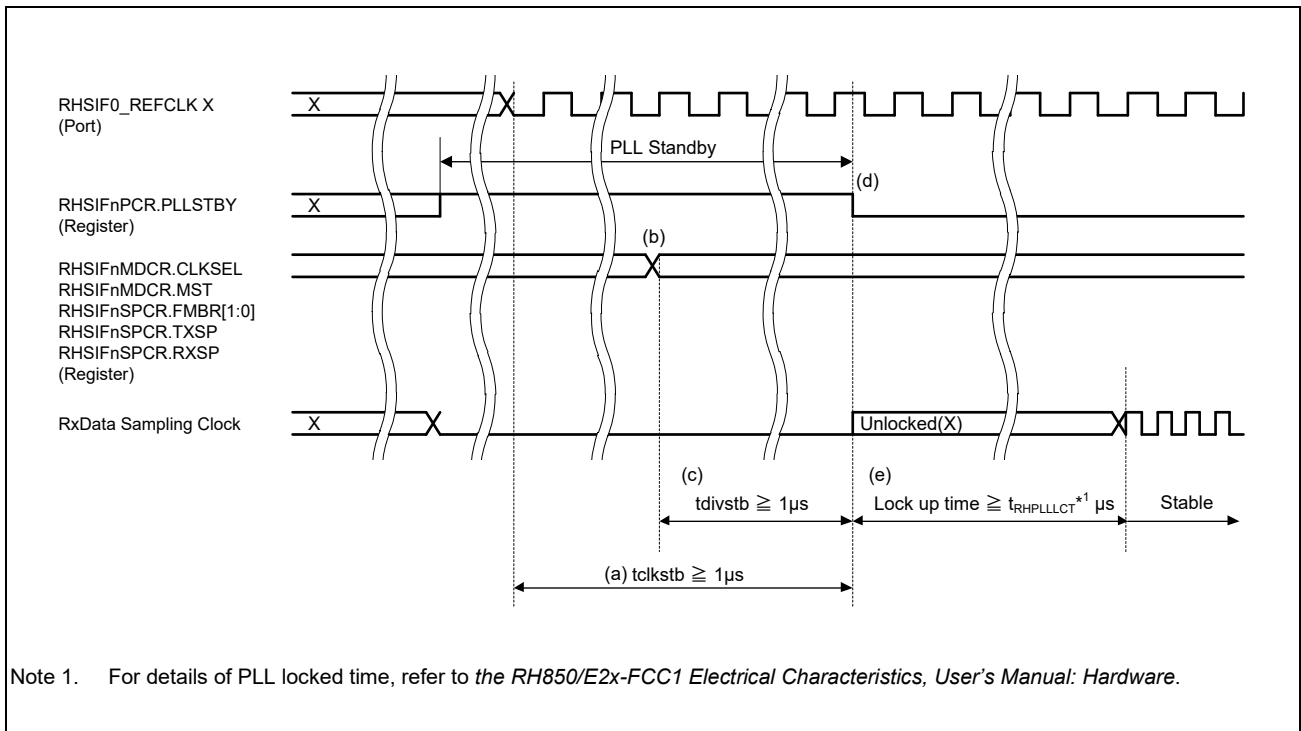


Figure 27.30 Timing Chart for Releasing CLKGEN from Reset State

27.5.3.2 ICLC Command Control Method

(1) Basic operation: ICLC command issuance

An ICLC command can be issued from the master to a slave. An ICLC command request is transmitted by firmware. Transmit a request by accessing the register and check the response by using an interrupt as trigger.

Figure 27.31 below shows the procedure flow for issuing an ICLC command.

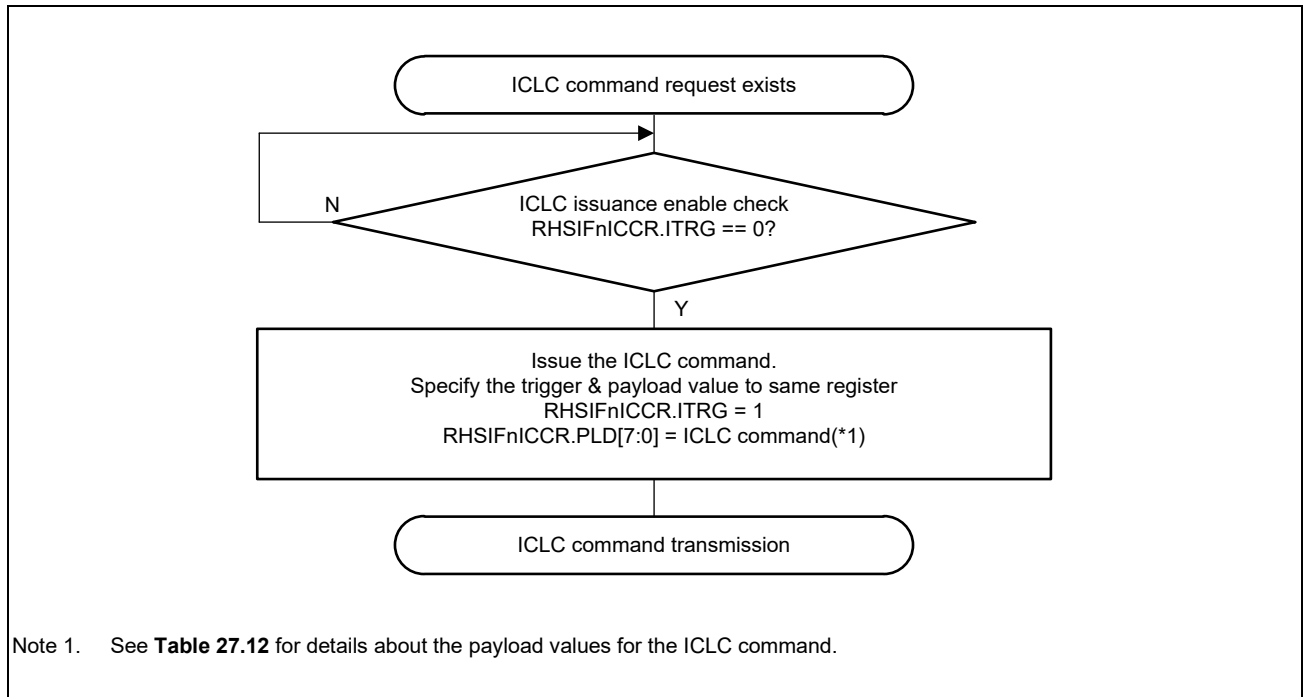


Figure 27.31 Procedure Flow for Issuing ICLC Command

For details about the ICLC commands supported by this product, see **Table 27.12, List of ICLC Command**.

(2) Basic operation: Interrupt

Figure 27.32 below shows the procedure flow for processing an interrupt, using transmission completion interrupt as example.

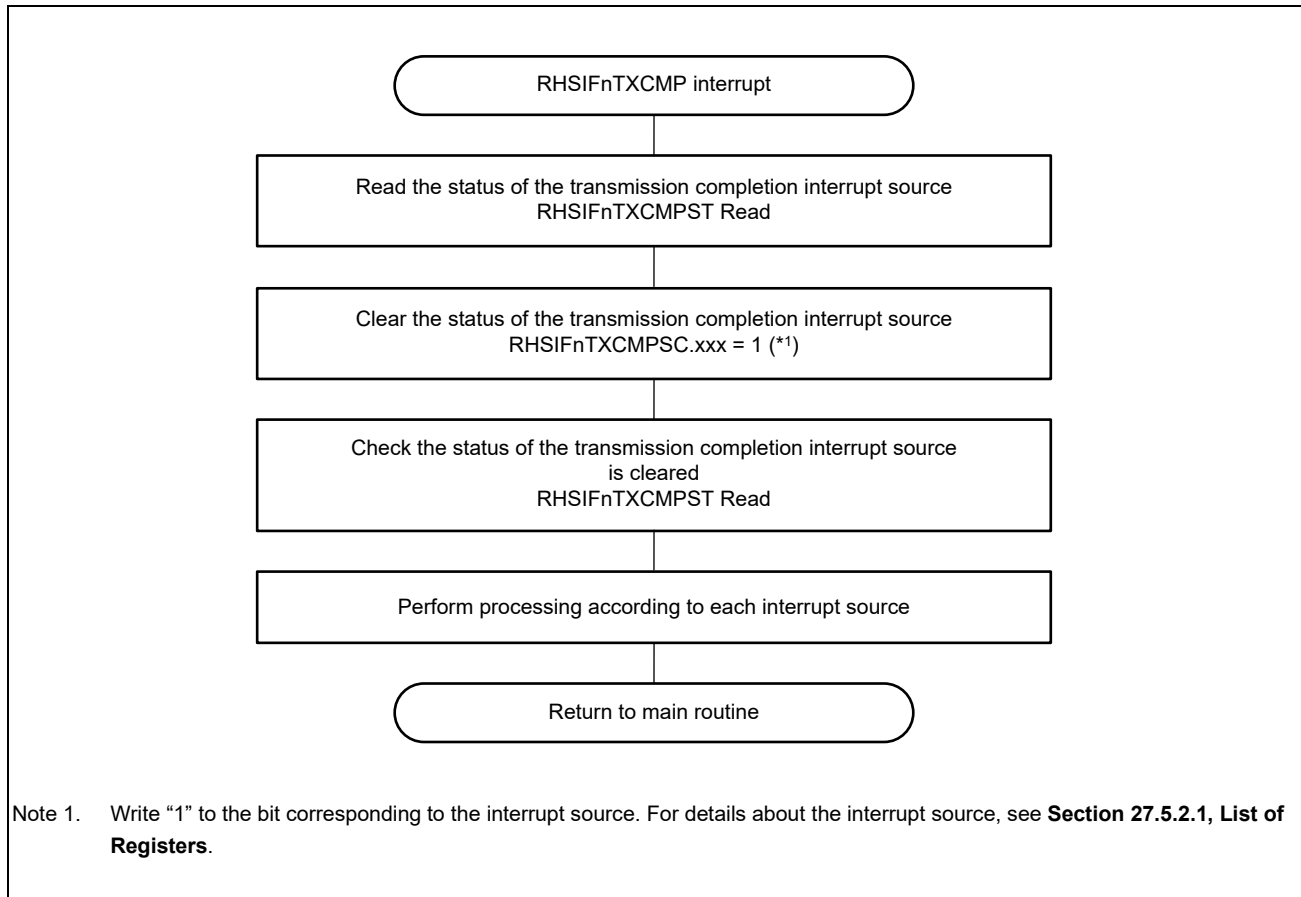


Figure 27.32 Procedure Flow for Processing Transmission Completion Interrupt

The processing is similar to the other interrupts. See **Table 27.89** below for the status registers, status clear registers and interrupt enable registers for the other interrupts.

Table 27.89 List of Interrupt Related Registers

Interrupt	Status Register	Status Clear Register	Interrupt Enable Register
RHSIFnTXCMP	RHSIFnTXCMPST	RHSIFnTXCMPSC	RHSIFnTXCMPIE
RHSIFnTXERR	RHSIFnTXERRST	RHSIFnTXERRSC	RHSIFnTXERRIE
RHSIFnRXCMP	RHSIFnRXCMPST	RHSIFnRXCMPSC	RHSIFnRXCMPIE
RHSIFnRXERR	RHSIFnRXERRST	RHSIFnRXERRSC	RHSIFnRXERRIE
RHSIFnRXICLC	RHSIFnRXICST	RHSIFnRXICSC	RHSIFnRXICIE

(3) Basic operation: Execution of ICLC “PING” command

Figure 27.33 below shows a flow for transmitting an ICLC “PING” command and receiving an ICLC “PING Answer”. The ICLC “PING Answer” can automatically be transmitted without firmware.

After completion of transmission, the device waits for a response from the link partner. The following five cases can be considered according to the responding state of the link partner and whether there is a frame error.

- (a) When the device operates as the master and the link partner sends back an ICLC “PING Answer” (normal case)

When the device completes transmission of a PING command, a transmission completion interrupt is generated.

Wait until the device receives the ICLC “PING Answer”.

When the device receives the “PING Answer”, a reception completion interrupt and an ICLC reception interrupt are generated.
- (b) Transmission errors when the device operates as the master

If an error occurs after the device transmits an ICLC “PING” command, a transmission error interrupt is generated.
- (c) Reception errors when the device operates as the master

If an error occurs after the device receives the ICLC “PING Answer”, a reception error interrupt is generated.
- (d) Reception errors when the device operates as a slave

If an error occurs after the device receives the ICLC “PING” command, a reception error interrupt is generated.
- (e) Transmission errors when the device operates as a slave

If an error occurs after the device transmits an ICLC “PING Answer”, a transmission error interrupt is generated.

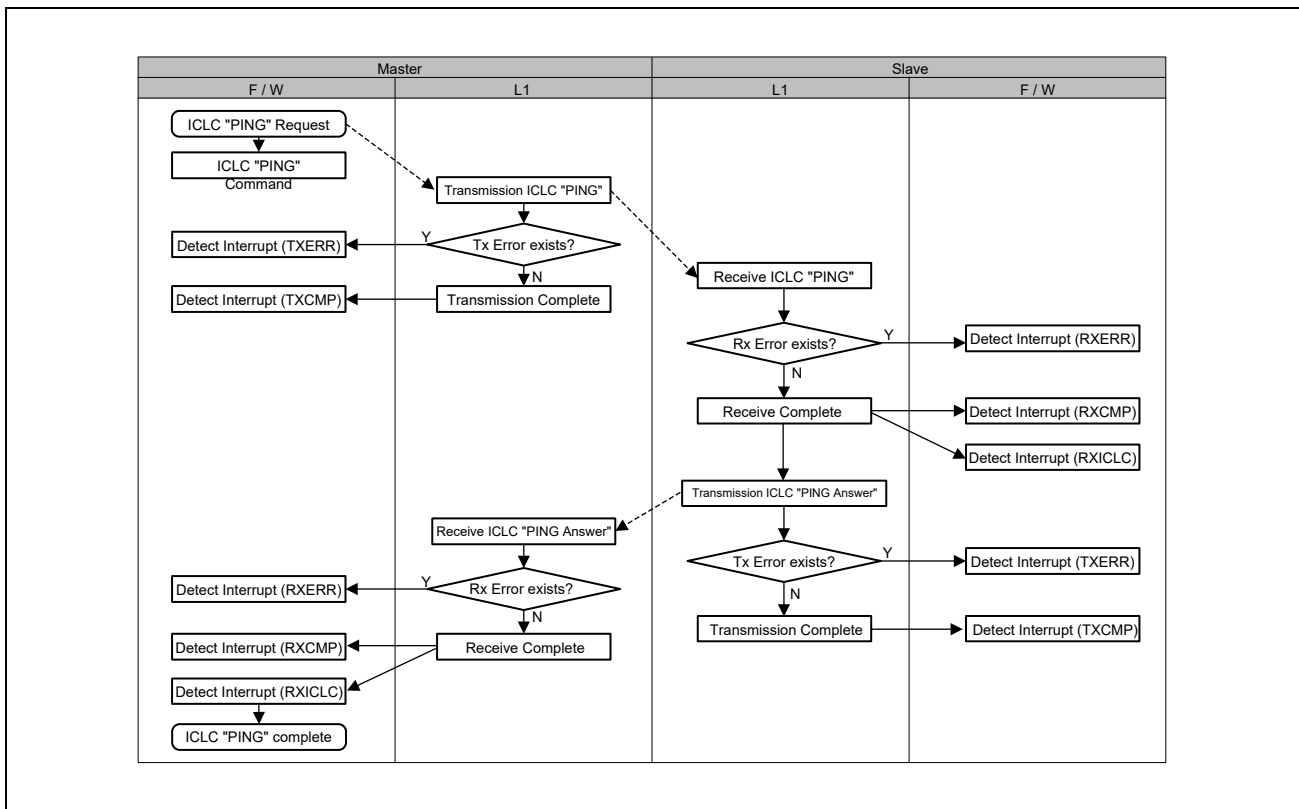


Figure 27.33 Flow for ICLC “PING” Command Execution

For the ICLC commands other than “PING”, the processing is the same except the slave does not send back an ICLC “PING Answer”.

27.5.3.3 Changing Communication Speed Using ICLC Commands

(1) Changing from slow transfer mode to fast transfer mode

Figure 27.34 to **Figure 27.40** show the procedure for changing the communication speed (transfer mode) from slow transfer mode to fast transfer mode.

For fast transfer mode, the transfer rate (baud rate) can be selected from 160 and 80 M Baud. Be sure to specify the transfer rate in advance, by setting the RHSIFnSPCR register.

Also, be sure to enable transmission for the master.

(a) Enable transmission for the RX link slave.

Transmit the ICLC “Enable Slave interface transmitter” command.

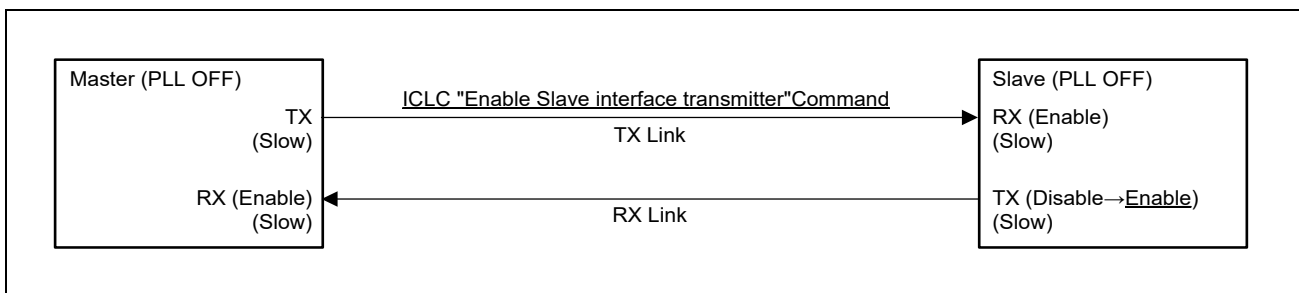


Figure 27.34 Enabling Transmission for RX Link Slave

(b) Start operation of the PLL in slave

Transmit the ICLC “Slave interface PLL start” command. Refer to **(8), Procedure for releasing CLKGEN standby** in **Section 27.5.3.1(2), L1 setting** of **Section 27.5.3.1, Initial Setting** for details.

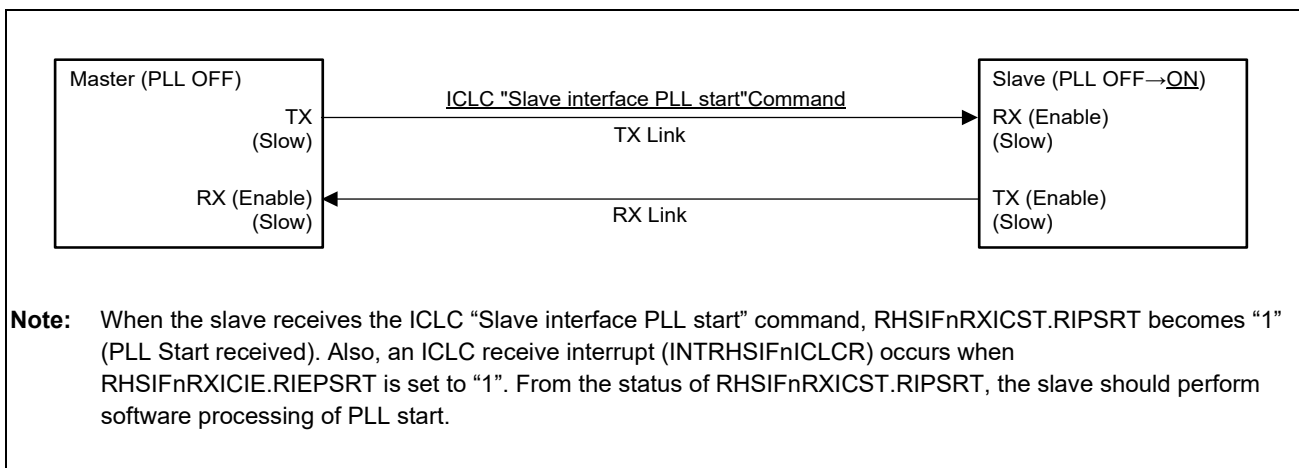


Figure 27.35 Starting Operation of PLL in Slave

(c) Set the transmission speed mode for Rx link slave to fast mode

Transmit the ICLC “Select Fast Speed mode for transfers from the Slave interface to the Master interface” command.

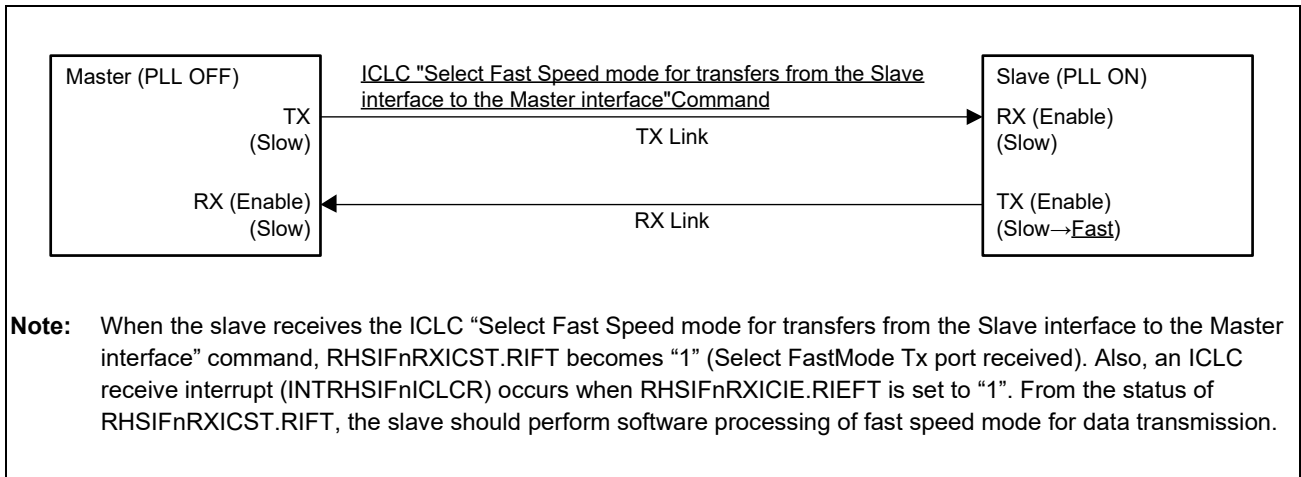


Figure 27.36 Setting Transmission Speed Mode for Slave to Fast

(d) Set the reception speed mode for Tx link slave to fast mode

Transmit the ICLC “Select Fast Speed mode for transfers from the Master interface to the Slave interface” command.

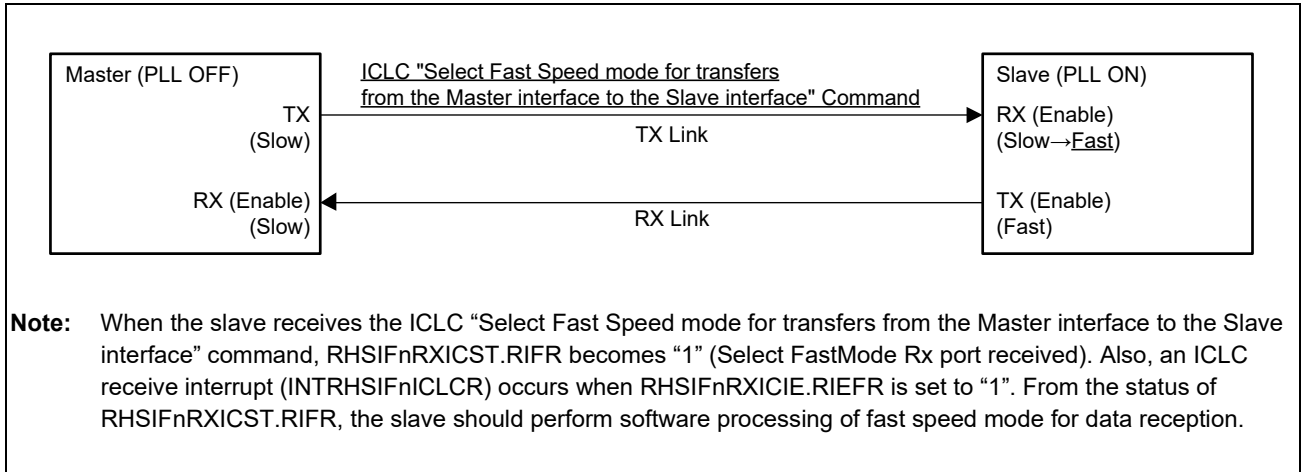


Figure 27.37 Setting Reception Speed for Slave to Fast

(e) Start operation of the PLL in master

Release the PLL from the standby state and start operation of the PLL by setting the APB registers. Refer to **(8)**, **Procedure for releasing CLKGEN standby** in **Section 27.5.3.1(2)**, **L1 setting** of **Section 27.5.3.1**, **Initial Setting** for details.

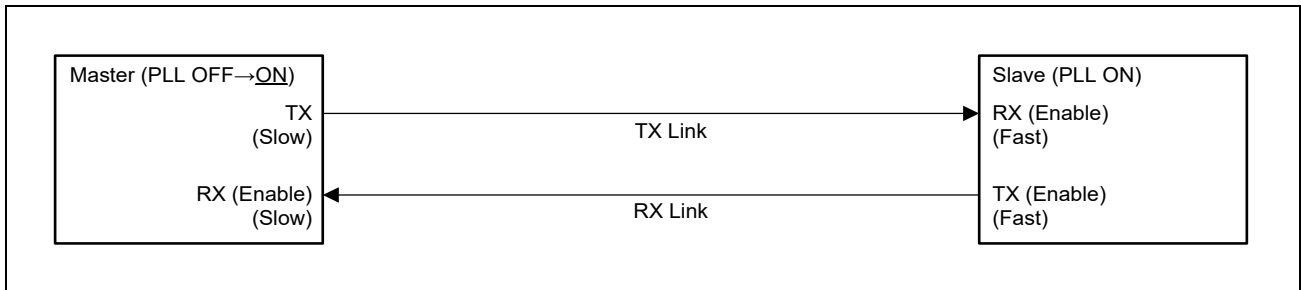


Figure 27.38 Starting Operation of PLL in Master

(f) Set the reception speed mode for Rx link and the transmission speed mode for Tx link of master to fast mode

Set the Rx and Tx speed modes to fast mode by setting RHSIFnSPCR.TXSP to “1” and RHSIFnSPCR.RXSP to “1”.

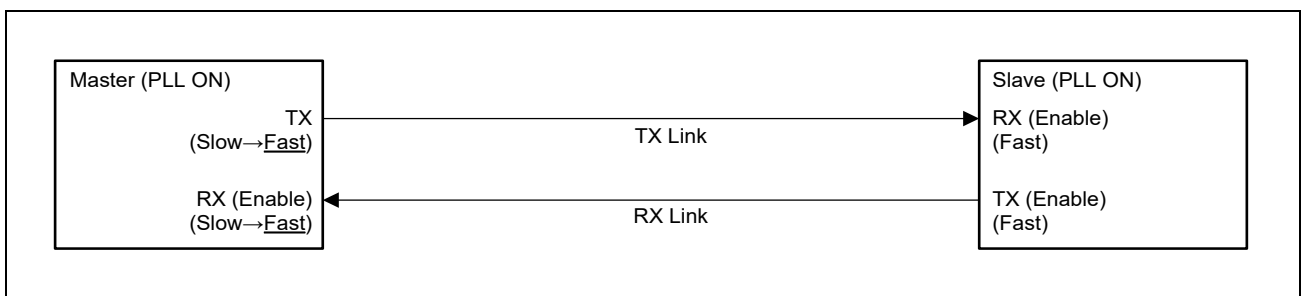


Figure 27.39 Setting Rx and Tx Speed Modes for Master to Fast

(g) Check the link establishment

Transmit the ICLC “PING” command and make sure that a “PING ANSWER” is sent back.

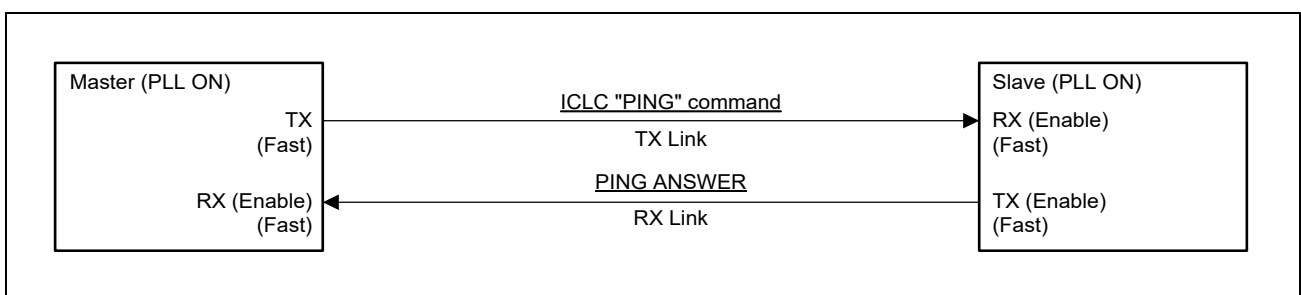


Figure 27.40 Checking Link Establishment

(2) Changing from fast transfer mode to slow transfer mode

Figure 27.41 to **Figure 27.47** show the procedure for changing the communication speed (transfer mode) from fast transfer mode to slow transfer mode.

Be sure to enable transmission for the master in advance

(a) Enable transmission for the RX link slave

Transmit the ICLC “Enable Slave interface transmitter” from the master to slave.

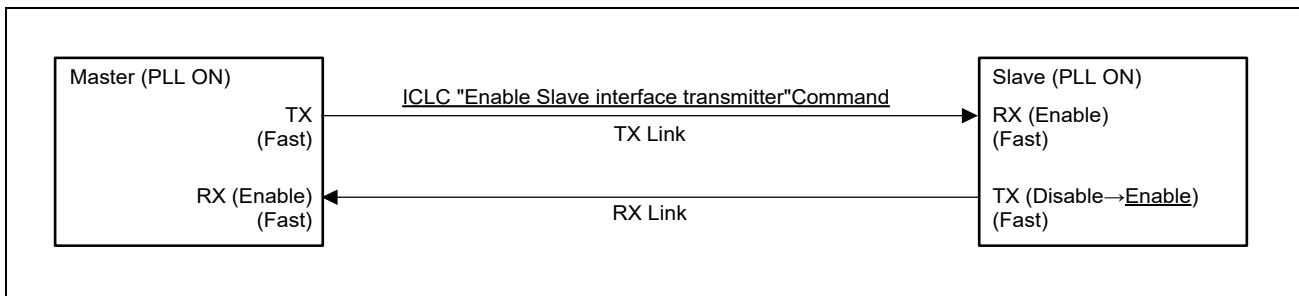
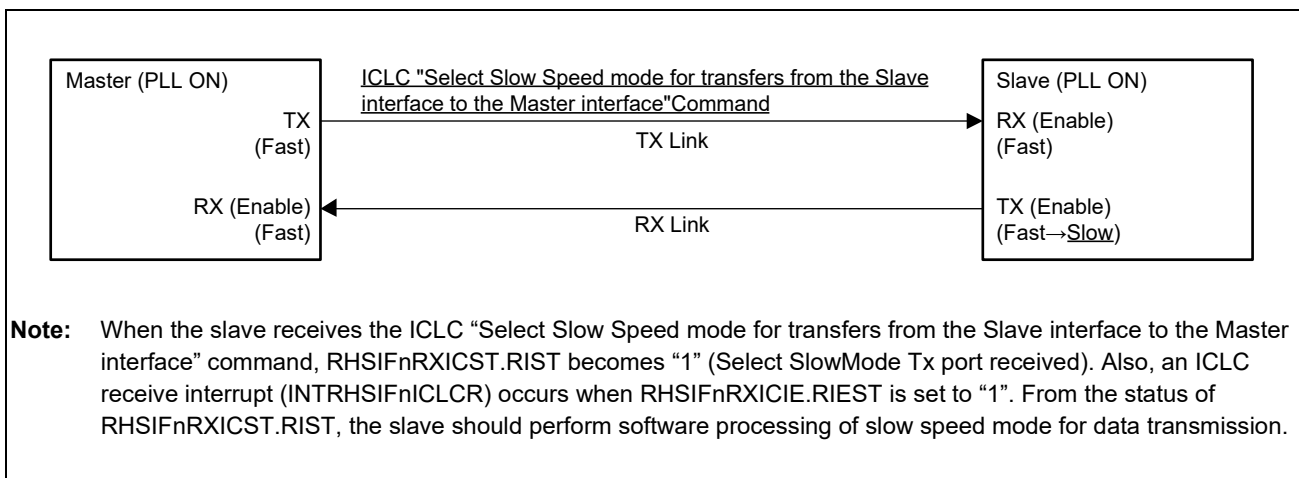


Figure 27.41 Enabling Transmission for RX Link Slave

(b) Set the transmission speed mode for slave to slow mode

Transmit the ICLC “Select Slow Speed mode for transfers from the Slave interface to the Master interface” command from the master to slave.



Note: When the slave receives the ICLC “Select Slow Speed mode for transfers from the Slave interface to the Master interface” command, RHSIFnRXICST.RIST becomes “1” (Select SlowMode Tx port received). Also, an ICLC receive interrupt (INTRHSIFnICLGR) occurs when RHSIFnRXICIE.RIEST is set to “1”. From the status of RHSIFnRXICST.RIST, the slave should perform software processing of slow speed mode for data transmission.

Figure 27.42 Setting Transmission Speed Mode for Slave to Slow

(c) Set the reception speed mode for slave to slow mode

Transmit the ICLC “Select Slow Speed mode for transfers from the Master interface to the Slave interface” command from the master to slave.

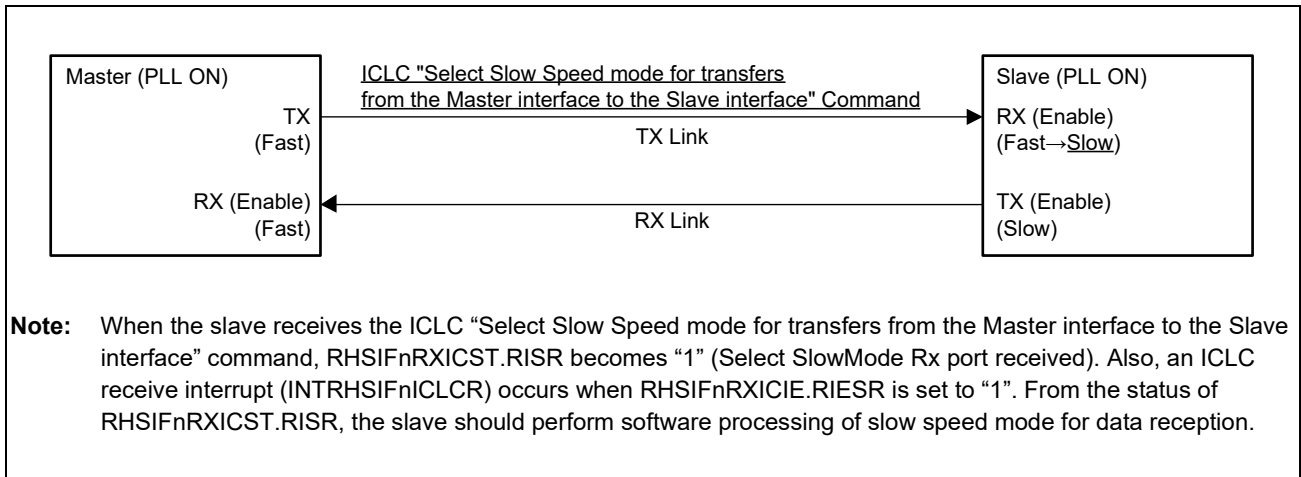


Figure 27.43 Setting Reception Speed for Slave to Slow

(d) Set the reception and transmission speed modes for master to slow mode

Set the Rx and Tx speed modes to slow by setting RHSIFnSPCR.RXSP to “0” and RHSIFnSPCR.TXSP to “0”.

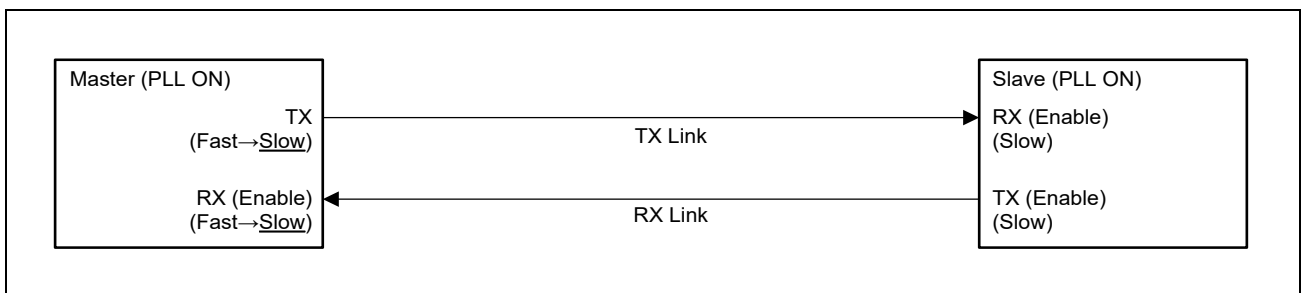


Figure 27.44 Setting Rx and Tx Speed Modes for Master to Slow

- (e) Stop the operation of the PLL in slave
 Transmit the ICLC “Slave interface PLL stop” command.

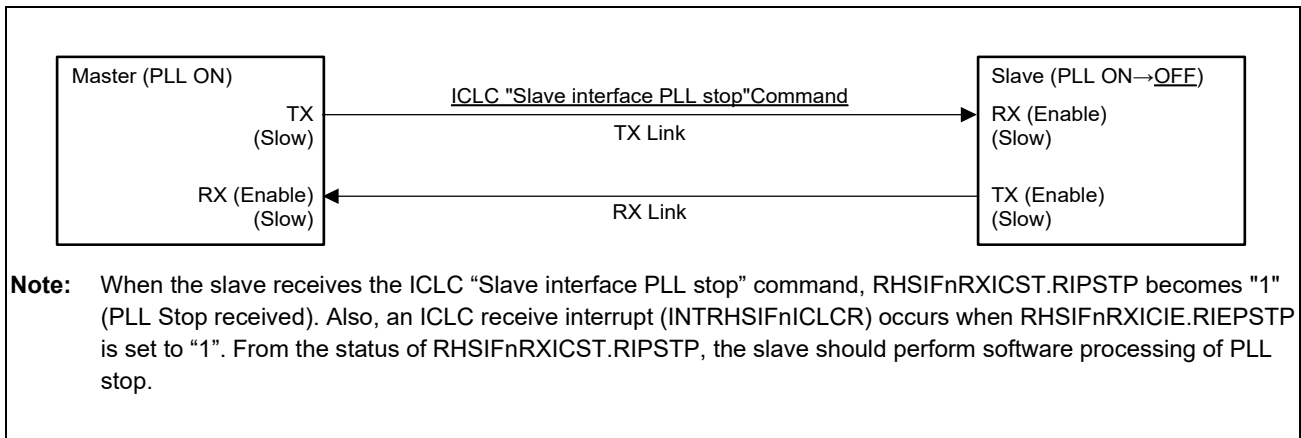


Figure 27.45 Stopping Operation of PLL in Slave

- (f) Stop the operation of the PLL in master
 Set RHSIFnPCR.PLLSTBY to “1” to stop the PLL operation in master.

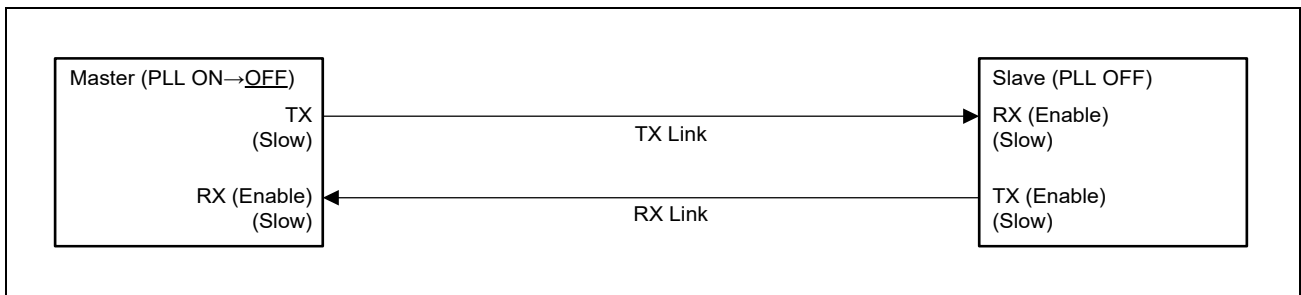


Figure 27.46 Stopping PLL Operation in Master

- (g) Check the link establishment
 Transmit the ICLC “PING” command from the master to slave, and make sure that a “PING ANSWER” is sent back.

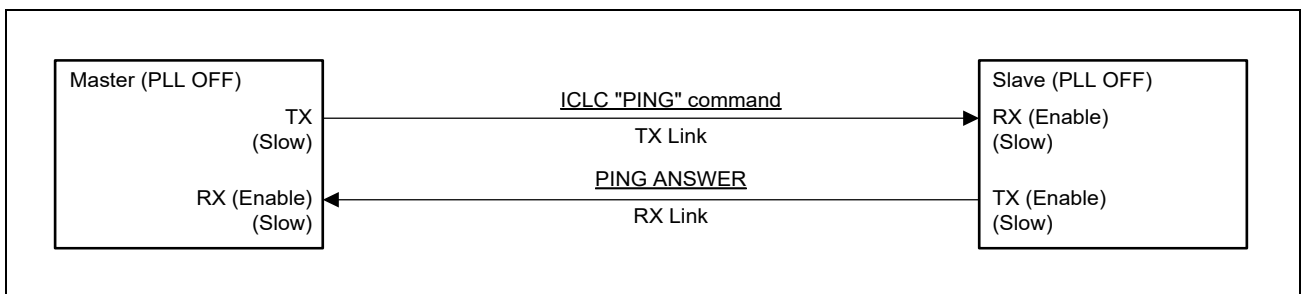


Figure 27.47 Checking Link Establishment

27.5.3.4 Transiting Slave State Using ICLC Commands

When the device is operating as a slave, the state transitions according to the ICLC command that is received as follows.

(1) Transmission enabled/disabled state

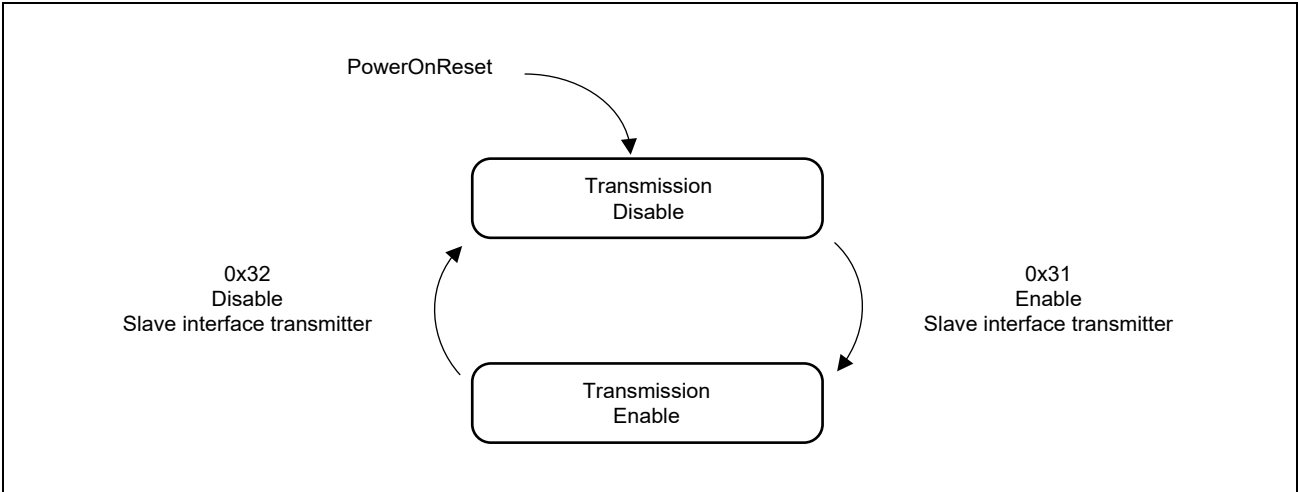


Figure 27.48 Transition between Transmission Enabled and Disabled States

(2) Tx Speed Mode Slow/Fast

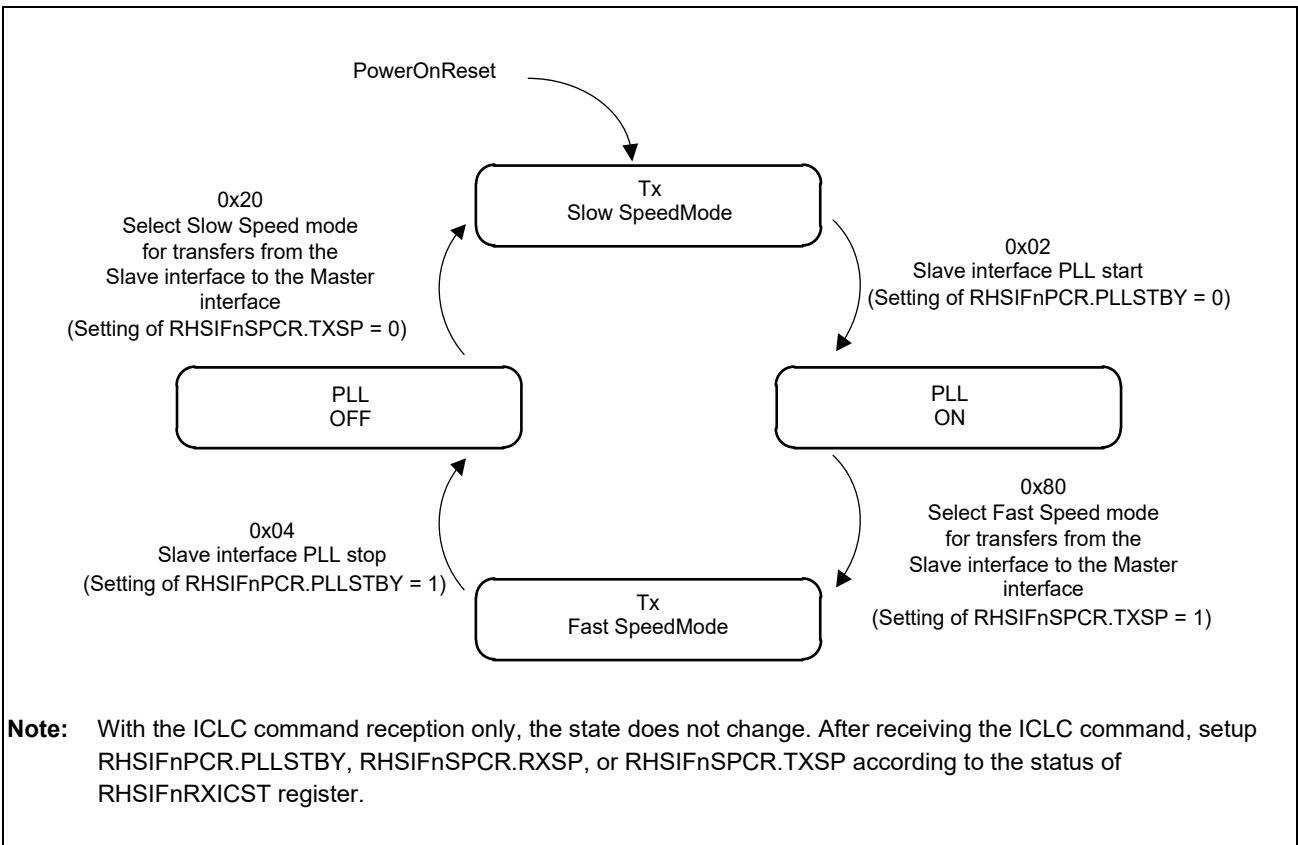


Figure 27.49 Transition between Tx Speed Modes Slow and Fast

(3) Rx Speed Mode Slow / Fast

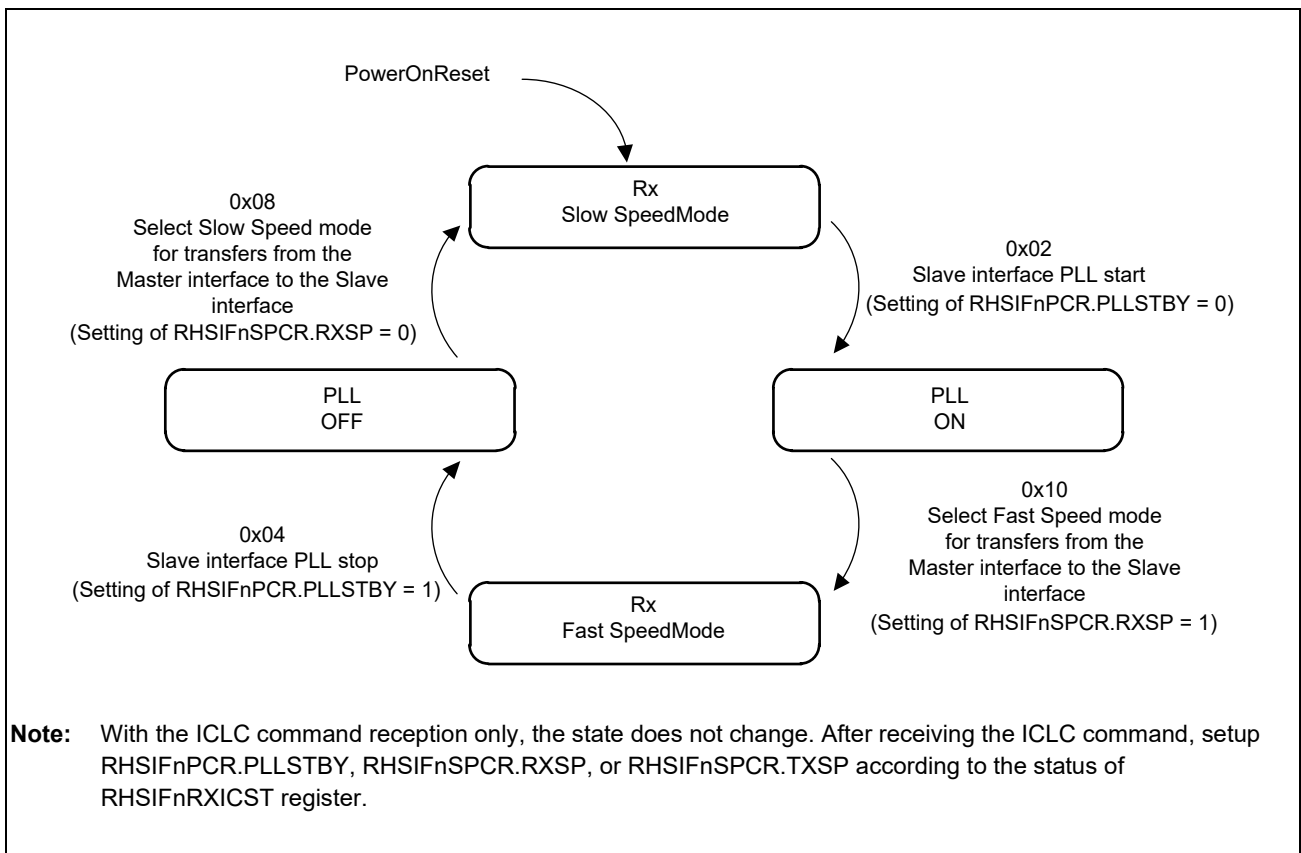


Figure 27.50 Transition between the Slow and Fast RX Speed Modes

(4) Test Mode

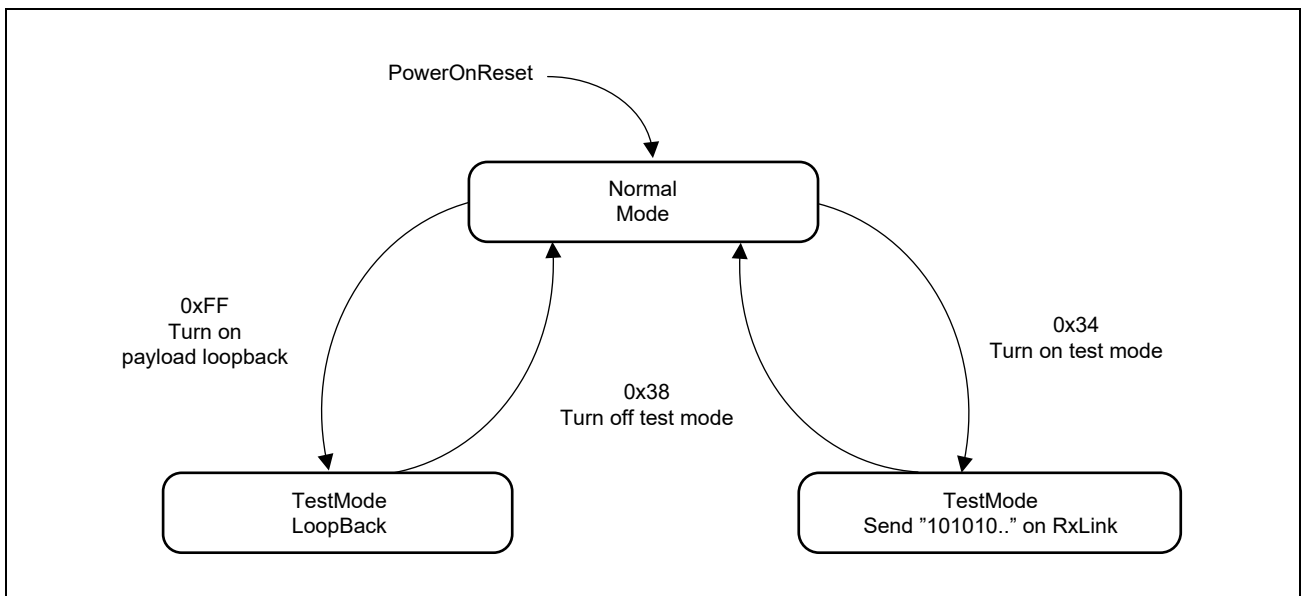


Figure 27.51 Transition between Test Modes

27.5.3.5 CTS Commands

(1) For Flow Control by CTS

The automatic flow control is enabled after PowerOnReset.

The automatic flow control is disabled by the RHSIFnMDCR.CTSEN = 0.

(a) When the Automatic Flow Control is Enabled

(i) Link partner is not ready

Link partner: Transmits a frame with a CTS value = 0.

This Product: Receives a frame and reflects the CTS value to RHSIFnL1SR.RCTS.

This Product: Suppresses the next frame transmission.

Link partner: Transmits a frame with a CTS value = 1 after becoming receivable.

This Product: Receives a frame and reflects the CTS value to RHSIFnL1SR.RCTS.

This Product: Resumes the transmission of the next frame.

(ii) This product is not ready

This Product: When RxDIFIFO is full, transmits a CTS command frame of CTS value = 0.

This Product: When RxDIFIFO is not full, transmits a CTS command frame with a CTS value = 1.

Caution: Do not write '0' to RHSIFnMDCR.CTSV while Automatic Flow Control is enabled.

(b) When the Automatic Flow Control is Disabled

By using the following register, it is possible to control the flow manually.

(i) reception status of Link partner

RHSIFnL1SR.RCTS

(ii) reception status of This Product

RHSIFnL1SR.RFFUL

RHSIFnMDCR.CTSV

(2) Basic Operation: CTS Command Issuance

A CTS command request is transmitted by firmware. Transmit a request by accessing the register.

Figure 27.52 below shows the procedure flow for issuing a CTS command.

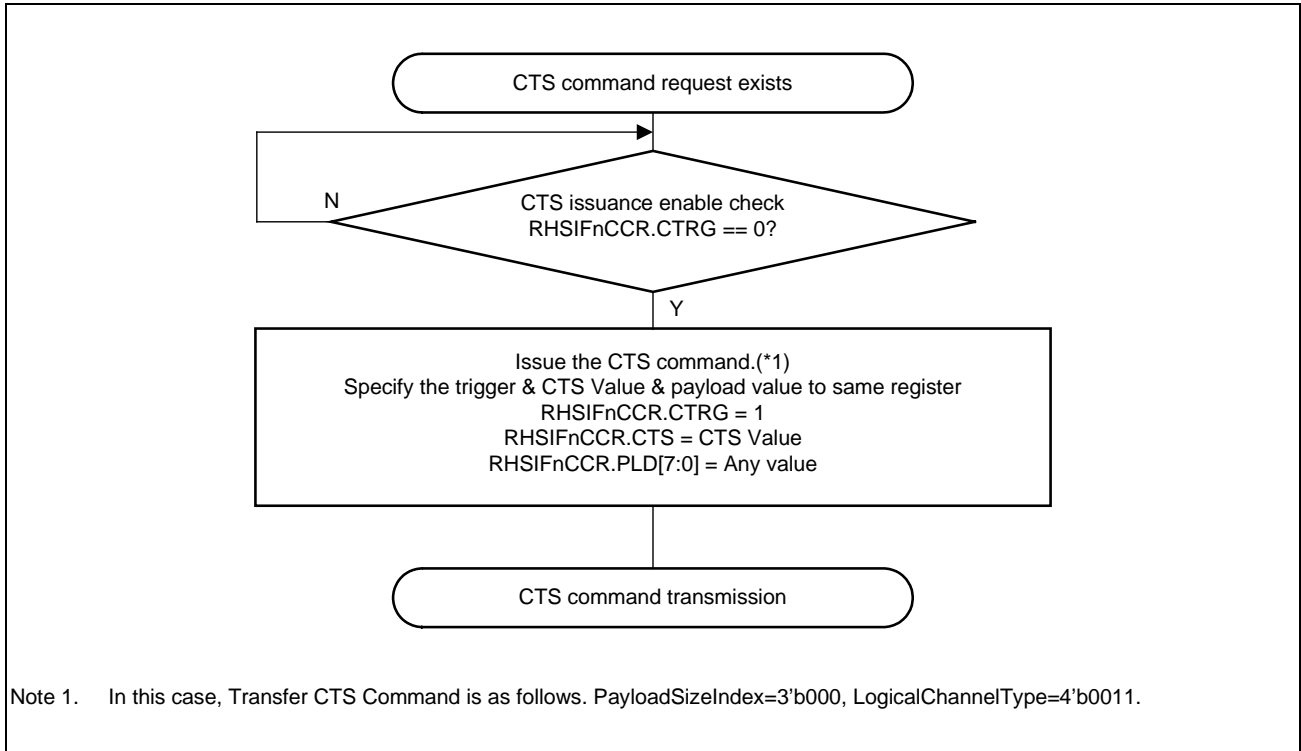


Figure 27.52 Procedure Flow for Issuing CTS Command

27.5.3.6 Sleep Mode

(1) Transitioning to Sleep Mode for the Link Partner

If an L2 frame, ICLC command, or CTS command is transmitted after RHSIFnSMCR.SLP is set to “1”, the L1 END bit continues to transmit “1”. This makes the Link Partner to transit to sleep mode.

(2) Recovering from Sleep Mode

This device does not support sleep mode when receiving.

RHSIF communication has no issues if RHSIF receives the sleep mode restoration condition specified by the communication protocol standard shown below.

- In fast transfer mode: Zero-level signal of 8-bit rate or more
- In slow transfer mode: Zero-level signal of 1-bit rate or more

27.5.4 Interrupt Specification

27.5.4.1 Interrupt Source List

The tables below list the error interrupt sources detected in L1.

Table 27.90 Transmission Completion Interrupt

No.	Register		Description
	Register Name	Bit Name	
1	RHSIFnTXCMP	TCL2	Transmission frame L2 completion interrupt
2		TCCT	Transmission frame CTS completion interrupt
3		TCIC	Transmission frame ICLC completion interrupt

Table 27.91 Transmission Error Interrupts

No.	Register		Description
	Register Name	Bit Name	
1	RHSIFnTXERRST	TERSZ	This bit indicates the status of payload size error of transmission.
2		TERSZI5	Transmission frame L1 header payload code error (undefined code 0x5 detected)
3		TERSZI4	Transmission frame L1 header payload code error (undefined code 0x4 detected)
4		TERCTF	Transmission frame L1 header channel type code error (undefined code 0xF detected)
5		TERCTE	Transmission frame L1 header channel type code error (undefined code 0xE detected)
6		TERCTD	Transmission frame L1 header channel type code error (undefined code 0xD detected)
7		TERCTC	Transmission frame L1 header channel type code error (undefined code 0xC detected)
8		TERCTB	Transmission frame L1 header channel type code error (undefined code 0xB detected)
9		TERCTA	Transmission frame L1 header channel type code error (undefined code 0xA detected)
10		TERCT9	Transmission frame L1 header channel type code error (undefined code 0x9 detected)
11		TERCT8	Transmission frame L1 header channel type code error (undefined code 0x8 detected)
12		TERCT2	Transmission frame L1 header channel type code error (undefined code 0x2 detected)
13		TERCT1	Transmission frame L1 header channel type code error (undefined code 0x1 detected)

Table 27.92 Reception Completion Interrupt

No.	Register		Description
	Register Name	Bit Name	
1	RHSIFnRXCMPST	RCL2	Reception frame L2 completion interrupt
2		RCCT	Reception frame CTS completion interrupt
3		RCIC	Reception frame ICLC completion interrupt

Table 27.93 Reception Error Interrupts

No.	Register		Description
	Register Name	Bit Name	
1	RHSIFnRXERRST	RERSZ	This bit indicates the status of payload size error of reception.
2		RERSZI5	Reception frame L1 header payload code error (undefined code 0x5 detected)
3		RERSZI4	Reception frame L1 header payload code error (undefined code 0x4 detected)
4		RERCTF	Reception frame L1 header channel type code error (undefined code 0xF detected)
5		RERCTE	Reception frame L1 header channel type code error (undefined code 0xE detected)
6		RERCTD	Reception frame L1 header channel type code error (undefined code 0xD detected)
7		RERCTC	Reception frame L1 header channel type code error (undefined code 0xC detected)
8		RERCTB	Reception frame L1 header channel type code error (undefined code 0xB detected)
9		RERCTA	Reception frame L1 header channel type code error (undefined code 0xA detected)
10		RERCT9	Reception frame L1 header channel type code error (undefined code 0x9 detected)
11		RERCT8	Reception frame L1 header channel type code error (undefined code 0x8 detected)
12		RERCT2	Reception frame L1 header channel type code error (undefined code 0x2 detected)
13		RERCT1	Reception frame L1 header channel type code error (undefined code 0x1 detected)
14		RERIPV	Reception ICLC command undefined error.

Table 27.94 ICLC Reception Completion Interrupt

No.	Register		Description
	Register Name	Bit Name	
1	RHSIFnRXICST	RIPA	Reception of PING answer frame
2		RITOL	Reception of Turn on payload loopback frame
3		RITOF	Reception of Turn off test mode frame
4		RITON	Reception of Turn on test mode frame
5		RIDT	Reception of Disable Slave interface transmitter frame
6		RIET	Reception of Enable Slave interface transmitter frame
7		RIFT	Reception of Select Fast Speed mode for transfers from the Slave interface to the Master interface frame
8		RIST	Reception of Select Slow Speed mode for transfers from the Slave interface to the Master interface frame
9		RIFR	Reception of Select Fast Speed mode for transfers from the Master interface to the Slave interface frame
10		RISR	Reception of Select Slow Speed mode for transfers from the Master interface to the Slave interface frame
11		RIPSTP	Reception of Slave interface PLL stop
12		RIPSRT	Reception of Slave interface PLL start
13		RIPG	Reception of PING frame

27.5.4.2 Error Detect Conditions

The tables below list the error detect conditions.

Table 27.95 The combination of Transmission Payload size index and Transmission Logical channel type

			Logical Channel Type						
			ICLC*2	CTS*2	L2Frame				
			b0000	b0011	b0100	b0101	b0110	b0111	others*3
Payload_Size_Index	8 bit	b000	TCIC	TCCT	TERSZ	TERSZ	TERSZ	TERSZ	TERSZ
	32 bit	b001	TCIC	— *1	TCL2	TCL2	TCL2	TCL2	TERSZ
	64 bit	b010	— *1	— *1	TCL2	TCL2	TCL2	TCL2	TERSZ
	96 bit	b011	— *1	— *1	TCL2	TCL2	TCL2	TCL2	TERSZ
	Not used	b100	— *1	— *1	TERSZI4	TERSZI4	TERSZI4	TERSZI4	TERSZ TERSZI4*4
		b101	— *1	— *1	TERSZI5	TERSZI5	TERSZI5	TERSZI5	TERSZ TERSZI5*4
	160 bit	b110	— *1	— *1	TCL2	TCL2	TCL2	TCL2	TERSZ
	288 bit	b111	— *1	— *1	TCL2	TCL2	TCL2	TCL2	TERSZ

Note 1. These conditions will not occur.

Note 2. In L1 Frame, error bits are generated only when Logical Channel Type is b0000(ICLC) or b0011(CTS).

Note 3. In L2 Frame, the generated error bit is always TERSZ when the Logical Channel Type is other than b01xx.

Note 4. Either TERSZI4 or TERSZI5 is simultaneously generated with TERSZ according to the Payload Size Index value.

Table 27.96 The combination of Reception Payload size index and Reception Logical channel type (1 of 2)

			Logical_Channel_Type							
			ICLC	Reserved		CTS	L2Frame			
			b0000	b0001	b0010	b0011	b0100	b0101	b0110	b0111
Payload_Size_Index	8 bit	b000	RCIC	RERCT1	RERCT2	RCCT	RERSZ	RERSZ	RERSZ	RERSZ
	32 bit	b001	RCIC RERSZ	RERCT1	RERCT2	RCCT	RCL2	RCL2	RCL2	RCL2
	64 bit	b010	RERSZ	RERCT1	RERCT2	RCCT	RCL2	RCL2	RCL2	RCL2
	96 bit	b011	RERSZ	RERCT1	RERCT2	RCCT	RCL2	RCL2	RCL2	RCL2
	Not used	b100	RERSZI4	RERCT1 RERSZI4*1	RERCT2 RERSZI4*1	RERSZI4	RERSZI4	RERSZI4	RERSZI4	RERSZI4
		b101	RERSZI5	RERCT1 RERSZI5*1	RERCT2 RERSZI5*1	RERSZI5	RERSZI5	RERSZI5	RERSZI5	RERSZI5
	160 bit	b110	RERSZ	RERCT1	RERCT2	RCCT	RCL2	RCL2	RCL2	RCL2
288 bit	b111	RERSZ	RERCT1	RERCT2	RCCT	RCL2	RCL2	RCL2	RCL2	

Note 1. Either RERSZI4 or RERSZI5 is simultaneously generated with RERCT1 or RERCT2 according to the Payload Size Index value.

Table 27.97 The combination of Reception Payload size index and Reception Logical channel type (2 of 2)

			Logical_Channel_Type							
			Reserved							
			b1000	b1001	b1010	b1011	b1100	b1101	b1110	b1111
Payload_Size_Index	8bit	b000	RERCT8	RERCT9	RERCTA	RERCTB	RERCTC	RERCTD	RERCTE	RERCTF
	32bit	b001	RERCT8	RERCT9	RERCTA	RERCTB	RERCTC	RERCTD	RERCTE	RERCTF
	64bit	b010	RERCT8	RERCT9	RERCTA	RERCTB	RERCTC	RERCTD	RERCTE	RERCTF
	96bit	b011	RERCT8	RERCT9	RERCTA	RERCTB	RERCTC	RERCTD	RERCTE	RERCTF
	Not used	b100	RERCT8 RERSZI4*1	RERCT9 RERSZI4*1	RERCTA RERSZI4*1	RERCTB RERSZI4*1	RERCTC RERSZI4*1	RERCTD RERSZI4*1	RERCTE RERSZI4*1	RERCTF RERSZI4*1
		b101	RERCT8 RERSZI5*1	RERCT9 RERSZI5*1	RERCTA RERSZI5*1	RERCTB RERSZI5*1	RERCTC RERSZI5*1	RERCTD RERSZI5*1	RERCTE RERSZI5*1	RERCTF RERSZI5*1
	160bit	b110	RERCT8	RERCT9	RERCTA	RERCTB	RERCTC	RERCTD	RERCTE	RERCTF
288bit	b111	RERCT8	RERCT9	RERCTA	RERCTB	RERCTC	RERCTD	RERCTE	RERCTF	

Note 1. Either RERSZI4 or RERSZI5 is simultaneously generated with RERCT8~RERCTF according to the Payload Size Index value.

27.5.4.3 Operation Upon Error Occurrence

The operations upon error occurrence in L1 consist only of the following.

- Assertion of error interrupt
- Indication of error interrupt status by register

27.5.5 Debug

Table 27.98 List of Debug Function

Purpose	Item	Content		APB Register
Debug	ICLC(0x34)	Turn on test mode (Send 101010.. Txline data rate)		RHSIFnTMDCR.CKTMSEL
	Loopback	TxD->RxD		RHSIFnTMDCR.TXLPBK
	Sent Frame information	Last sent frame information display	L1 frame	RHSIFnLTXFRMRL1.TL1E
				RHSIFnLTXFRMRL1.TL1P
				RHSIFnLTXFRMRL1.TL1H
			L2 frame	RHSIFnLTXFRMRL2.T2L2H
				RHSIFnLTXFRMRL2.T2L1H
	Received frame information	Last received frame information display	L1 frame	RHSIFnLRXFRMRL1.RL1E
				RHSIFnLRXFRMRL1.RL1P
				RHSIFnLRXFRMRL1.RL1H
			L2 frame	RHSIFnLRXFRMRL2.R2L2H
RHSIFnLRXFRMRL2.R2L1H				

27.5.6 Usage Note

27.5.6.1 About Fast Mode of Slave

A fast mode of Slave (It's sampled using a reference clock.) recommends following correspondence because there is a possibility that BER (the bit error rate) deteriorates by influence of a jitter.

- Use Slave's fast mode only when sending / receiving various ICLC commands (during communication establishment).
- After confirming the establishment of communication by the PING command of ICLC when communicating by a fast mode, begin to change to Master (sampling by MOSC) and communicate from $t_{RHPLLCT}$ μ s later (PLL locked time of a clock change). For details of PLL locked time, refer to *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

Section 28 Window Watchdog Timer (WDTB)

28.1 Features of WDTB

28.1.1 Units and Channels

This microcontroller has the following number of WDTB units.

Each WDTB unit has one channel interface.

Table 28.1 Number of Units

Product Name	RH850/E2x-FCC1 Series	
	373 Pins	292 Pins
Number of Units	2	
Name	WDTBn (n = 0 to 1)	

Meaning of n

Throughout this section, the individual units of a window watchdog timer are identified by the index “n” (n = 0, 1), for example, WDTBnWDTE for the WDTBn enable register.

28.1.2 Register Base Addresses

All WDTBn register addresses are given as address offsets from the individual base address <WDTBn_base>.

Table 28.2 shows the register base address of each WDTBn.

Table 28.2 Register Base Addresses of WDTBn

Base Address Name	Base Address	Bus Group
<WDTB0_base>	FFED 0000 _H	Peripheral Group 2L
<WDTB1_base>	FFED 1000 _H	Peripheral Group 2L

28.1.3 Clock Supply

The window watchdog timer uses WDTBTCKI and PCLK as a clock input.

Table 28.3 Clock Supply of WDTB

Unit Name	Unit Clock Name	Clock Supply Name
WDTBn	WDTBTCKI	CLK_WDT* ¹
	PCLK	CLK_LSB* ²

Note 1. CLK_WDT is a 250 kHz clock selectable from CLK_MOSC or CLK_IOSC.

Note 2. CLK_LSB is a 40 MHz clock. For details on the clock, see **Section 15, Clock Controller**.

28.1.4 Interrupt Requests

WDTBn interrupt requests are listed in the following table.

Table 28.4 Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Description	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
INTWDTB0TIT	WDTBnTIT (n = 0)	WDTB0 interrupt	CPU0-22	—	—
INTWDTB1TIT	WDTBnTIT (n = 1)	WDTB1 interrupt	CPU1-22	—	—

28.1.5 Reset Sources

WDTBn reset source are listed in the following table. WDTBn is initialized by these reset sources.

Table 28.5 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
WDTBn	All registers	✓	✓	✓	✓	—	—

28.1.6 Correspondence of WDTB and CPU

Table 28.6 Correspondence of WDTB and CPU

WDTBn	CPU Unit
WDTB0	CPU0
WDTB1	CPU1

28.2 Overview

28.2.1 Functional Overview

The WDTB has the following functions:

- WDTB counter start function by software.
- Default start function (WDTB0 only)
 - Selection of the operating mode after release from reset using the option byte.
 - Starting/stopping of the counter after WDTB is reset, enabling/disabling of interrupt requests, selection of the window-open period and the overflow interval time.
- Interrupt request signals
 - The interrupt request output timing can be set using a register.
- Window function
 - The period during which writing to the WDTB trigger register is valid (window-open period) can be set. Writing to the WDTB trigger register outside the window-open period causes an error.
 - The window open period can be set using a register.
- WDTB error detection
 - When an error is detected, the WDTBnTRES signal indicates the error to the ECM.
For details on WDTB error detection, see **Section 28.4.3, WDTB Error Detection**.
- Selectable counter clock for internal clock or external clock by option byte.
 - External clock is CLK_MOSC, and internal clock is CLK_IOSC.
For details on the clock, see **Section 15, Clock Controller**.

28.2.2 Block Diagram

Figure 28.1 shows the main components of WDTB.

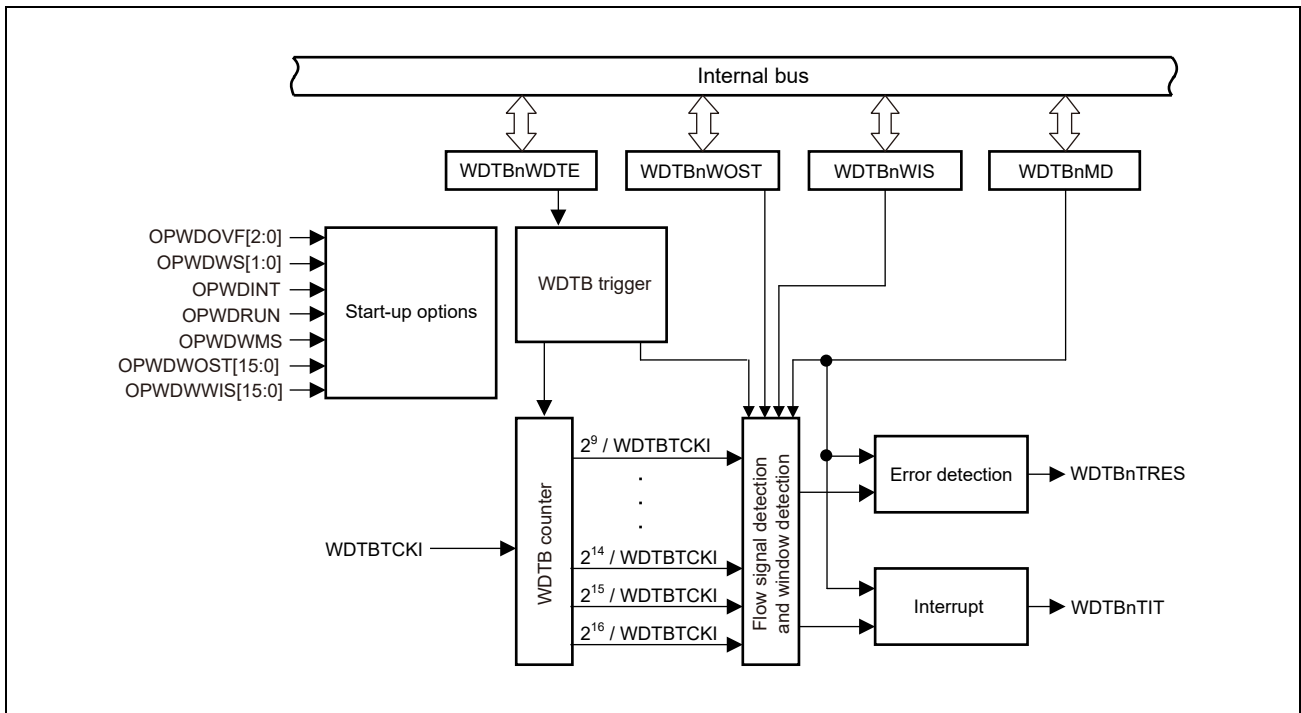


Figure 28.1 Block Diagram of WDTB

28.3 Registers

WDTB is controlled and operated by the following registers.

28.3.1 List of Registers

The following shows the list of registers and the memory address of WDTBn.

For the base addresses, see **Table 28.2**.

The actual address can be obtained by adding the offset shown in the table to this base address.

Table 28.7 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection
WDTBn	WDTB Enable Register	WDTBnWDTE	<WDTBn_base> + 0000 _H	8	—
	WDTB Mode Register	WDTBnMD	<WDTBn_base> + 000C _H	8	—
	WDTB Window Open Start Register	WDTBnWOST	<WDTBn_base> + 0010 _H	16	—
	WDTB Interrupt Output Timing Setting Register	WDTBnWIS	<WDTBn_base> + 0014 _H	16	—

28.3.2 Detail of Registers

28.3.2.1 WDTBnWDTE — WDTB Enable Register

This register is the WDTB start control and trigger register.

Writing AC_H to this register generates a WDTB trigger and starts or restarts the WDTB counter.

Please refer to **Section 28.4.2, WDTB Trigger** for details.

The value that can be written in this register is only AC_H .

Value after reset: (1) WDTB0

This value depends on the set value of OPBT0.

The value after reset depends on the setting of the OPWDRUN bit.

The initial state of the OPWDRUN bit at shipment is $OPWDRUN = 0_B$.

Therefore, the register value after reset of WDTB0 at shipment is $2C_H$.

(2) WDTBn ($n = 1$)

The register value after reset is also $2C_H$.

Bit	7	6	5	4	3	2	1	0
	WDTBnRUN[7:0]							
Value after reset	0	0	1	0	1	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.8 WDTBnWDTE Register Contents

Bit Position	Bit Name	Function
7 to 0	WDTBnRUN[7:0]	The WDTB trigger is generated by writing fixed start code (AC_H), and start/restart of the WDTB counter is controlled. An error occurs when values other than AC_H are written. After it starts, WDTB counter cannot be stopped.

The values after reset of the WDTB0RUN[7] bit, which differ according to the start-up option, are indicated in the following table.

Table 28.9 WDTB0RUN[7] Value After Reset

Start-up Options	WDTB0RUN[7] Value after Reset
OPWDRUN	
1	1
0	0

28.3.2.2 WDTBnMD — WDTB Mode Register

This register specifies the overflow interval time, enabling or disabling of the interrupt, window-open function mode selection, and the window-open period.

Value after reset: (1) WDTB0

This value depends on a set value of the OPBT0.

The value after reset depends on the setting of OPWDWMS bit, OPWDOVF[2:0] bits, OPWDINT bit, and OPWDWS[1:0] bits.

The value of OPWDWMS bit is set to 0_B; the values of OPWDOVF[2:0] bits, OPWDINT bit and OPWDWS[1:0] bits are set to 1_B because the initial state of WDTB0 set by the option byte at shipment is the same as that of WDTBn (n = 1).

Therefore, the register value after reset of WDTB0 at shipment is 7F_H.

(2) WDTBn (n = 1)

The value after reset is 7F_H.

Bit	7	6	5	4	3	2	1	0
	WDTBnWMS	WDTBnOVF[2:0]			WDTBnWIE	—	WDTBnWS[1:0]	
Value after reset	0	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Table 28.10 WDTBnMD Register Contents (1/2)

Bit Position	Bit Name	Function																																				
7	WDTBnWMS	Window Open function mode selection 0: Window size of Window Open function is set by WDTBnWS[1:0]. WDTBnTIT outputs when the counter reaches 75% of the overflow interval time defined by WDTBnMD.WDTBnOVF[2:0]. 1: Window size of Window Open function is set by WDTBnWOST register. WDTBnTIT outputs when WDTBnWIS matches WDTB counter.																																				
6 to 4	WDTBnOVF[2:0]	These bits select the overflow interval time. <table border="1"> <thead> <tr> <th>WDTBn OVF2</th> <th>WDTBn OVF1</th> <th>WDTBn OVF0</th> <th>Overflow Interval Time</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>2⁹ / WDTBTCKI</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>2¹⁰ / WDTBTCKI</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>2¹¹ / WDTBTCKI</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>2¹² / WDTBTCKI</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>2¹³ / WDTBTCKI</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>2¹⁴ / WDTBTCKI</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>2¹⁵ / WDTBTCKI</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>2¹⁶ / WDTBTCKI</td> </tr> </tbody> </table>	WDTBn OVF2	WDTBn OVF1	WDTBn OVF0	Overflow Interval Time	0	0	0	2 ⁹ / WDTBTCKI	0	0	1	2 ¹⁰ / WDTBTCKI	0	1	0	2 ¹¹ / WDTBTCKI	0	1	1	2 ¹² / WDTBTCKI	1	0	0	2 ¹³ / WDTBTCKI	1	0	1	2 ¹⁴ / WDTBTCKI	1	1	0	2 ¹⁵ / WDTBTCKI	1	1	1	2 ¹⁶ / WDTBTCKI
WDTBn OVF2	WDTBn OVF1	WDTBn OVF0	Overflow Interval Time																																			
0	0	0	2 ⁹ / WDTBTCKI																																			
0	0	1	2 ¹⁰ / WDTBTCKI																																			
0	1	0	2 ¹¹ / WDTBTCKI																																			
0	1	1	2 ¹² / WDTBTCKI																																			
1	0	0	2 ¹³ / WDTBTCKI																																			
1	0	1	2 ¹⁴ / WDTBTCKI																																			
1	1	0	2 ¹⁵ / WDTBTCKI																																			
1	1	1	2 ¹⁶ / WDTBTCKI																																			
3	WDTBnWIE	Enables/disables the interrupt request WDTBnTIT 0: WDTBnTIT is disabled 1: WDTBnTIT is enabled																																				
2	—	Reserved This bit is read as 1. The write value should be 1																																				

Table 28.10 WDTBnMD Register Contents (2/2)

Bit Position	Bit Name	Function															
1, 0	WDTBnWS[1:0]	These bits select the period over which the window is open. If WDTBnWMS = 1, this register is disabled.															
		<table border="1"> <thead> <tr> <th>WDTBnWS1</th> <th>WDTBnWS0</th> <th>Window-Open Period</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>25%</td> </tr> <tr> <td>0</td> <td>1</td> <td>50%</td> </tr> <tr> <td>1</td> <td>0</td> <td>75%</td> </tr> <tr> <td>1</td> <td>1</td> <td>100%</td> </tr> </tbody> </table>	WDTBnWS1	WDTBnWS0	Window-Open Period	0	0	25%	0	1	50%	1	0	75%	1	1	100%
WDTBnWS1	WDTBnWS0	Window-Open Period															
0	0	25%															
0	1	50%															
1	0	75%															
1	1	100%															

NOTE

WDTBnMD register update timing

- Default start mode: The WDTBnMD register cannot be updated after any reset.
- Software trigger start mode: The WDTBnMD register can be updated only once before the first write to the trigger register.

The WDTBnWS[1:0] setting is applied after the first trigger.

28.3.2.3 WDTBnWOST — WDTB Window Open Start Register

This register configures the start time of the open-window period.

Value after reset: (1) WDTB0

This value depends on the set value of OPBT0.

The value after reset depends on the settings of the OPWDWOST[15:0] bits.

The values of the OPWDWOST[15:0] bits are set to 0010_H because the initial state of WDTB0 set by the Option byte at shipment is the same as that of WDTBn (n = 1).

Therefore, the register value after reset of WDTB0 at shipment is 0010_H.

(2) WDTBn (n = 1)

The register value after reset is also 0010_H.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTBnWOST[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.11 WDTBnWOST Register Contents

Bit position	Bit name	Function
15 to 0	WDTBnWOST[15:0]	If WDTBnMD.WDTBnWMS = 1, this register is enabled. When WDTB counter value \geq WDTBnWOST, WDTB is inside the window-open period.

NOTE

WDTBnWOST register update timing

- Default start mode: WDTBnWOST register cannot be updated after any reset.
- Software trigger start mode: WDTBnWOST register can be updated only once before first WDTB trigger.

The WDTBnWOST register must be set to a value greater than 0010_H (when WDTBnMD.WDTBnWMS = 1).

The WDTBnWOST register setting is applied after the first trigger.

28.3.2.4 WDTBnWIS — WDTB Interrupt Output Timing Setting Register

This register configures the timing of WDTB interrupt (WDTBnTIT) generation.

Value after reset: (1) WDTB0

This value depends on a set value of OPBT1.

The value after reset depends on the settings of the OPWDWWIS[15:0] bits.

The values of the OPWDWWIS[15:0] bits are set to 0000_H because the initial state of WDTB0 set by the option byte at shipment is the same as that of WDTBn (n = 1).

Therefore, the register value after reset of WDTB0 at shipment is 0000_H.

(2) WDTBn (n = 1)

The register value after reset is also 0000_H.

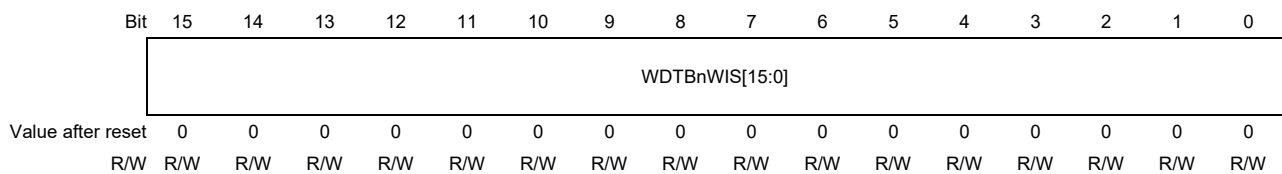


Table 28.12 WDTBnWIS Register Contents

Bit position	Bit name	Function
15 to 0	WDTBnWIS[15:0]	If WDTBnMD.WDTBnWMS = 1 and WDTBnMD.WDTBnWIE = 1, this register is enabled. When WDTB counter value = WDTBnWIS[15:0], interrupt request WDTBnTIT is generated.

NOTE

WDTBnWIS register update timing

- Default start mode: The WDTBnWIS register cannot be updated after any reset.
- Software trigger start mode: The WDTBnWIS register can be updated only once before the first WDTB trigger.

CAUTION

The register value after reset is 0000_H.

When using this register, it must be set to a value greater than 0001_H.

28.4 Operation

28.4.1 WDTB after Reset Release

28.4.1.1 Start Modes

The WDTB provides two modes for the counter start after release from the reset state:

- Software trigger start mode (for all WDTB)

The counter value remains 0000_H after reset release

The counter is started with the first WDTB trigger.

- Default start mode (only for WDTB0)

The counter starts automatically after release from the reset state. However, default start mode is disabled in serial programming mode even if OPWDRUN is set to 1_B for the start-up options.

WDTB other than WDTB0 are fixed to software trigger start mode, and cannot be set by the option byte.

28.4.1.2 Start Mode Selection (Only for WDTB0)

The start mode can be selected by the start-up options.

Table 28.13 shows the selection of the start mode.

Table 28.13 Start Modes

Start-up Option	Start Mode
OPWDRUN	
0	Software trigger
1	Default

28.4.1.3 WDTB Settings after Reset Release

The following table shows the settings of WDTB after reset release.

Table 28.14 Setting of WDTB after Reset Release

Function	Setting after WDTB0 is Reset	Setting after WDTB other than WDTB0 is Reset
Start mode	Specified by the start-up option.	Software trigger mode
Overflow interval time	Specified by the start-up option.	2 ¹⁶ / WDTBTCKI
Interrupt mode	Specified by the start-up option.	Interrupt disabled
Window-open period	Specified by the start-up option.	100%
Window open function mode selection	Specified by the start-up option.	Window size of Window Open function is set by WDTBnMD.WDTBnWS[1:0] and interrupt is generated when the counter reaches 75% of the overflow interval time defined by WDTBnMD.WDTBnOVF[2:0].
Counter value of window open start	Specified by the start-up option.	0010 _H
Counter value of WDTB interrupt output timing	Specified by the start-up option.*1	0000 _H *1

Note 1. When using this register, it must be set to a value greater than 0001_H.

For details on the option bytes, see **Section 41, Flash Memory**.

Change WDTB settings

In default start mode, WDTB cannot be changed after any reset.

In software trigger start mode, WDTB can be changed only once before the first WDTB trigger.

28.4.1.4 Default Start Mode Timing (Only for WDTB0)

Figure 28.2 shows the timing of the default start mode.

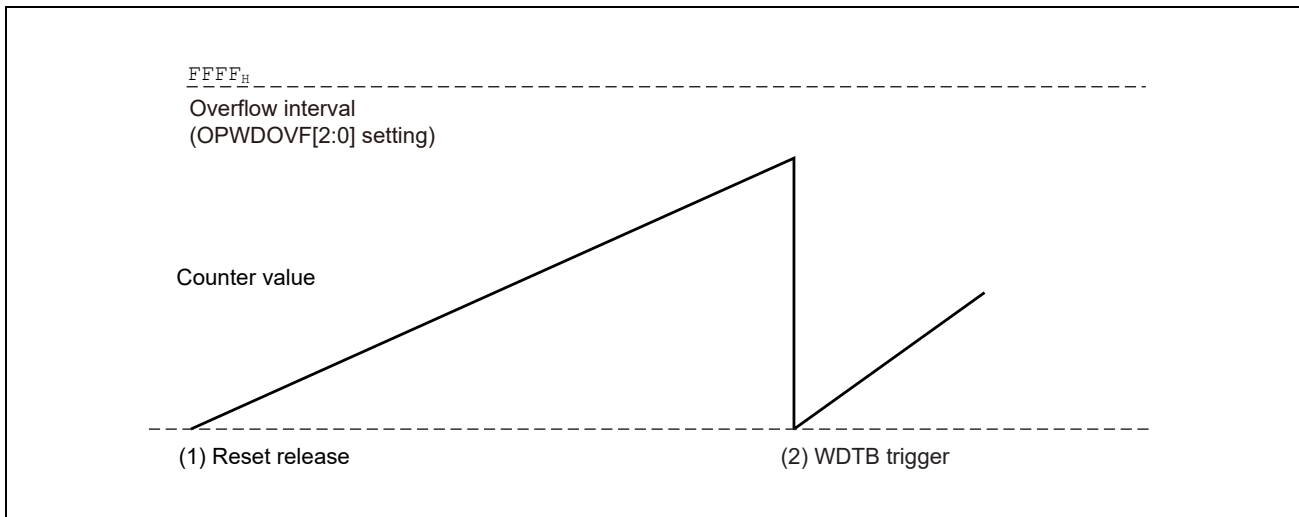


Figure 28.2 Timing Diagram of WDTB Start in Default Start Mode

The timing chart of **Figure 28.2** shows the following behaviors:

- (1) In default start mode, the WDTB counter starts after release from the reset state. The overflow interval time after release from the reset state is set by start-up options.

Example: overflow interval time after release from the reset state

$$= 2^{16} / \text{WDTBTCKI} (\text{OPWDOVF}[2:0] = 111_{\text{B}})$$

In default start mode, after reset release, the WDTBnMD register cannot be changed because WDTBnRUN = 1.

- (2) Write to the WDTB trigger register before the WDTB counter overflows.
The WDTB counter restarts with the WDTB trigger.

28.4.1.5 Software Trigger Start Mode Timing (Common to all WDTB)

Figure 28.3 shows the timing of the software trigger start mode and the change to the WDTB setting.

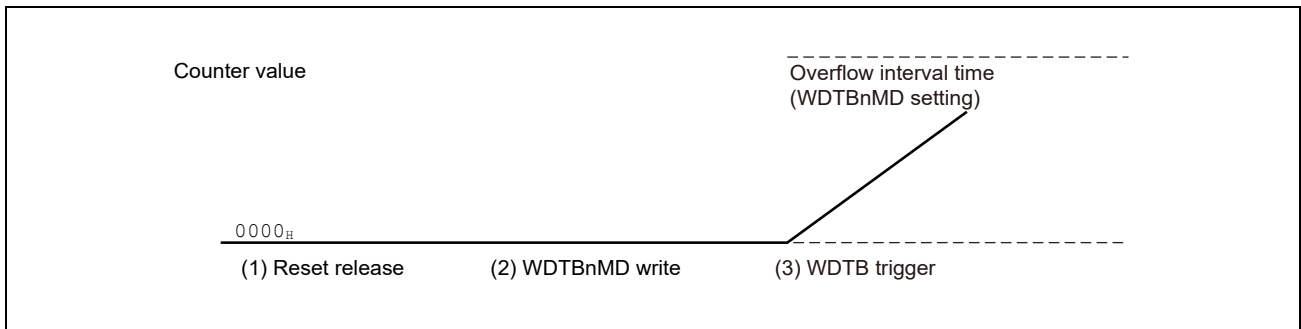


Figure 28.3 Timing Diagram of WDTB Start in Software Trigger Start Mode

The timing diagram in **Figure 28.3** shows the following:

- (1) After release from the reset state, the counter remains 0000_H until the first WDTB trigger. The overflow interval time is set by using the start-up options, but it does not have any effect.
- (2) WDTBnMD is set before the first WDTB trigger. However, the settings are not applied immediately.
- (3) The WDTB counter starts at the first WDTB trigger. The overflow interval time and other settings specified in WDTBnMD are applied.

28.4.2 WDTB Trigger

The WDTB trigger is generated in WDTB enable register (WDTBnWDTE) by writing the specific value that is called a start code.

The WDTB trigger has the following functions:

- Starting the WDTB counter in software trigger start mode
- Restarting the WDTB counter
- Setting the WDTB mode specified by the WDTBnMD register in software trigger start mode (only for the first WDTB trigger after release from the reset state).

Only the WDTBnWS[1:0] setting by WDTBnMD register in default start mode.

The WDTB can be triggered by writing a fixed activation code to the trigger register.

Table 28.15 Trigger Register and Activation Code

Type of Activation Code	Trigger Register	Activation Code
Fixed	WDTBnWDTE	AC _H

28.4.3 WDTB Error Detection

The WDTB detects an overflow of the WDTB counter and illegal operations as an error.

The conditions for error detection are:

- WDTB counter overflow
- Wrong activation code is written to the WDTB trigger register
- Writing to the trigger register outside the window-open period
- Illegal update of WDTBnMD, WDTBnWOST and WDTBnWIS registers

The following table describes illegal updates of WDTBnMD, WDTBnWOST and WDTBnWIS registers.

Table 28.16 Condition of WDTB Error Detection

Start Mode	Condition			Error Detection
	WDTBnRUN	Register Write First or Second	Write Value	
Default	0	This condition does not happen.		
	1	—	Same	Not Detect
		—	Another	Detect
Software Trigger	0	First	Same	Not Detect
			Another	Not Detect
		Second	Same	Not Detect
			Another	Detect
	1	—	Same	Not Detect
		—	Another	Detect

28.4.3.1 WDTB Error Mode

When an error is detected, it is notified to the ECM by the WDTBnTRES signal.

Figure 28.4 shows the generation of a reset when the counter overflows when the default start mode has been selected.

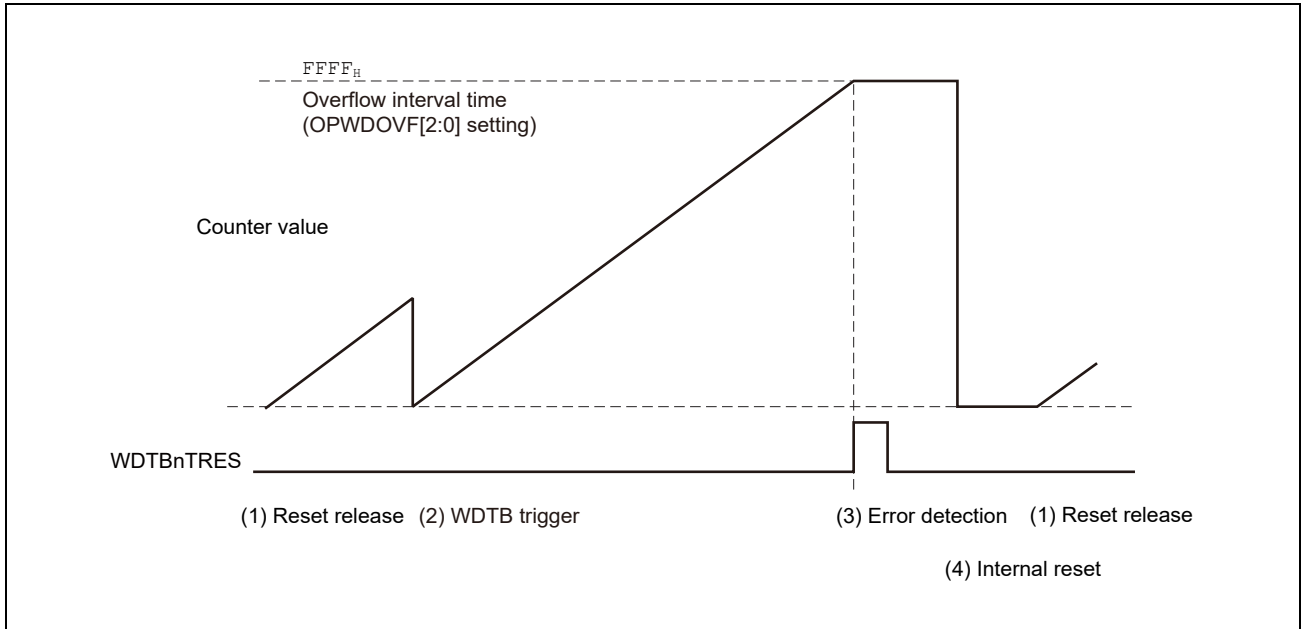


Figure 28.4 Timing Diagram of WDTB Internal Reset Generation

The timing diagram of **Figure 28.4** shows the following:

- (1) After reset is released, the WDTB counter is started in the default start mode. The overflow interval time after reset is released is set by the start-up option.
- (2) The counter restarts with the WDTB trigger.
- (3) The error is detected when the counter overflows.

Errors are notified to the ECM by the WDTBnTRES signal. The counter value does not change until internal reset is generated.

- (4) The WDTB counter stops until the counter is cleared when an internal reset is generated by a factor such as ECM, and reset is released.

28.4.4 WDTB Interrupt Output

When the WDTB counter reaches 75% of the overflow interval time and WDTBnWMS = 0, interrupt request WDTBnTIT is generated.

When the WDTB counter = WDTBnWIS and WDTBnWMS = 1, an interrupt request WDTBnTIT is generated.

This function can be selected by the WDTBnMD.WDTBnWIE register.

Figure 28.5 and **Figure 28.6** show the generation of the interrupt request under the following condition:

- The default start mode is selected.
- Interrupt request is enabled after the first WDTB trigger is generated.
- WDTB Overflow interval time: $2^{16} / \text{WDTBTCKI}$

• When WDTBnWMS = 0:

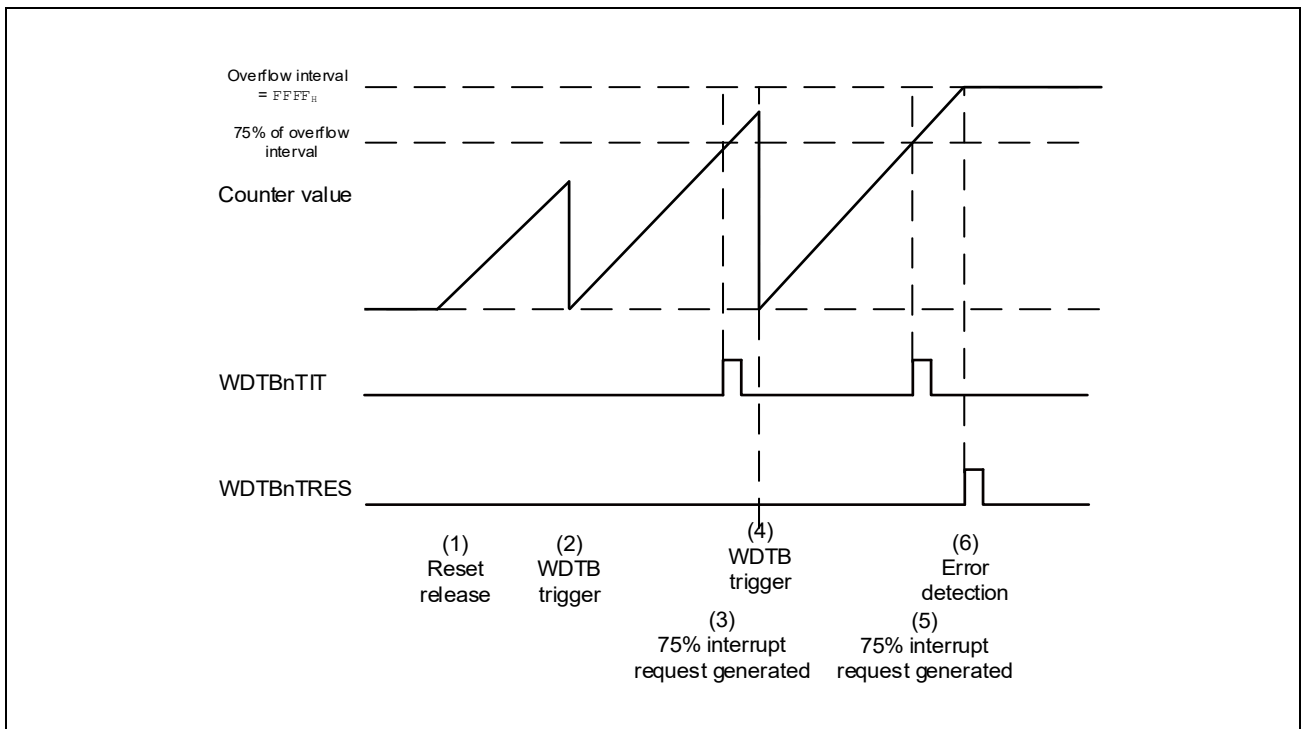


Figure 28.5 Timing Diagram of WDTB Interrupt Request Signals (WDTBnWMS = 0)

- (1) After reset is released, the WDTB counter is started in the default start mode. The overflow interval time after reset is released is set by the start-up option.
- (2) The counter restarts with the WDTB trigger.
- (3) When the WDTB counter reaches 75% of the overflow interval time, interrupt request WDTBnTIT is generated.
- (4) The counter restarts with the WDTB trigger.
- (5) When the WDTB counter reaches 75% of the overflow interval time, interrupt request WDTBnTIT is generated.
- (6) An error is detected when the counter overflows.
The error is notified to the ECM by the WDTBnTRES signal. The counter value does not change until an internal reset is generated.

• When WDTBnWMS = 1:

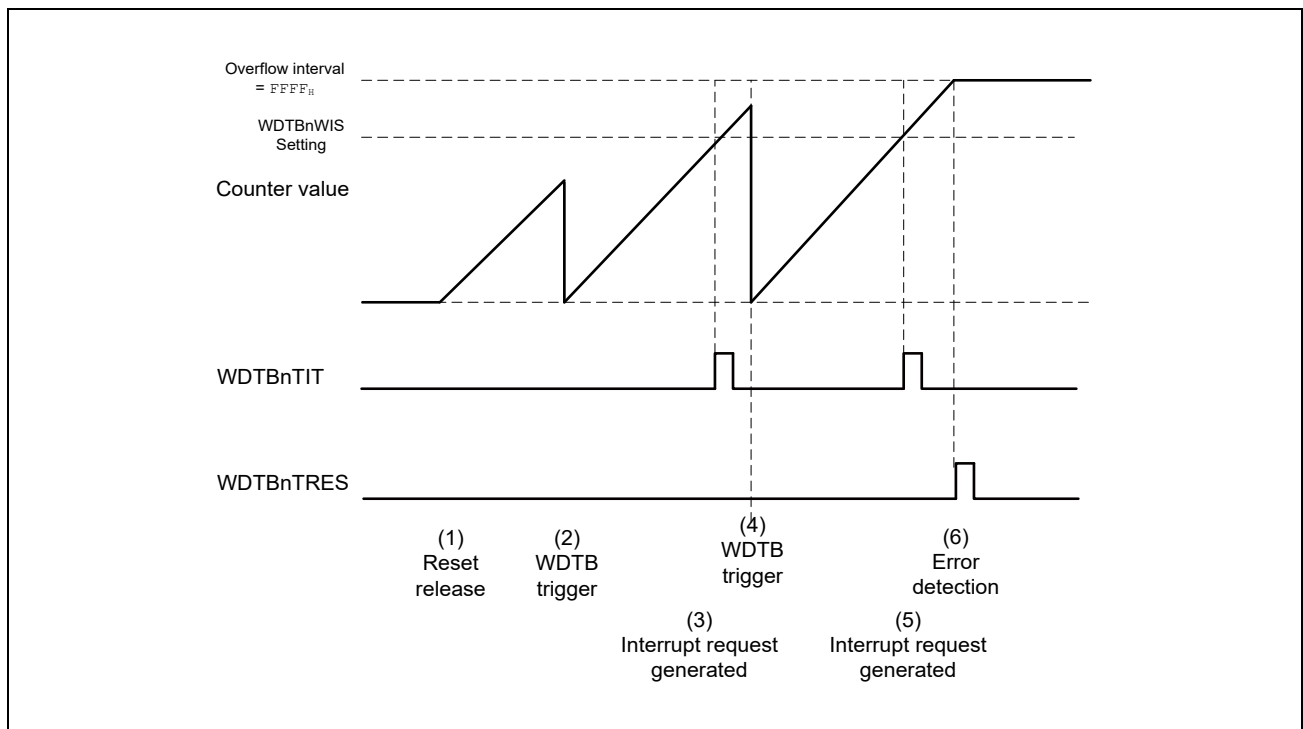


Figure 28.6 Timing Diagram of WDTB Interrupt Request Signals (WDTBnWMS = 1)

- (1) After reset is released, the WDTB counter is started in the default start mode. The overflow interval time after reset is released is set by the start-up option.
- (2) The counter restarts with the WDTB trigger.
- (3) When the WDTB counter value = WDTBnWIS[15:0], interrupt request WDTBnTIT is generated.
- (4) The counter restarts with the WDTB trigger.
- (5) When the WDTB counter value = WDTBnWIS[15:0], interrupt request WDTBnTIT is generated.
- (6) An error is detected when the counter overflows.
The error is notified to the ECM by the WDTBnTRES signal. The counter value does not change until an internal reset is generated.

28.4.5 Window Function

The period when a WDTB trigger is valid (window-open period) can be set. If the window-open period is set to a value less than 100%, an error occurs when the WDTB trigger is generated outside of the window-open period.

The window-open period after reset release is 100%. The period is set to the value configured by the WDTBnMD.WDTBnWS[1:0] or the WDTBnWOST register setting after the first WDTB trigger is generated.

Figure 28.7 and **Figure 28.8** show the window function operation under the following conditions.

- The default start mode is selected.
- 25% window-open period is effective after the first WDTB trigger (WDTBnWS[1:0] = 00_B).
- WDTB Overflow interval time: $2^{16} / \text{WDTBTCKI}$

• When WDTBnWMS = 0:

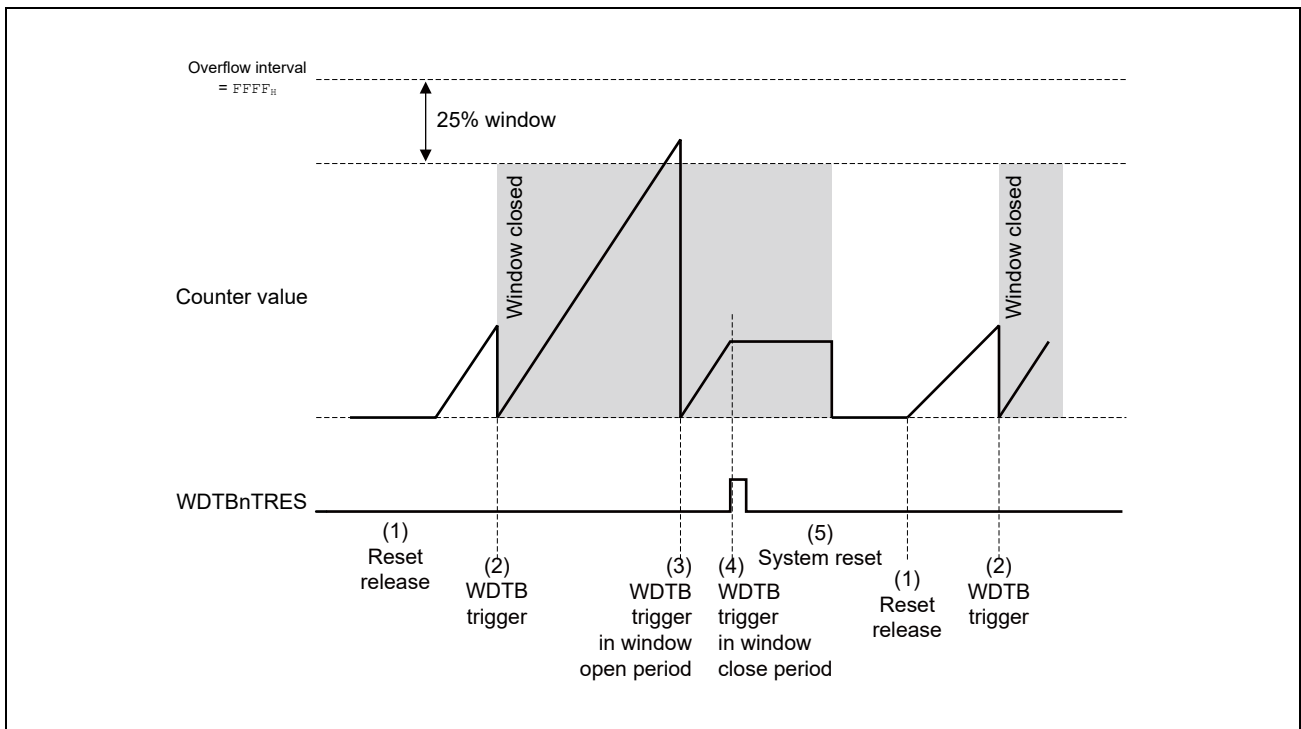


Figure 28.7 Timing Diagram of WDTB Window Function

- (1) In default start mode, the WDTB counter starts after reset is released.
The overflow interval time after reset is released is set by the start-up option.
- (2) The WDTBnMD.WDTBnWS[1:0] setting is applied at the WDTB trigger.
- (3) The WDTB counter restarts at the WDTB trigger during the window-open period.
- (4) An error is detected at the WDTB trigger during the window-close period and then the WDTBnTRES signal indicates the error the ECM.
The counter value remains unchanged until a System reset is performed.
- (5) If an internal reset occurs due to the ECM or some other sources, the counter is cleared and stops until release from the reset state.

• When WDTBnWMS = 1:

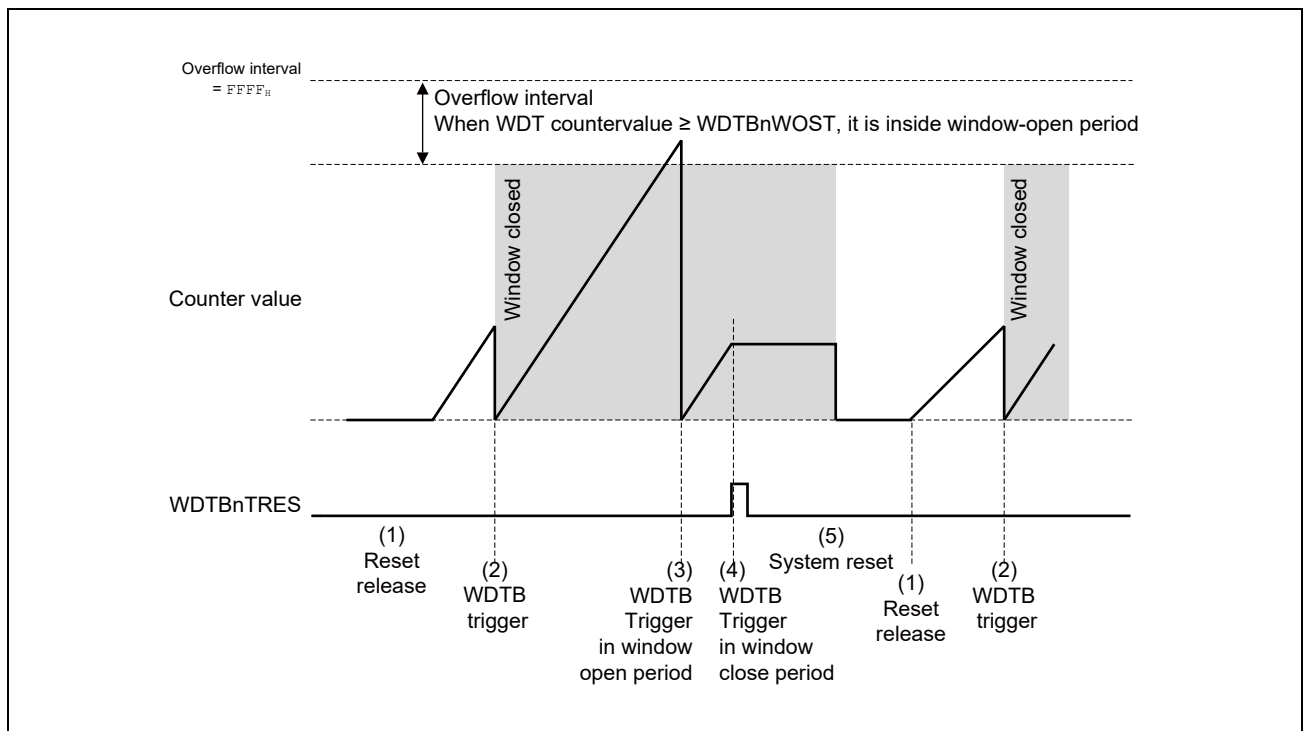


Figure 28.8 Timing Diagram of WDTB Window Function

- (1) In default start mode, the WDTB counter starts after reset is released.
The overflow interval time after reset is released is set by the start-up option.
- (2) The WDTBnWOST register setting is applied at the WDTB trigger.
- (3) The WDTB counter restarts at the WDTB trigger during the window-open period.
- (4) An error is detected at the WDTB trigger during the window-close period and then the WDTBnTRES signal indicates the error the ECM.
The counter value remains unchanged until a System reset is performed.
- (5) If an internal reset occurs due to the ECM or some other source, the counter is cleared and stops until release from the reset state.

Section 29 OS Timer (OSTM)

29.1 Features of OSTM

29.1.1 Number of Units

This microcontroller has the following number of OSTM units.

Each OSTM unit has one channel interface.

Table 29.1 Number of Units

Product Name	RH850/E2x-FCC1 Series	
	373 Pins	292 Pins
Number of Units	3	
Name	OSTMn (n = 0 to 2)	

Meaning of n

Throughout this section, the individual units of the OS timer are identified by the index “n” (n = 0 to 2), for example, OSTMnTO for the OS timer n output register.

29.1.2 Register Base Addresses

All OS timer register addresses are given as address offsets from the individual base addresses <OSTMn_base>.

The register base address of each OSTMn is listed in the following table.

Table 29.2 Register Base Addresses

Base Address Name	Base Address	Bus Group
<OSTM0_base>	FFEC 0000 _H	Peripheral Group 2L
<OSTM1_base>	FFEC 1000 _H	Peripheral Group 2L
<OSTM2_base>	FFEC 2000 _H	Peripheral Group 2L

29.1.3 Clock Supply

Table 29.3 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
OSTMn	PCLK	CLK_LSB
	Register access clock	CLK_LSB

29.1.4 Interrupts Requests

The OS timers can generate the following interrupt requests.

Table 29.4 OSTMn Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Description	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
INTOSTM0TINT	OSTMnTINT (n = 0)	OSTM0 interrupt	18	group0-201	group1-109
INTOSTM1TINT	OSTMnTINT (n = 1)	OSTM1 interrupt	CPU0-19	group0-202	group1-110
INTOSTM2TINT	OSTMnTINT (n = 2)	OSTM2 interrupt	CPU1-19	group0-203	group1-111

29.1.5 Reset Sources

OSTM reset sources are listed in the following table.

Table 29.5 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
OSTMn	All registers	✓	✓	✓	✓	—	—

29.1.6 Correspondence of OSTM and CPU

Table 29.6 Correspondence of OSTM and CPU

OSTMn	CPU Unit
OSTM0*1	all CPUs
OSTM1	CPU0
OSTM2	CPU1

Note 1. The global timer can be accessed from all CPU units.

29.2 Overview

29.2.1 Function Overview

- OSTM has two operating modes.
 - Interval timer mode
 - Free-run compare mode
- Simultaneous starting of multiple units of the OS timer (timer synchronization of PIC1)
- Interrupt and sDMA/DTS notification when counting by the counter is started, restarted, or ends
- Notifying the ECM of errors when an OSTMn (n = other than 0) interrupt occurs
- Generation of output waveform from the ERROROUT_M and ERROROUT_C pins with the OSTM0TTOUT signal of OSTM0 when ECM is in dynamic mode
- Counter start and restart of OSTM can be performed by software.
- The counter value at counter operation start can be set by software.
- Interrupt request of OSTM will be notified to the corresponding CPU.
 - Interrupt request of OSTM0 assigned to all CPUs
 - Interrupt request of OSTM1 assigned to CPU0
 - Interrupt request of OSTM2 assigned to CPU1
- Module standby can be set for each unit (see **Section 16, Standby Controller**, for details).

29.2.2 Block Diagram

The following block diagram shows the main components of OSTMn.

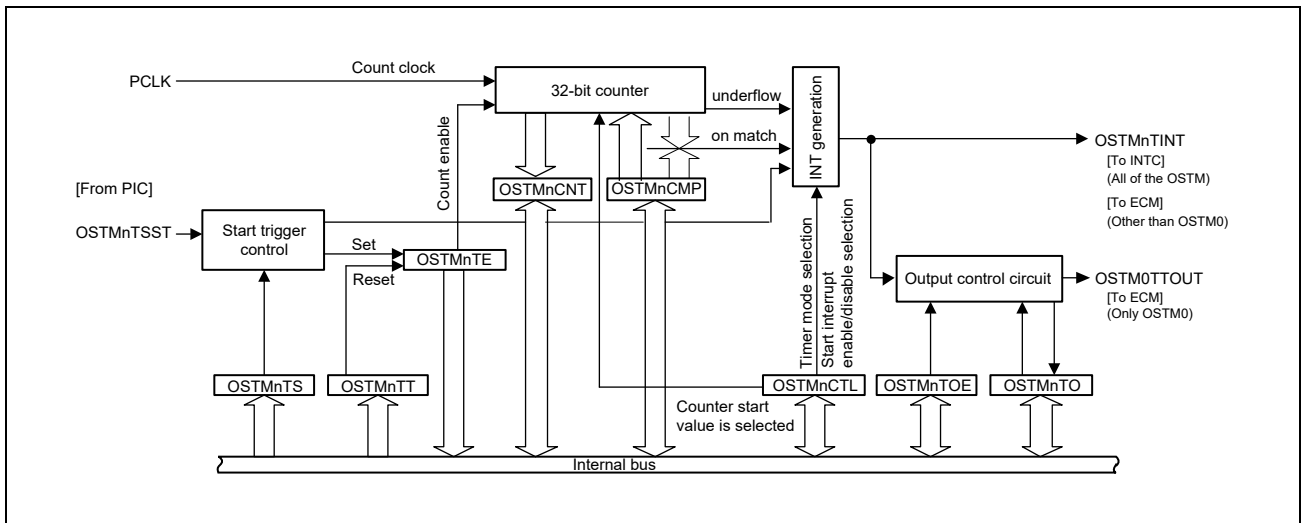


Figure 29.1 Block Diagram of OSTMn

29.2.3 Output Modes (only for OSTM0)

The OSTMn has the following output modes. The mode is selected by the setting of the OSTMnTOE.OSTMnTOE bit.

- Software control mode (the OSTMnTOE.OSTMnTOE bit is 0): The value set in the OSTMnTO.OSTMnTO bit is output to OSTMnTTOUT.
- Timer-output toggling mode (the OSTMnTOE.OSTMnTOE bit is 1): The OSTMnTTOUT output is toggled each time an OSTMnTINT request is generated.

Both output modes are illustrated in the following figure.

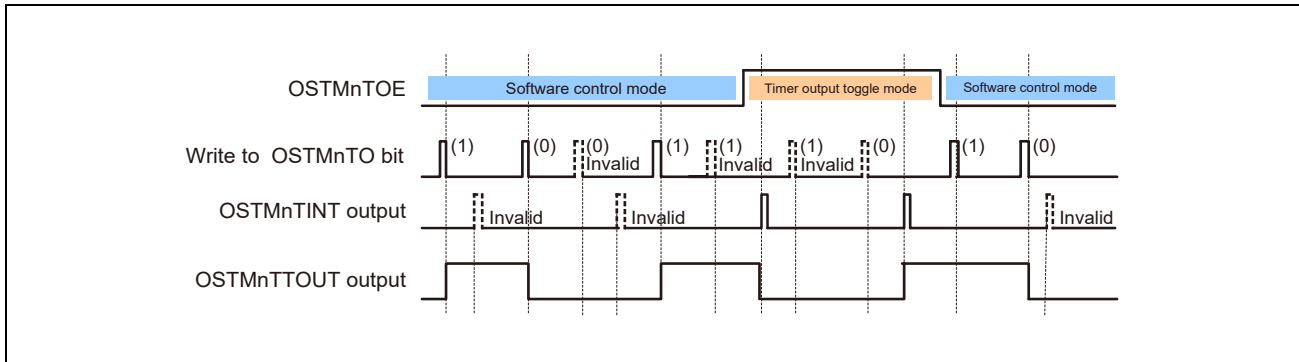


Figure 29.2 Timing Diagram of Output Modes

The above timing diagram shows the following operations.

- In software control mode, the level of the OSTMnTTOUT output changes in accordance with the value set in the OSTMnTO.OSTMnTO bit.
- In timer-output toggling mode, the value of the OSTMnTO.OSTMnTO bit and level of the OSTMnTTOUT output are toggled each time an OSTMnTINT interrupt request is generated.

29.2.4 Interrupt Requests (OSTMnTINT)

An OSTMnTINT interrupt request is generated whenever the counter reaches 0000 0000_H (in interval timer mode) or matches the comparison value (in free-run compare mode).

An interrupt request can also be generated on starting and restarting of the counter. This is controlled by the OSTMnCTL.OSTMnMD0 bit.

Since OSTMnTINT triggers toggling of the OSTMnTTOUT output in timer output toggling mode (OSTMnTOE.OSTMnTOE is 1), the setting of the OSTMnCTL.OSTMnMD0 bit also affects the output (OSTMnTTOUT).

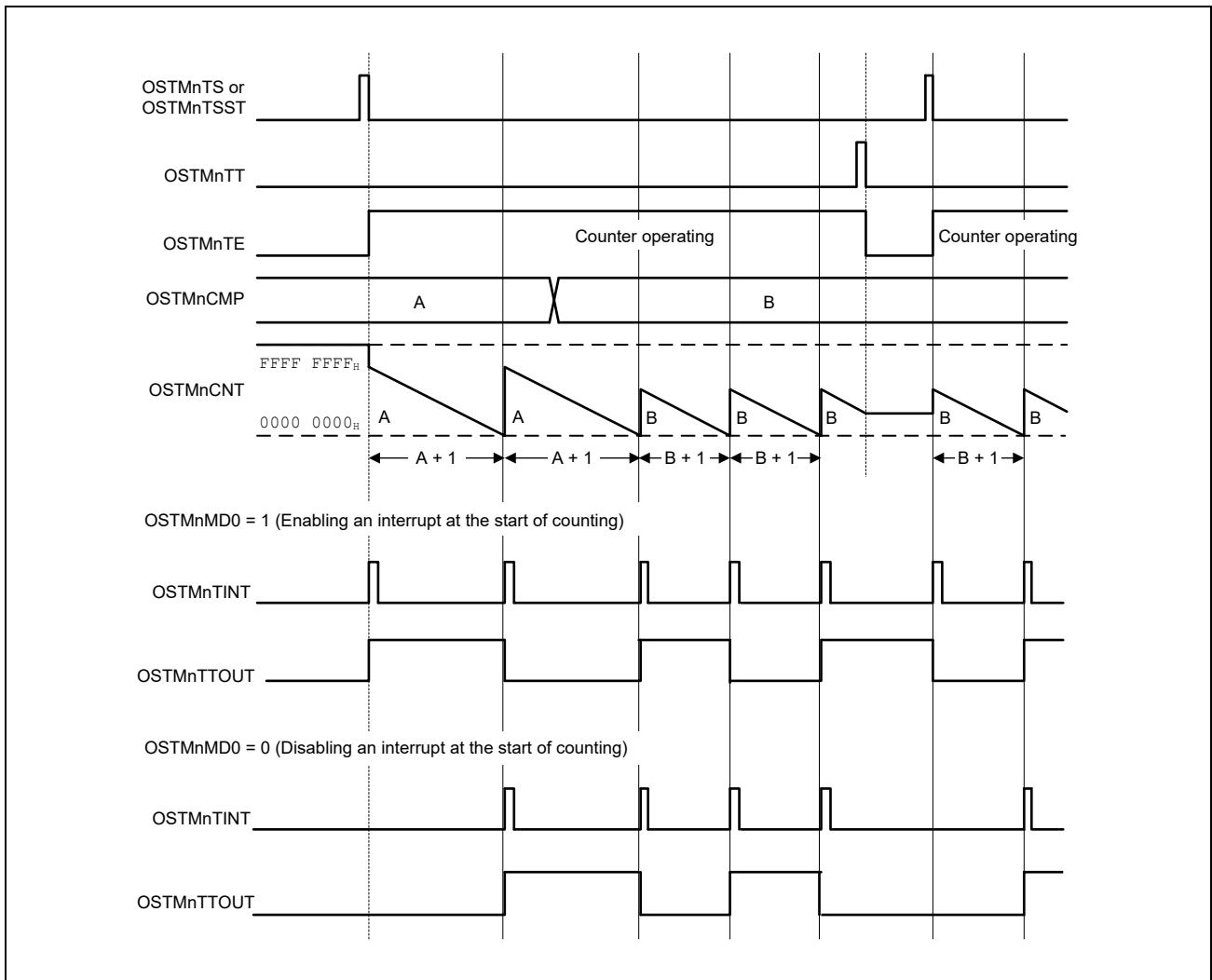


Figure 29.3 Generating an Interrupt when Counting Starts (Interval Timer Mode)

29.3 Registers

The OS timers are controlled and operated by the following registers.

29.3.1 List of Registers

The list of OSTMn registers and the memory addresses are as follows.

For the base addresses, see **Table 29.2**.

Table 29.7 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection
OSTMn	OSTM Compare Register	OSTMnCMP	<OSTMn_base> + 00 _H	32	—
	OSTM Counter Register	OSTMnCNT	<OSTMn_base> + 04 _H	32	—
	OSTM Output Register	OSTMnTO	<OSTMn_base> + 08 _H	8	—
	OSTM Output Enable Register	OSTMnTOE	<OSTMn_base> + 0C _H	8	—
	OSTM Count Enable Status Register	OSTMnTE	<OSTMn_base> + 10 _H	8	—
	OSTM Count Start Trigger Register	OSTMnTS	<OSTMn_base> + 14 _H	8	—
	OSTM Count Stop Trigger Register	OSTMnTT	<OSTMn_base> + 18 _H	8	—
	OSTM Control Register	OSTMnCTL	<OSTMn_base> + 20 _H	8	—

29.3.2 Detail of Registers

29.3.2.1 OSTMnCMP — OSTM Compare Register

Depending on the mode of operation, this register holds the start value for the down-counter or the value for comparison with that of the counter.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OSTMnCMP[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTMnCMP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.8 OSTMnCMP Register Contents

Bit Position	Bit Name	Function
31 to 0	OSTMnCMP[31:0]	In interval timer mode: Start value of the down-counter In free-run compare mode: Value for comparison

29.3.2.2 OSTMnCNT — OSTM Counter Register

This register indicates the counter value of the timer.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OSTMnCNT[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTMnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.9 OSTMnCNT Register Contents

Bit Position	Bit Name	Function
31 to 0	OSTMnCNT[31:0]	32-bit counter value

Table 29.10 lists the correspondence between the OSTM operating mode, counting direction, and start value.

The start value indicates the value to be read after the operating mode is changed.

Table 29.10 Correspondence between Operating Mode, Counting Direction and Start Value

Timer Operating Mode	OSTMnCTL.OSTMnMD1	Counting Direction	Start Value* ²
Interval timer mode	0* ¹	Down count	FFFF FFFF _H
Free-run compare mode	1	Up count	0000 0000 _H

Note 1. Value after reset

Note 2. The start value is the value that is read after the operating mode has been changed following reset input.

When OSTMnTE.OSTMnTE = 0 (counter stop status), the OSTMnCNT register can be written.

When the operation mode of the counter is changed with OSTMnCTL.OSTMnMD2 = 1, write any value to the OSTMnCNT register before the counter operation starts.

29.3.2.3 OSTMnTO — OSTM Output Register

The OSTM output register is used to specify and read the level of OSTMnTTOUT output signals. The setting of this register is only valid in OSTM0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTO
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 29.11 OSTMnTO Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
0	OSTMnTO	This bit specifies/reads the level of OSTMnTTOUT output signals. 0: Low level 1: High level

29.3.2.4 OSTMnTOE — OSTM Output Enable Register

The OSTM output enable register specifies OSTMnTTOUT output mode. The setting of this register is only valid in OSTM0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTOE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 29.12 OSTMnTOE Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
0	OSTMnTOE	This bit specifies the OSTMnTTOUT output mode. 0: Software control mode: The level corresponding to the setting of OSTMnTO.OSTMnTO is output to OSTMnTTOUT. 1: Timer-output toggling mode: OSTMnTTOUT output is toggled whenever an OSTMnTINT interrupt request is generated.

29.3.2.5 OSTMnTE — OSTM Count Enable Status Register

This register indicates whether the counter is enabled or disabled.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 29.13 OSTMnTE Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is read.
0	OSTMnTE	This bit indicates whether the counter is enabled or disabled. 0: Counter disabled 1: Counter enabled This bit is set to 1 in response to OSTMnTS.OSTMnTS being set to 1, or to the OSTMnTSST signal (when timer synchronization of PIC1 is in use) becoming 1. This bit is reset to 0 when the OSTMnTT.OSTMnTT bit is set to 1.

NOTE

When OSTMnTS.OSTMnTE = 0, the counter retains its value.

If the counter is restarted when OSTMnCTL.OSTMnMD2 = 0,

- It restarts counting down from the value in the OSTMnCMP register in interval timer mode.
- It restarts counting up from the counter value 0000 0000_H in free-run compare mode.

If OSTMnCTL.OSTMnMD2 = 1, the counter is not initialized.

29.3.2.6 OSTMnTS — OSTM Count Start Trigger Register

This register starts the counter.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 29.14 OSTMnTS Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When writing, write the value after reset.
0	OSTMnTS	This bit starts the counter. 0: This setting is invalid. 1: Starts the counter and sets OSTMnTE.OSTMnTE = 1. In interval timer mode, a forced restart is executed if this bit is set while OSTMnTE.OSTMnTE = 1. In free-run compare mode, setting this bit is ignored as long as OSTMnTE.OSTMnTE = 1.

29.3.2.7 OSTMnTT — OSTM Count Stop Trigger Register

This register stops the counter.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 29.15 OSTMnTT Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When writing, write the value after reset.
0	OSTMnTT	Stops the counter. 0: This setting is invalid. 1: Stops the counter and clears the OSTMnTE.OSTMnTE bit.

29.3.2.8 OSTMnCTL — OSTM Control Register

This register specifies the operating mode for the counter and controls enabling/disabling of OSTMnTINT interrupt requests when counting starts.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	OSTMnIE	—	—	—	—	OSTMnMD2	OSTMnMD1	OSTMnMD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R/W	R/W	R/W

Table 29.16 OSTMnCTL Register Contents

Bit Position	Bit Name	Function
7	OSTMnIE	OSTM Interrupt enable This bit is the enable bit of the OSTM Interrupt (OSTMnTINT) 0: OSTMn interrupt request is disabled. 1: OSTMn interrupt request is enabled.
6 to 3	—	Reserved When read, the value after reset is read. When writing, write the value after reset.
2	OSTMnMD2	Control of the counter operation at counter operation start This bit controls the counter operation at counter operation start regardless of whether the counter currently is disabled or enabled (OSTMnTE.OSTMnTE = 0 to 1). 0: OSTMnCNT is loaded at counter operation start The following describes the value of the counter at counter operation start. – Interval timer mode: The OSTMnCMP[31:0] value is loaded to OSTMnCNT[31:0] at counter start. – Free-run compare mode: The value of 0000 0000 _H is loaded to OSTMnCNT[31:0] at counter operation start. 1: OSTMnCNT is not loaded at counter operation start
1	OSTMnMD1	Specifies the operating mode for the counter. 0: Interval timer mode 1: Free-run compare mode
0	OSTMnMD0	Controls enabling/disabling of OSTMnTINT interrupt requests when counting starts. 0: Disables the interrupts when counting starts. 1: Enables the interrupts when counting starts.

29.4 Operation

Each OS timer is a 32-bit timer/counter.

The settings for the operating mode specify the direction of counting (up or down) and the generation of interrupt requests.

29.4.1 Starting and Stopping OSTM

The OS timer is started and stopped as follows.

Starting the timer

Figure 29.4 shows the flow of starting the counter of OSTM_n. For details on the operations of the timer in interval timer mode and free-run timer mode in a given step of the figure, see the descriptions in **Section 29.4.2, Interval Timer Mode**, and **Section 29.4.3, Free-Run Compare Mode**.

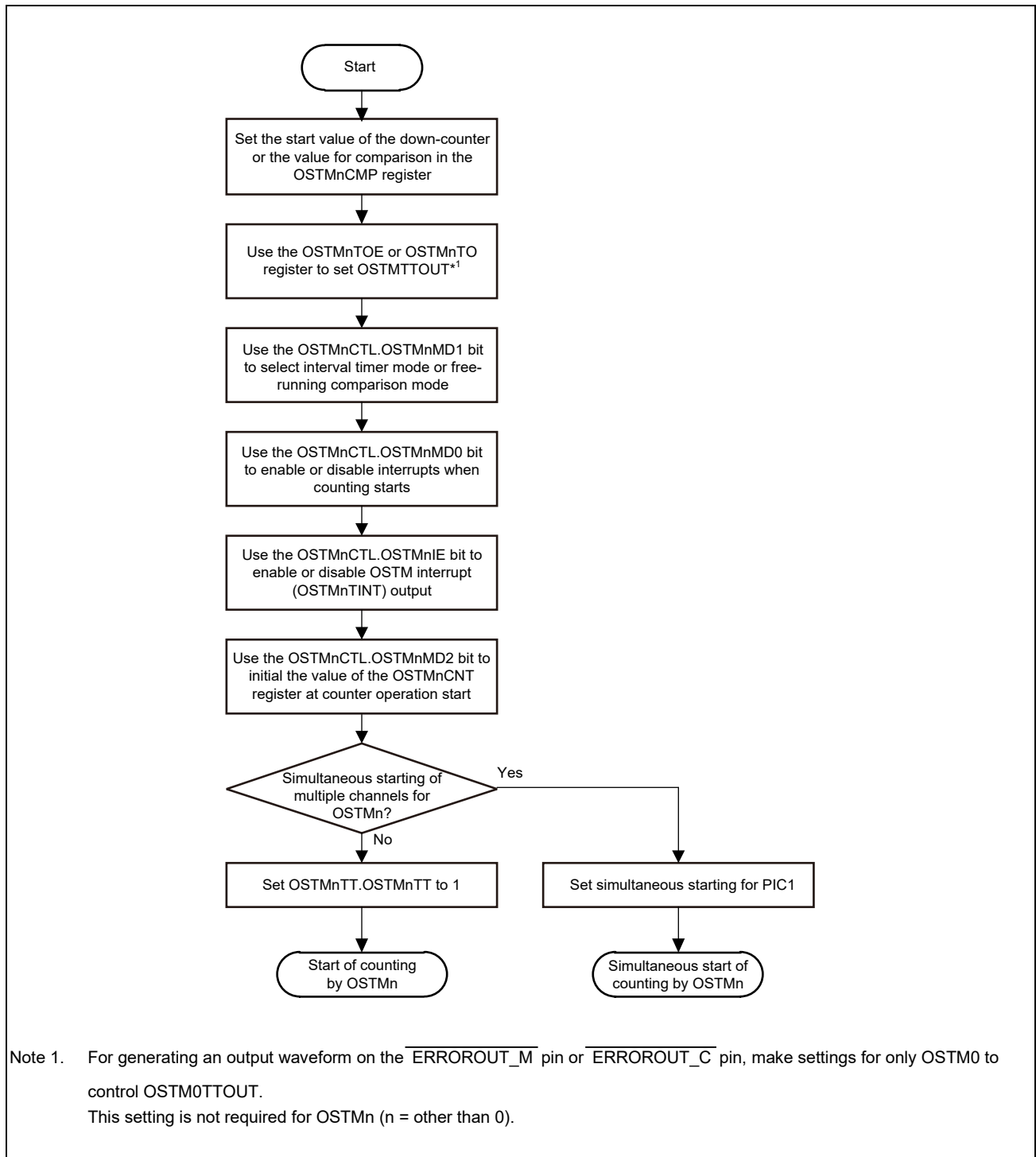


Figure 29.4 Flow of Starting the OS Timer

Stopping the timer

Setting the OSTMnTT.OSTMnTT bit to 1 stops the timer and clears the OSTMnTE.OSTMnTE status flag bit.

Timer Synchronization (PIC1)

The OSTMnTSST signal can be used to start multiple timers at the same time. For detailed connections, refer to **Section 33, Peripheral Interconnection (PIC)**.

29.4.2 Interval Timer Mode

Select the interval timer mode when an OS timer is to be used as a reference timer for generating interrupt requests at a fixed interval.

29.4.2.1 Basic Operation in Interval Timer Mode

The interval timer mode is set when count mode selection bit OSTMnMD1 is set to “0”. In the interval timer mode, the OSTM can be used as the reference timer generating the OSTMnTINT interrupt at fixed intervals. Upon occurrence of an interrupt, OSTMnTTOUT performs a toggle operation and outputs a square wave. The periods of OSTMnTINT and OSTMnTTOUT are as follows.

- OSTMnTINT generation cycle = Count-clock cycle × (OSTMnCMP + 1)
- OSTMnTTOUT output cycle = OSTMnTINT generation period × 2

OSTMnCMP can be rewritten at any timing, and if it is rewritten during OSTMnCNT operation (OSTMnTE = 1), OSTMnCNT reloads the new OSTMnCMP value upon the next 0000 0000_H match detection, and the count operation continues. Then, when “1” is written to OSTMnTT to clear OSTMnTE to “0”, the counter and OSTMnTO stops operation while holding their values as they are.

When OSTMnMD2 = 0, OSTMnCMP[31:0] value is stored to OSTMnCNT at counter operation start.

When OSTMnMD2 = 1, OSTMnCMP[31:0] value is not stored to OSTMnCNT at counter operation start.

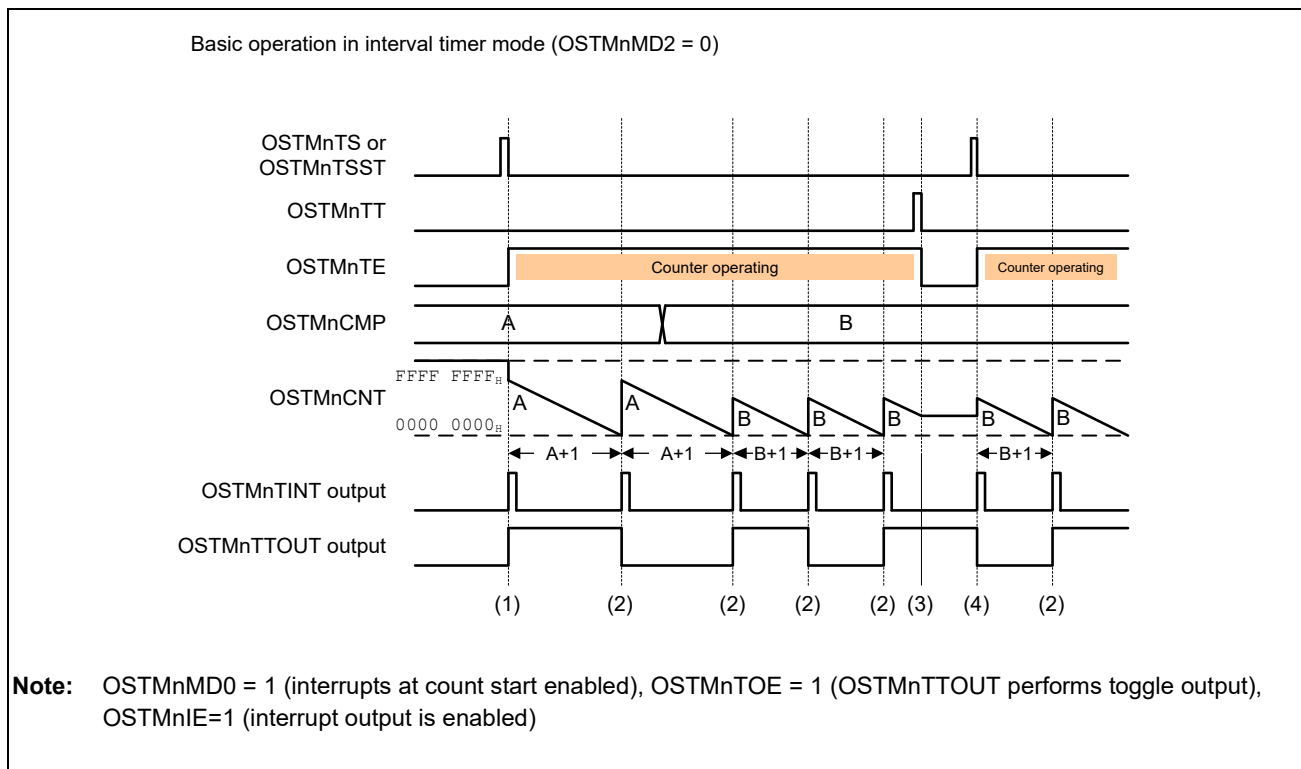


Figure 29.5 Timing Diagram of OSTMn in Interval Timer Mode (OSTMnMD2 = 0)

The timing diagram above shows the following:

- (1) The counter starts counting when OSTMnTS.OSTMnTS = 1 or OSTMnTSST = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter.
The counter starts counting-down from the value of OSTMnCMP. If OSTMnCTL.OSTMnMD0 is 1,

OSTMnTINT interrupt requests are generated at the start of counting and the OSTMnTTOUT signal is toggled. The OSTMnCNT register indicates the counter value.

- (2) When the counter reaches 0000 0000_H, an OSTMnTINT interrupt request is generated and the OSTMnTTOUT signal is toggled. The counter loads the new start value from OSTMnCMP and continues counting down.
- (3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.
- (4) When counting is restarted (OSTMnTS.OSTMnTS = 1, or OSTMnTSST = 1), the counter loads the new start value from OSTMnCMP and starts counting down.

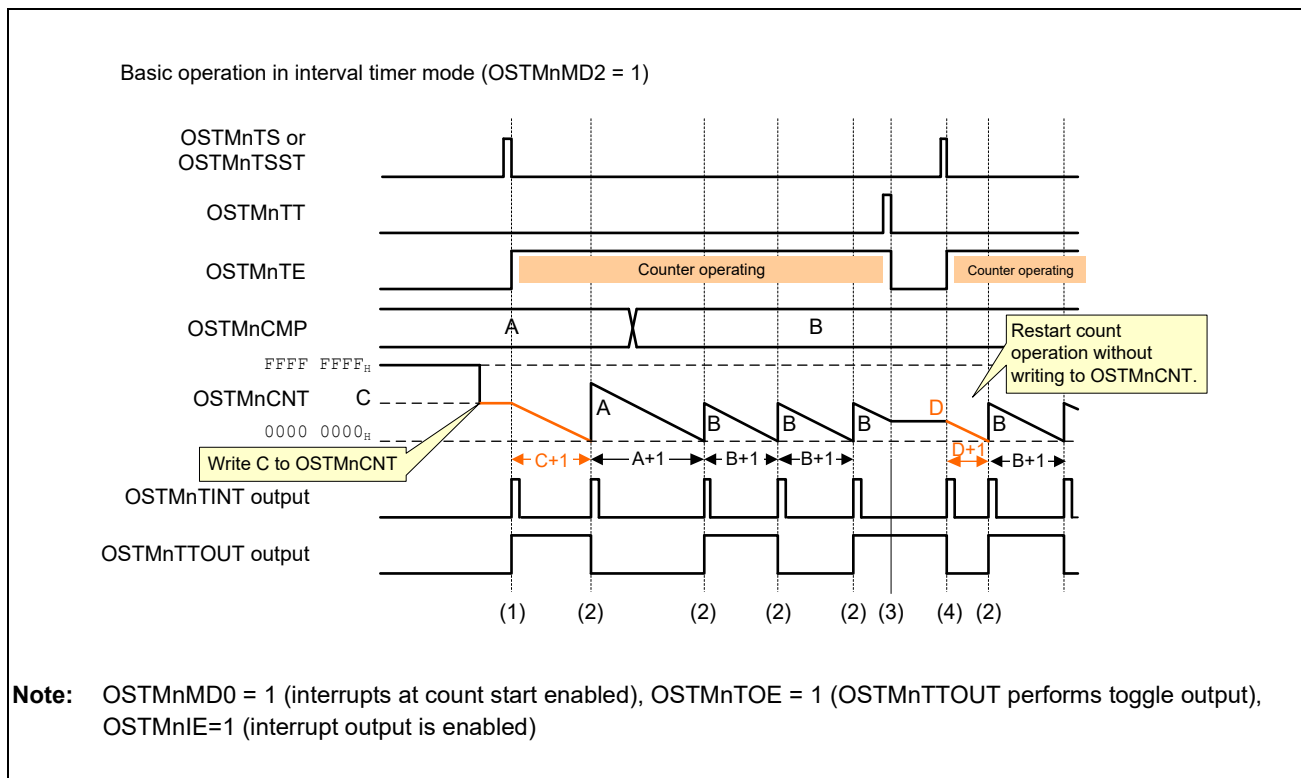


Figure 29.6 Timing Diagram of OSTMn in Interval Timer Mode (OSTMnMD2 = 1)

The timing diagram above shows the following:

- (1) The counter starts counting when OSTMnTS.OSTMnTS = 1 or OSTMnTSST = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter. The counter starts counting-down from the value written to OSTMnCNT. If OSTMnCTL.OSTMnMD0 is 1, OSTMnTINT interrupt requests are generated at the start of counting and the OSTMnTTOUT signal is toggled. The OSTMnCNT register contains the current value as the counter.
- (2) When the counter reaches 0000 0000_H, an OSTMnTINT interrupt request is generated and the OSTMnTTOUT signal is toggled. The counter loads the new start value from OSTMnCMP and continues counting down.
- (3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.
- (4) When counting is restarted (OSTMnTS.OSTMnTS = 1, or OSTMnTSST = 1), the counter keeps the value when it was stopped and starts counting down.

29.4.2.2 Operation when OSTMnCMP = 0000 0000_H

When OSTMnCMP = 0000 0000_H, the OS timer behaves as follows.

- If the counter is enabled, the OSTMnTINT interrupt request will always be set to 1. However, the timer (OSTMnTTOUT) output signal can still be used. Timer (OSTMnTTOUT) output using timer output toggling mode results in OSTMnTTOUT being toggled at every cycle of the count clock.

The following figure shows the operation of OSTMn when OSTMnCMP = 0000 0000_H, the counter start interrupt is enabled (OSTMnCTL.OSTMnMD0 = 1), and the timer is in timer output toggling mode, where OSTMnTTOUT is toggled (OSTMnTOE.OSTMnTOE = 1).

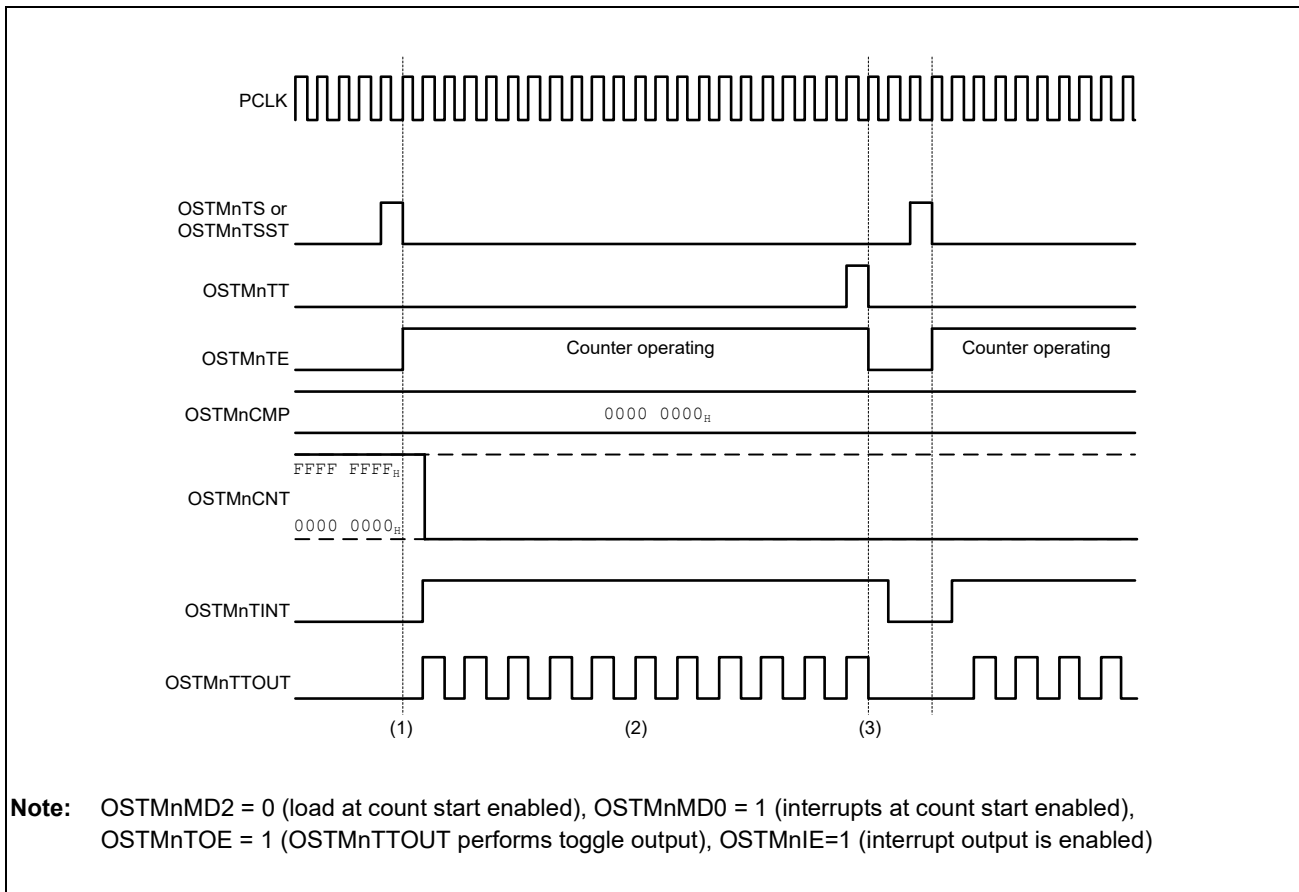


Figure 29.7 Timing Diagram when OSTMnCMP = 0000 0000_H in Interval Timer Mode

The timing diagram above shows the following operations:

- (1) The counter is reloaded with the value in OSTMnCMP as soon as it starts counting, so the value 0000 0000_H is retained in OSTMnCNT.
- (2) While the OSTMnTINT interrupt request signal is continuously asserted, OSTMnTTOUT is toggled (in **Figure 29.7**, OSTMnTINT is fixed to the high level because PCLK is selected as the count clock).
- (3) After the counter stops, the OSTMnTINT interrupt request signal is deasserted and the output level of the OSTMnTTOUT signal is maintained.

When an interrupt is prohibited when the counter starts, 1 clock cycle of the count clock is not generated at the start timing of the counter.

29.4.2.3 Forced Restart Operation in Interval Timer Mode

Forced restart operation is performed by writing “1” to the OSTMnTS count start register during the count operation (OSTMnTE = 1) in the interval timer mode. At this time, the OSTMnCNT counter immediately reloads the OSTMnCMP compare register and the count operation continues.

CAUTION

If the OSTMnTS.OSTMnTS bit for a desired channel is set to 1 while simultaneous starting of multiple channels of the timer is in progress, the individual channels are forcibly restarted, so synchronous operation between the channels is not guaranteed.

If simultaneous starting of PIC1 is used while multiple channels of the timer start individually, the timers set for simultaneous starting are forcibly restarted at the same time.

Forced restart operation is performed regardless of the value of OSTMnCTL.OSTMnMD2.

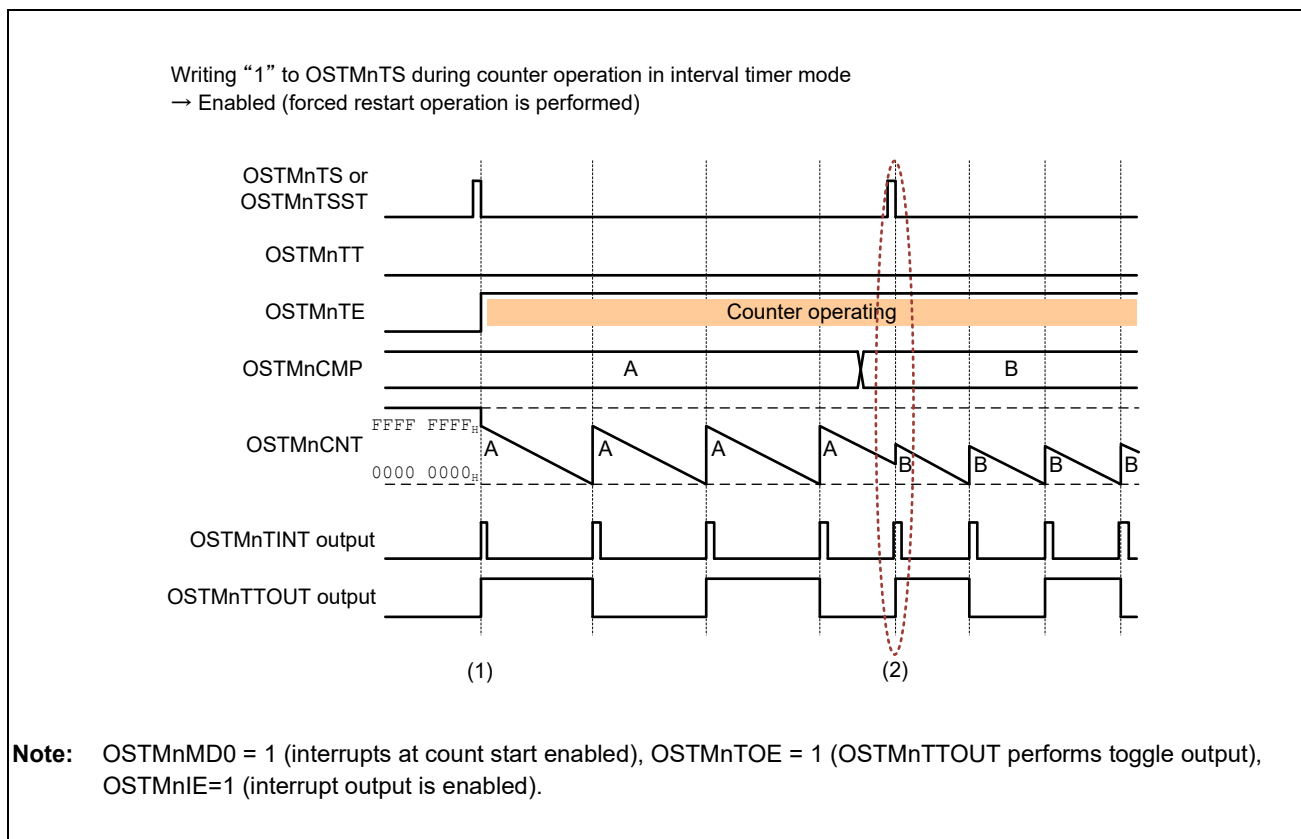


Figure 29.8 Forced Restart Operation in Interval Timer Mode

The operations shown in the above timing diagram are as follows.

- (1) The counter is started and stopped as described under **Figure 29.5** or **Figure 29.6**.
- (2) Setting OSTMnTS.OSTMnTS = 1 or OSTMnTSST = 1 restarts the counter while counting is in progress (i.e. while OSTMnTE.OSTMnTE = 1).
The counter immediately restarts counting down, starting with the current value of OSTMnCMP.
When OSTMnCTL.OSTMnMD0 = 1, an OSTMnTINT interrupt request is generated when counting starts and the OSTMnTTOUT signal is toggled.

29.4.3 Free-Run Compare Mode

29.4.3.1 Basic Operation in Free-Run Compare Mode

The free-run compare mode is set when count mode selection bit OSTMnMD1 is set to “1”. In the free-run compare mode, the OSTMnTINT interrupt is output when the value of the OSTMnCMP compare register and the count value match. OSTMnTTOUT performs a toggle operation.

When OSTMnMD2 = 0, OSTMnCNT[31:0] is cleared to 0000 0000_H at counter operation start.

When OSTMnMD2 = 1, OSTMnCNT is not cleared at counter operation start.

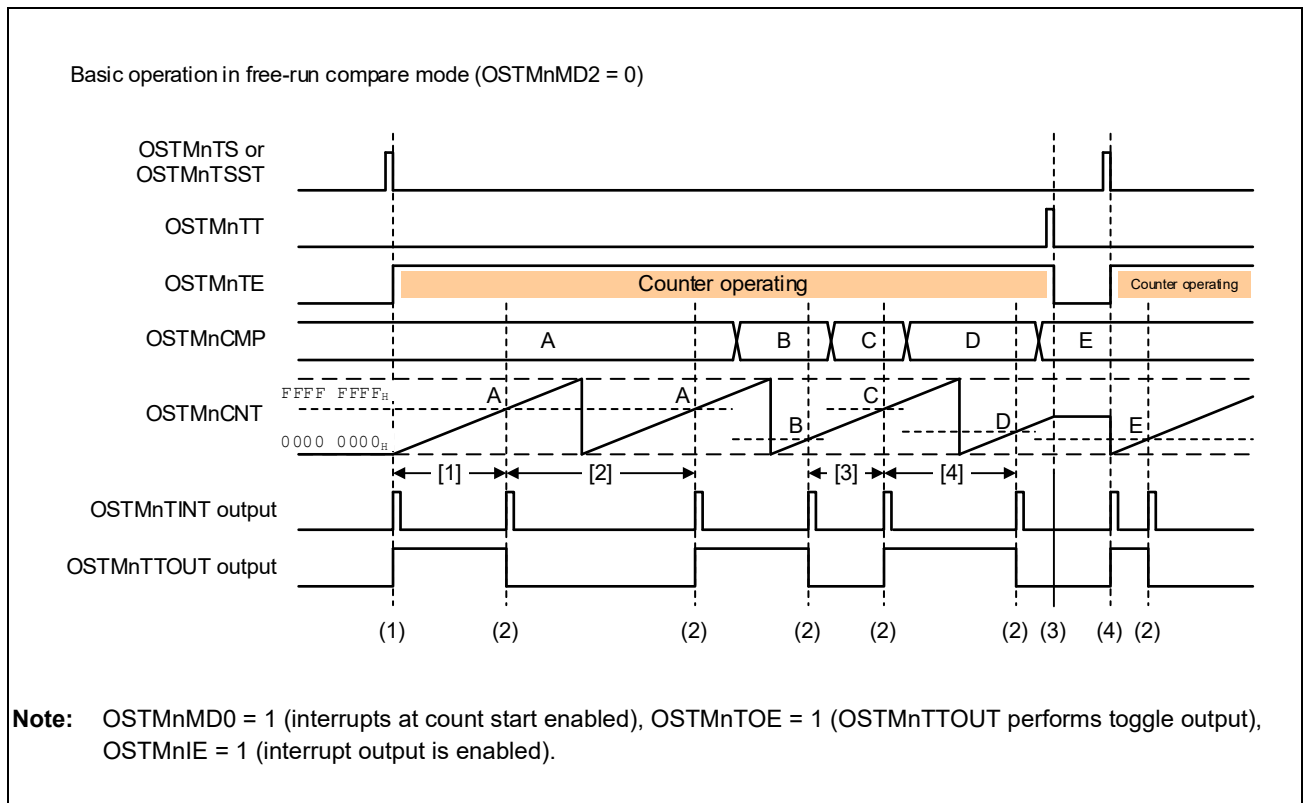


Figure 29.9 Timing Diagram of OSTMn in Free-Run Compare Mode (OSTMnMD2 = 0)

The timing diagram above shows the following:

- (1) The counter starts counting when OSTMnTS.OSTMnTS = 1, or when OSTMnTSST = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter. The counter counts up from 0000 0000_H to FFFF FFFF_H. The OSTMnCNT register indicates the counter value. When the OSTMnCTL.OSTMnMD0 set to 1, the interrupt request OSTMnTINT is generated when the counting starts.
- (2) When the current counter value matches the value in the OSTMnCMP register, an OSTMnTINT interrupt request is generated and the OSTMnTTOUT signal is toggled.
- (3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.
- (4) When restarting the counter by setting OSTMnTS.OSTMnTS=1, the counter starts from 0000 0000_H.

The OSTMnTINT and OSTMnTTOUT occurrence timing in the free-run compare mode may not be a constant period depending on the usage conditions. This occurs at count start and when OSTMnCMP is rewritten during the count operation. This is explained using **Figure 29.9**.

Table 29.17 Timing of OSTMnTINT Generation

Old value for comparison	New value for comparison	Counter value at time of rewriting	Period of OSTMnTINT Generation	Label in timing diagram
Counter starts			$(A + 1) \times \text{count clock period}$	(1)
A	A	No rewriting	$(\text{FFFF FFFF}_H + 1) \times \text{count clock period}$	(2)
B	$C > B$	$B < \text{counter value} < C$	$(C - B) \times \text{count clock period}$	(3)
C	$D < C$	Counter value $> D, C$	$(\text{FFFF FFFF}_H - C + D + 1) \times \text{count clock period}$	(4)

Even during the count operation, OSTMnCMP rewrite is immediately reflected, and OSTMnCNT match judgment is executed. When "1" is written to OSTMnTT to set OSTMnTE = 0, the counter and OSTMnTO stop operation while holding their values.

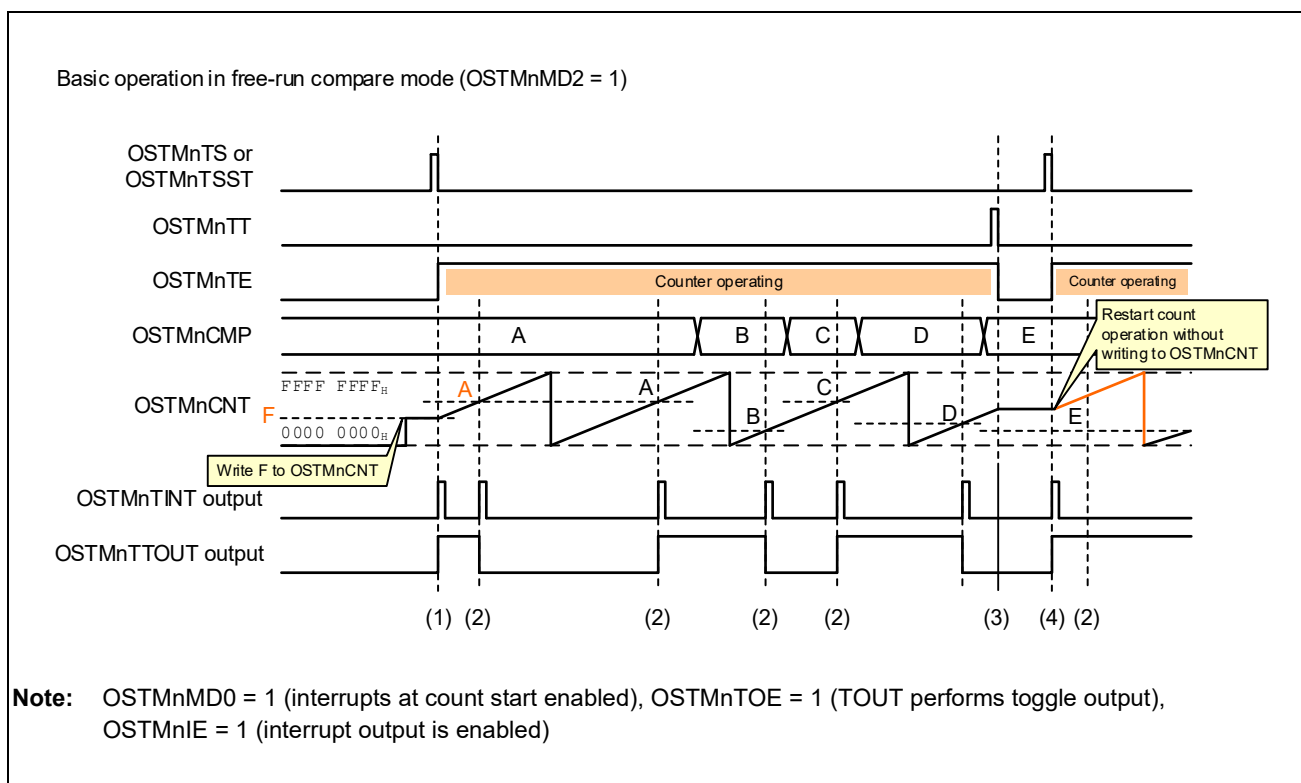


Figure 29.10 Timing Diagram of OSTMn in Free-Run Compare Mode (OSTMnMD2 = 1)

The timing diagram above shows the following:

- (1) The counter starts counting when OSTMnTS.OSTMnTS = 1, or when OSTMnTSST = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter. The counter starts count up to FFFF FFFF_H from the value written to the OSTMnCNT register. The OSTMnCNT register is the counter, so it contains the current value. When the OSTMnCTL.OSTMnMD0 set to 1, the OSTMnTINT interrupt request is generated when counting starts.
- (2) When the current counter value matches the value in the OSTMnCMP register, an OSTMnTINT interrupt request is generated and the OSTMnTTOUT signal is toggled.

- (3) When the counter is stopped ($OSTMnTT.OSTMnTT = 1$), the $OSTMnTE.OSTMnTE$ bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.
- (4) Counting by the counter restarts from the value at which counting stopped when $OSTMnTS.OSTMnTS = 1$, or when $OSTMnTSST = 1$.

29.4.3.2 Operation when $OSTMnCMP = 0000\ 0000_H$

The following figure shows the operation of $OSTMn$ when $OSTMnCMP = 0000\ 0000_H$, counter-start interrupts are enabled ($OSTMnCTL.OSTMnMD0 = 1$) and the timer is in timer output toggling mode, where $OSTMnTTOUT$ is toggled ($OSTMnTOE.OSTMnTOE = 1$).

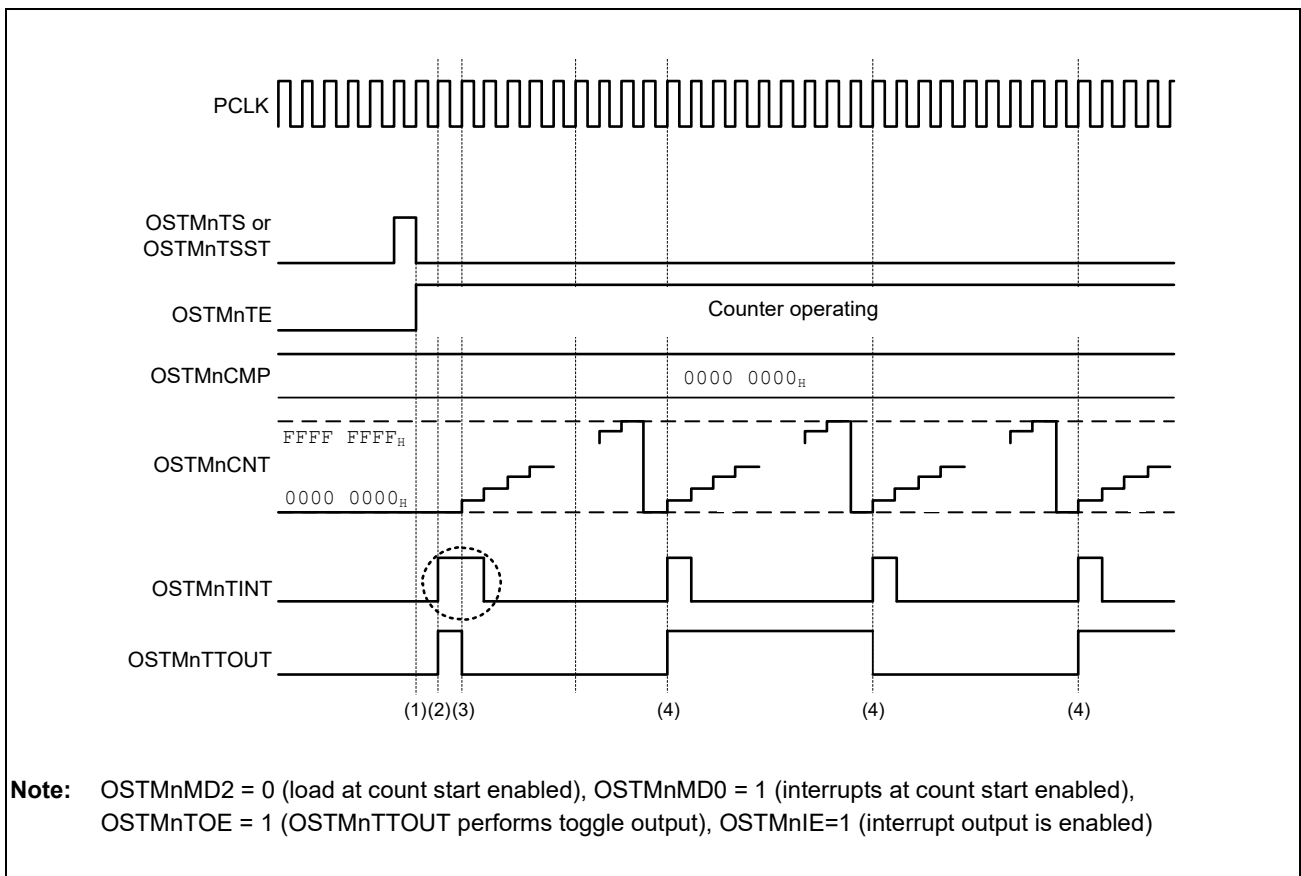


Figure 29.11 Timing Diagram when $OSTMnCMP = 0000\ 0000_H$ in Free-Run Compare Mode

The timing diagram above shows the following operations.

- (1) Once the counter starts, it counts up from $0000\ 0000_H$ to $FFFF\ FFFF_H$.
- (2) An $OSTMnTINT$ interrupt request is generated when counting starts and the $OSTMnTTOUT$ signal is toggled.
- (3) If the current counter value matches $OSTMnCMP$, the interrupt request $OSTMnTINT$ is generated. If $OSTMnCMP = 0000\ 0000_H$ as shown above, $OSTMnTINT$ is generated over two clock cycles and the $OSTMnTTOUT$ signal is toggled.
- (4) For every $(FFFF\ FFFF_H + 1)$ clock cycles, the $OSTMnTINT$ interrupt request signal is asserted and the $OSTMnTTOUT$ signal is toggled.

When an interrupt is prohibited when the counter starts, 1 clock cycle of the count clock is not generated at the start timing of the counter.

29.4.3.3 Forced Restart Operation in Free-Run Compare Mode

In the free-run compare mode, forced restart operation is not performed even when “1” is written to OSTMnTS during the count operation (OSTMnTS = 1). The OSTMnCNT counter continues counting at this time.

Forced restart operation is performed regardless of the value of OSTMnCTL.OSTMnMD2.

Section 30 ATU-V

30.1 Features of ATU-V

30.1.1 Units and Channels

This microcontroller has the following number of ATU-V units.

Table 30.1 Number of Units (1/2)

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Units	1	1
Unit Name	ATU-Vn (n = 0)	ATU-Vn (n = 0)

Table 30.1 Number of Units (2/2)

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Units	1	1
Unit Name	ATU-Vn (n = 0)	ATU-Vn (n = 0)

Table 30.2 Index

Index	Description
n	This section identifies each ATU-V channel by "n" (n = 0).

30.1.2 Register Base Address

ATU-V register addresses are represented by an offset from the base address.

The following table shows the base address of each ATU-V.

Table 30.3 Register Base Address

Base Address Name	Base Address	Bus Group
<ATU5CTRL_base>	FFE6 0000 _H	Peripheral Group 2H
<ATU5A_base>	FFE6 0200 _H	Peripheral Group 2H
<ATU5B_base>	FFE6 0400 _H	Peripheral Group 2H
<ATU5C_base>	FFE6 0800 _H	Peripheral Group 2H
<ATU5D_base>	FFE6 2000 _H	Peripheral Group 2H
<ATU5E_base>	FFE6 4000 _H	Peripheral Group 2H
<ATU5F_base>	FFE6 5000 _H	Peripheral Group 2H
<ATU5G_base>	FFE6 5C00 _H	Peripheral Group 2H
<ATU5TRG_base>	FFE6 8000 _H	Peripheral Group 2H

30.1.3 Clock Supply

The ATU-V clock supply is shown in the following table.

Table 30.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
ATU-Vn	ATU-V counter clock source (PCLK)	CLK_LSB

30.1.4 Interrupts Requests

ATU-V interrupt requests are listed in the following table.

Table 30.5 Interrupt Requests (1/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUASLIA0	INTATUASLIA0	ICRA0 input capture interrupt ICRA1 input capture interrupt ICRA2 input capture interrupt ICRA3 input capture interrupt ICRA4 input capture interrupt ICRA5 input capture interrupt ICRA6 input capture interrupt ICRA7 input capture interrupt	10	—	—	Selected by ATU_INTSEL_A0[2:0] bits of ATUINTSELA register.
INTATUASLIA1	INTATUASLIA1	ICRA0 input capture interrupt ICRA1 input capture interrupt ICRA2 input capture interrupt ICRA3 input capture interrupt ICRA4 input capture interrupt ICRA5 input capture interrupt ICRA6 input capture interrupt ICRA7 input capture interrupt	11	—	—	Selected by ATU_INTSEL_A1[2:0] bits of ATUINTSELA register.
INTATUASLIA2	INTATUASLIA2	ICRA0 input capture interrupt ICRA1 input capture interrupt ICRA2 input capture interrupt ICRA3 input capture interrupt ICRA4 input capture interrupt ICRA5 input capture interrupt ICRA6 input capture interrupt ICRA7 input capture interrupt	12	—	—	Selected by ATU_INTSEL_A2[2:0] bits of ATUINTSELA register.
INTATUASLIA3	INTATUASLIA3	ICRA0 input capture interrupt ICRA1 input capture interrupt ICRA2 input capture interrupt ICRA3 input capture interrupt ICRA4 input capture interrupt ICRA5 input capture interrupt ICRA6 input capture interrupt ICRA7 input capture interrupt	13	—	—	Selected by ATU_INTSEL_A3[2:0] bits of ATUINTSELA register.
INTATUAICIA0	INTATUAICIA0	ICRA0 input capture interrupt	78	group1-38, group1-166	group2-38, group2-102	—
INTATUAICIA1	INTATUAICIA1	ICRA1 input capture interrupt	79	group1-39, group1-167	group2-39, group2-103	—
INTATUAICIA2	INTATUAICIA2	ICRA2 input capture interrupt	80	group1-40, group1-168	group2-40, group2-104	—
INTATUAICIA3	INTATUAICIA3	ICRA3 input capture interrupt	81	group1-41, group1-169	group2-41, group2-105	—
INTATUAICIA4	INTATUAICIA4	ICRA4 input capture interrupt	82	group1-42, group1-170	group2-42, group2-106	—
INTATUAICIA5	INTATUAICIA5	ICRA5 input capture interrupt	83	group1-43, group1-171	group2-43, group2-107	—
INTATUAICIA6	INTATUAICIA6	ICRA6 input capture interrupt	84	group1-44, group1-172	group2-44, group2-108	—
INTATUAICIA7	INTATUAICIA7	ICRA7 input capture interrupt	85	group1-45, group1-173	group2-45, group2-109	—

Table 30.5 Interrupt Requests (2/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUACDFOVF	INTATUACDFOVF	ATU all overflow (TimerA/C/D/F overflow multiplexed)	86	—	—	—
INTATUBCMIB0	INTATUBCMIB0	OCRB0 compare match interrupt	87	—	—	—
INTATUBCMIB1	INTATUBCMIB1	OCRB1 compare match interrupt	88	—	—	—
INTATUBCMIB6	INTATUBCMIB6	OCRB6 compare match interrupt	89	—	—	—
INTATUBCMIB10	INTATUBCMIB10	OCRB10 compare match interrupt	90	—	—	—
INTATUBCMIB11	INTATUBCMIB11	OCRB11 compare match interrupt	91	—	—	—
INTATUBCMIB12	INTATUBCMIB12	OCRB12 compare match interrupt	92	—	—	—
INTATUBCMIB6M	INTATUBCMIB6M	Comparison between TCNTB6M and ICRB6 condition match interrupt	93	—	—	—
INTATUBCMIB66M	INTATUBCMIB66M	AND/OR condition of CMFB6 and CMFB6M condition match interrupt	94	—	—	—
INTATUBICIB0	INTATUBICIB0	ICRB0 input capture interrupt	95	—	—	—
INTATUCLIC00	INTATUCLIC00	GRC0_0 input capture/compare match interrupt	96	—	—	All interrupts combined by OR
		OCRC0_0 input capture/compare match interrupt				
INTATUCLIC01	INTATUCLIC01	GRC0_1 input capture/compare match interrupt	97	—	—	All interrupts combined by OR
		OCRC0_1 input capture/compare match interrupt				
INTATUCLIC02	INTATUCLIC02	GRC0_2 input capture/compare match interrupt	98	—	—	All interrupts combined by OR
		OCRC0_2 input capture/compare match interrupt				
INTATUCLIC03	INTATUCLIC03	GRC0_3 input capture/compare match interrupt	99	—	—	All interrupts combined by OR
		OCRC0_3 input capture/compare match interrupt				
INTATUCLIC10	INTATUCLIC10	GRC1_0 input capture/compare match interrupt	100	—	—	All interrupts combined by OR
		OCRC1_0 input capture/compare match interrupt				
INTATUCLIC11	INTATUCLIC11	GRC1_1 input capture/compare match interrupt	101	—	—	All interrupts combined by OR
		OCRC1_1 input capture/compare match interrupt				
INTATUCLIC12	INTATUCLIC12	GRC1_2 input capture/compare match interrupt	102	—	—	All interrupts combined by OR
		OCRC1_2 input capture/compare match interrupt				
INTATUCLIC13	INTATUCLIC13	GRC1_3 input capture/compare match interrupt	103	—	—	All interrupts combined by OR
		OCRC1_3 input capture/compare match interrupt				
INTATUCLIC20	INTATUCLIC20	GRC2_0 input capture/compare match interrupt	104	—	—	All interrupts combined by OR
		OCRC2_0 input capture/compare match interrupt				
INTATUCLIC21	INTATUCLIC21	GRC2_1 input capture/compare match interrupt	105	—	—	All interrupts combined by OR
		OCRC2_1 input capture/compare match interrupt				
INTATUCLIC22	INTATUCLIC22	GRC2_2 input capture/compare match interrupt	106	—	—	All interrupts combined by OR
		OCRC2_2 input capture/compare match interrupt				

Table 30.5 Interrupt Requests (3/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUCSLIC23	INTATUCSLIC23	GRC2_3 input capture/compare match interrupt	107	—	—	All interrupts combined by OR
		OCRC2_3 input capture/compare match interrupt				
INTATUCSLIC30	INTATUCSLIC30	GRC3_0 input capture/compare match interrupt	108	—	—	All interrupts combined by OR
		OCRC3_0 input capture/compare match interrupt				
INTATUCSLIC31	INTATUCSLIC31	GRC3_1 input capture/compare match interrupt	109	—	—	All interrupts combined by OR
		OCRC3_1 input capture/compare match interrupt				
INTATUCSLIC32	INTATUCSLIC32	GRC3_2 input capture/compare match interrupt	110	—	—	All interrupts combined by OR
		OCRC3_2 input capture/compare match interrupt				
INTATUCSLIC33	INTATUCSLIC33	GRC3_3 input capture/compare match interrupt	111	—	—	All interrupts combined by OR
		OCRC3_3 input capture/compare match interrupt				
INTATUCSLIC40	INTATUCSLIC40	GRC4_0 input capture/compare match interrupt	112	—	—	All interrupts combined by OR
		OCRC4_0 input capture/compare match interrupt				
INTATUCSLIC41	INTATUCSLIC41	GRC4_1 input capture/compare match interrupt	113	—	—	All interrupts combined by OR
		OCRC4_1 input capture/compare match interrupt				
INTATUCSLIC42	INTATUCSLIC42	GRC4_2 input capture/compare match interrupt	114	—	—	All interrupts combined by OR
		OCRC4_2 input capture/compare match interrupt				
INTATUCSLIC43	INTATUCSLIC43	GRC4_3 input capture/compare match interrupt	115	—	—	All interrupts combined by OR
		OCRC4_3 input capture/compare match interrupt				
INTATUCSLIC50	INTATUCSLIC50	GRC5_0 input capture/compare match interrupt	116	—	—	All interrupts combined by OR
		OCRC5_0 input capture/compare match interrupt				
INTATUCSLIC51	INTATUCSLIC51	GRC5_1 input capture/compare match interrupt	117	—	—	All interrupts combined by OR
		OCRC5_1 input capture/compare match interrupt				
INTATUCSLIC52	INTATUCSLIC52	GRC5_2 input capture/compare match interrupt	118	—	—	All interrupts combined by OR
		OCRC5_2 input capture/compare match interrupt				
INTATUCSLIC53	INTATUCSLIC53	GRC5_3 input capture/compare match interrupt	119	—	—	All interrupts combined by OR
		OCRC5_3 input capture/compare match interrupt				
INTATUCSLIC60	INTATUCSLIC60	GRC6_0 input capture/compare match interrupt	120	—	—	All interrupts combined by OR
		OCRC6_0 input capture/compare match interrupt				
INTATUCSLIC61	INTATUCSLIC61	GRC6_1 input capture/compare match interrupt	121	—	—	All interrupts combined by OR
		OCRC6_1 input capture/compare match interrupt				

Table 30.5 Interrupt Requests (4/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUCLIC62	INTATUCLIC62	GRC6_2 input capture/compare match interrupt	122	—	—	All interrupts combined by OR
		OCRC6_2 input capture/compare match interrupt				
INTATUCLIC63	INTATUCLIC63	GRC6_3 input capture/compare match interrupt	123	—	—	All interrupts combined by OR
		OCRC6_3 input capture/compare match interrupt				
INTATUCLIC70	INTATUCLIC70	GRC7_0 input capture/compare match interrupt	124	—	—	All interrupts combined by OR
		OCRC7_0 input capture/compare match interrupt				
INTATUCLIC71	INTATUCLIC71	GRC7_1 input capture/compare match interrupt	125	—	—	All interrupts combined by OR
		OCRC7_1 input capture/compare match interrupt				
INTATUCLIC72	INTATUCLIC72	GRC7_2 input capture/compare match interrupt	126	—	—	All interrupts combined by OR
		OCRC7_2 input capture/compare match interrupt				
INTATUCLIC73	INTATUCLIC73	GRC7_3 input capture/compare match interrupt	127	—	—	All interrupts combined by OR
		OCRC7_3 input capture/compare match interrupt				
INTATUCLIC80	INTATUCLIC80	GRC8_0 input capture/compare match interrupt	128	—	—	All interrupts combined by OR
		OCRC8_0 input capture/compare match interrupt				
INTATUCLIC81	INTATUCLIC81	GRC8_1 input capture/compare match interrupt	129	—	—	All interrupts combined by OR
		OCRC8_1 input capture/compare match interrupt				
INTATUCLIC82	INTATUCLIC82	GRC8_2 input capture/compare match interrupt	130	—	—	All interrupts combined by OR
		OCRC8_2 input capture/compare match interrupt				
INTATUCLIC83	INTATUCLIC83	GRC8_3 input capture/compare match interrupt	131	—	—	All interrupts combined by OR
		OCRC8_3 input capture/compare match interrupt				
INTATUCLIC90	INTATUCLIC90	GRC9_0 input capture/compare match interrupt	132	—	—	All interrupts combined by OR
		OCRC9_0 input capture/compare match interrupt				
INTATUCLIC91	INTATUCLIC91	GRC9_1 input capture/compare match interrupt	133	—	—	All interrupts combined by OR
		OCRC9_1 input capture/compare match interrupt				
INTATUCLIC92	INTATUCLIC92	GRC9_2 input capture/compare match interrupt	134	—	—	All interrupts combined by OR
		OCRC9_2 input capture/compare match interrupt				
INTATUCLIC93	INTATUCLIC93	GRC9_3 input capture/compare match interrupt	135	—	—	All interrupts combined by OR
		OCRC9_3 input capture/compare match interrupt				
INTATUCLIC100	INTATUCLIC100	GRC10_0 input capture/compare match interrupt	136	—	—	All interrupts combined by OR
		OCRC10_0 input capture/compare match interrupt				

Table 30.5 Interrupt Requests (5/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUCSLIC101	INTATUCSLIC101	GRC10_1 input capture/compare match interrupt	137	—	—	All interrupts combined by OR
		OCRC10_1 input capture/compare match interrupt				
INTATUCSLIC102	INTATUCSLIC102	GRC10_2 input capture/compare match interrupt	138	—	—	All interrupts combined by OR
		OCRC10_2 input capture/compare match interrupt				
INTATUCSLIC103	INTATUCSLIC103	GRC10_3 input capture/compare match interrupt	139	—	—	All interrupts combined by OR
		OCRC10_3 input capture/compare match interrupt				
INTATUDSLID00	INTATUDSLID00	OCR1D0_0 compare match interrupt	140	—	—	Selected by ATU_INTSEL_D00[2:0] bits of ATUINTELD0 register.
		OCR2D0_0 compare match interrupt				
		1shot pulse ON D0_0 occurrence interrupt				
		1shot pulse OFF D0_0 occurrence interrupt				
		UDID0_0 down-counter underflow interrupt				
INTATUDSLID01	INTATUDSLID01	OCR1D0_1 compare match interrupt	141	—	—	Selected by ATU_INTSEL_D01[2:0] bits of ATUINTELD0 register.
		OCR2D0_1 compare match interrupt				
		1shot pulse ON D0_1 occurrence interrupt				
		1shot pulse OFF D0_1 occurrence interrupt				
		UDID0_1 down-counter underflow interrupt				
INTATUDSLID02	INTATUDSLID02	OCR1D0_2 compare match interrupt	142	—	—	Selected by ATU_INTSEL_D02[2:0] bits of ATUINTELD0 register.
		OCR2D0_2 compare match interrupt				
		1shot pulse ON D0_2 occurrence interrupt				
		1shot pulse OFF D0_2 occurrence interrupt				
		UDID0_2 down-counter underflow interrupt				
INTATUDSLID03	INTATUDSLID03	OCR1D0_3 compare match interrupt	143	—	—	Selected by ATU_INTSEL_D03[2:0] bits of ATUINTELD0 register.
		OCR2D0_3 compare match interrupt				
		1shot pulse ON D0_3 occurrence interrupt				
		1shot pulse OFF D0_3 occurrence interrupt				
		UDID0_3 down-counter underflow interrupt				
INTATUDSLID10	INTATUDSLID10	OCR1D1_0 compare match interrupt	144	—	—	Selected by ATU_INTSEL_D10[2:0] bits of ATUINTELD0 register.
		OCR2D1_0 compare match interrupt				
		1shot pulse ON D1_0 occurrence interrupt				
		1shot pulse OFF D1_0 occurrence interrupt				
		UDID1_0 down-counter underflow interrupt				

Table 30.5 Interrupt Requests (6/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUDSLID11	INTATUDSLID11	OCR1D1_1 compare match interrupt	145	—	—	Selected by ATU_INTSEL_D11[2:0] bits of ATUINTELD0 register.
		OCR2D1_1 compare match interrupt				
		1shot pulse ON D1_1 occurrence interrupt				
		1shot pulse OFF D1_1 occurrence interrupt				
		UDID1_1 down-counter underflow interrupt				
INTATUDSLID12	INTATUDSLID12	OCR1D1_2 compare match interrupt	146	—	—	Selected by ATU_INTSEL_D12[2:0] bits of ATUINTELD0 register.
		OCR2D1_2 compare match interrupt				
		1shot pulse ON D1_2 occurrence interrupt				
		1shot pulse OFF D1_2 occurrence interrupt				
		UDID1_2 down-counter underflow interrupt				
INTATUDSLID13	INTATUDSLID13	OCR1D1_3 compare match interrupt	147	—	—	Selected by ATU_INTSEL_D13[2:0] bits of ATUINTELD0 register.
		OCR2D1_3 compare match interrupt				
		1shot pulse ON D1_3 occurrence interrupt				
		1shot pulse OFF D1_3 occurrence interrupt				
		UDID1_3 down-counter underflow interrupt				
INTATUDSLID20	INTATUDSLID20	OCR1D2_0 compare match interrupt	148	—	—	Selected by ATU_INTSEL_D20[2:0] bits of ATUINTELD1 register.
		OCR2D2_0 compare match interrupt				
		1shot pulse ON D2_0 occurrence interrupt				
		1shot pulse OFF D2_0 occurrence interrupt				
		UDID2_0 down-counter underflow interrupt				
INTATUDSLID21	INTATUDSLID21	OCR1D2_1 compare match interrupt	149	—	—	Selected by ATU_INTSEL_D21[2:0] bits of ATUINTELD1 register.
		OCR2D2_1 compare match interrupt				
		1shot pulse ON D2_1 occurrence interrupt				
		1shot pulse OFF D2_1 occurrence interrupt				
		UDID2_1 down-counter underflow interrupt				
INTATUDSLID22	INTATUDSLID22	OCR1D2_2 compare match interrupt	150	—	—	Selected by ATU_INTSEL_D22[2:0] bits of ATUINTELD1 register.
		OCR2D2_2 compare match interrupt				
		1shot pulse ON D2_2 occurrence interrupt				
		1shot pulse OFF D2_2 occurrence interrupt				
		UDID2_2 down-counter underflow interrupt				
INTATUDSLID23	INTATUDSLID23	OCR1D2_3 compare match interrupt	151	—	—	Selected by ATU_INTSEL_D23[2:0] bits of ATUINTELD1 register.
		OCR2D2_3 compare match interrupt				
		1shot pulse ON D2_3 occurrence interrupt				
		1shot pulse OFF D2_3 occurrence interrupt				
		UDID2_3 down-counter underflow interrupt				

Table 30.5 Interrupt Requests (7/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUDSLID30	INTATUDSLID30	OCR1D3_0 compare match interrupt	152	—	—	Selected by ATU_INTSEL_D30[2:0] bits of ATUINTSELD1 register.
		OCR2D3_0 compare match interrupt				
		1shot pulse ON D3_0 occurrence interrupt				
		1shot pulse OFF D3_0 occurrence interrupt				
		UDID3_0 down-counter underflow interrupt				
INTATUDSLID31	INTATUDSLID31	OCR1D3_1 compare match interrupt	153	—	—	Selected by ATU_INTSEL_D31[2:0] bits of ATUINTSELD1 register.
		OCR2D3_1 compare match interrupt				
		1shot pulse ON D3_1 occurrence interrupt				
		1shot pulse OFF D3_1 occurrence interrupt				
		UDID3_1 down-counter underflow interrupt				
INTATUDSLID32	INTATUDSLID32	OCR1D3_2 compare match interrupt	154	—	—	Selected by ATU_INTSEL_D32[2:0] bits of ATUINTSELD1 register.
		OCR2D3_2 compare match interrupt				
		1shot pulse ON D3_2 occurrence interrupt				
		1shot pulse OFF D3_2 occurrence interrupt				
		UDID3_2 down-counter underflow interrupt				
INTATUDSLID33	INTATUDSLID33	OCR1D3_3 compare match interrupt	155	—	—	Selected by ATU_INTSEL_D33[2:0] bits of ATUINTSELD1 register.
		OCR2D3_3 compare match interrupt				
		1shot pulse ON D3_3 occurrence interrupt				
		1shot pulse OFF D3_3 occurrence interrupt				
		UDID3_3 down-counter underflow interrupt				
INTATUDSLID40	INTATUDSLID40	OCR1D4_0 compare match interrupt	156	—	—	Selected by ATU_INTSEL_D40[2:0] bits of ATUINTSELD2 register.
		OCR2D4_0 compare match interrupt				
		1shot pulse ON D4_0 occurrence interrupt				
		1shot pulse OFF D4_0 occurrence interrupt				
		UDID4_0 down-counter underflow interrupt				
INTATUDSLID41	INTATUDSLID41	OCR1D4_1 compare match interrupt	157	—	—	Selected by ATU_INTSEL_D41[2:0] bits of ATUINTSELD2 register.
		OCR2D4_1 compare match interrupt				
		1shot pulse ON D4_1 occurrence interrupt				
		1shot pulse OFF D4_1 occurrence interrupt				
		UDID4_1 down-counter underflow interrupt				
INTATUDSLID42	INTATUDSLID42	OCR1D4_2 compare match interrupt	158	—	—	Selected by ATU_INTSEL_D42[2:0] bits of ATUINTSELD2 register.
		OCR2D4_2 compare match interrupt				
		1shot pulse ON D4_2 occurrence interrupt				
		1shot pulse OFF D4_2 occurrence interrupt				
		UDID4_2 down-counter underflow interrupt				

Table 30.5 Interrupt Requests (8/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUDSLID43	INTATUDSLID43	OCR1D4_3 compare match interrupt	159	—	—	Selected by ATU_INTSEL_D43[2:0] bits of ATUINTSELD2 register.
		OCR2D4_3 compare match interrupt				
		1shot pulse ON D4_3 occurrence interrupt				
		1shot pulse OFF D4_3 occurrence interrupt				
		UDID4_3 down-counter underflow interrupt				
INTATUDSLID50	INTATUDSLID50	OCR1D5_0 compare match interrupt	160	—	—	Selected by ATU_INTSEL_D50[2:0] bits of ATUINTSELD2 register.
		OCR2D5_0 compare match interrupt				
		1shot pulse ON D5_0 occurrence interrupt				
		1shot pulse OFF D5_0 occurrence interrupt				
		UDID5_0 down-counter underflow interrupt				
INTATUDSLID51	INTATUDSLID51	OCR1D5_1 compare match interrupt	161	—	—	Selected by ATU_INTSEL_D51[2:0] bits of ATUINTSELD2 register.
		OCR2D5_1 compare match interrupt				
		1shot pulse ON D5_1 occurrence interrupt				
		1shot pulse OFF D5_1 occurrence interrupt				
		UDID5_1 down-counter underflow interrupt				
INTATUDSLID52	INTATUDSLID52	OCR1D5_2 compare match interrupt	162	—	—	Selected by ATU_INTSEL_D52[2:0] bits of ATUINTSELD2 register.
		OCR2D5_2 compare match interrupt				
		1shot pulse ON D5_2 occurrence interrupt				
		1shot pulse OFF D5_2 occurrence interrupt				
		UDID5_2 down-counter underflow interrupt				
INTATUDSLID53	INTATUDSLID53	OCR1D5_3 compare match interrupt	163	—	—	Selected by ATU_INTSEL_D53[2:0] bits of ATUINTSELD2 register.
		OCR2D5_3 compare match interrupt				
		1shot pulse ON D5_3 occurrence interrupt				
		1shot pulse OFF D5_3 occurrence interrupt				
		UDID5_3 down-counter underflow interrupt				
INTATUDSLID60	INTATUDSLID60	OCR1D6_0 compare match interrupt	164	—	—	Selected by ATU_INTSEL_D60[2:0] bits of ATUINTSELD3 register.
		OCR2D6_0 compare match interrupt				
		1shot pulse ON D6_0 occurrence interrupt				
		1shot pulse OFF D6_0 occurrence interrupt				
		UDID6_0 down-counter underflow interrupt				
INTATUDSLID61	INTATUDSLID61	OCR1D6_1 compare match interrupt	165	—	—	Selected by ATU_INTSEL_D61[2:0] bits of ATUINTSELD3 register.
		OCR2D6_1 compare match interrupt				
		1shot pulse ON D6_1 occurrence interrupt				
		1shot pulse OFF D6_1 occurrence interrupt				
		UDID6_1 down-counter underflow interrupt				

Table 30.5 Interrupt Requests (9/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUDSLID62	INTATUDSLID62	OCR1D6_2 compare match interrupt	166	—	—	Selected by ATU_INTSEL_D62[2:0] bits of ATUINTSELD3 register.
		OCR2D6_2 compare match interrupt				
		1shot pulse ON D6_2 occurrence interrupt				
		1shot pulse OFF D6_2 occurrence interrupt				
		UDID6_2 down-counter underflow interrupt				
INTATUDSLID63	INTATUDSLID63	OCR1D6_3 compare match interrupt	167	—	—	Selected by ATU_INTSEL_D63[2:0] bits of ATUINTSELD3 register.
		OCR2D6_3 compare match interrupt				
		1shot pulse ON D6_3 occurrence interrupt				
		1shot pulse OFF D6_3 occurrence interrupt				
		UDID6_3 down-counter underflow interrupt				
INTATUDSLID70	INTATUDSLID70	OCR1D7_0 compare match interrupt	168	—	—	Selected by ATU_INTSEL_D70[2:0] bits of ATUINTSELD3 register.
		OCR2D7_0 compare match interrupt				
		1shot pulse ON D7_0 occurrence interrupt				
		1shot pulse OFF D7_0 occurrence interrupt				
		UDID7_0 down-counter underflow interrupt				
INTATUDSLID71	INTATUDSLID71	OCR1D7_1 compare match interrupt	169	—	—	Selected by ATU_INTSEL_D71[2:0] bits of ATUINTSELD3 register.
		OCR2D7_1 compare match interrupt				
		1shot pulse ON D7_1 occurrence interrupt				
		1shot pulse OFF D7_1 occurrence interrupt				
		UDID7_1 down-counter underflow interrupt				
INTATUDSLID72	INTATUDSLID72	OCR1D7_2 compare match interrupt	170	—	—	Selected by ATU_INTSEL_D72[2:0] bits of ATUINTSELD3 register.
		OCR2D7_2 compare match interrupt				
		1shot pulse ON D7_2 occurrence interrupt				
		1shot pulse OFF D7_2 occurrence interrupt				
		UDID7_2 down-counter underflow interrupt				
INTATUDSLID73	INTATUDSLID73	OCR1D7_3 compare match interrupt	171	—	—	Selected by ATU_INTSEL_D73[2:0] bits of ATUINTSELD3 register.
		OCR2D7_3 compare match interrupt				
		1shot pulse ON D7_3 occurrence interrupt				
		1shot pulse OFF D7_3 occurrence interrupt				
		UDID7_3 down-counter underflow interrupt				
INTATUDSLID80	INTATUDSLID80	OCR1D8_0 compare match interrupt	172	—	—	Selected by ATU_INTSEL_D80[2:0] bits of ATUINTSELD4 register.
		OCR2D8_0 compare match interrupt				
		1shot pulse ON D8_0 occurrence interrupt				
		1shot pulse OFF D8_0 occurrence interrupt				
		UDID8_0 down-counter underflow interrupt				

Table 30.5 Interrupt Requests (10/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUDSLID81	INTATUDSLID81	OCR1D8_1 compare match interrupt	173	—	—	Selected by ATU_INTSEL_D81[2:0] bits of ATUINTSELD4 register.
		OCR2D8_1 compare match interrupt				
		1shot pulse ON D8_1 occurrence interrupt				
		1shot pulse OFF D8_1 occurrence interrupt				
		UDID8_1 down-counter underflow interrupt				
INTATUDSLID82	INTATUDSLID82	OCR1D8_2 compare match interrupt	174	—	—	Selected by ATU_INTSEL_D82[2:0] bits of ATUINTSELD4 register.
		OCR2D8_2 compare match interrupt				
		1shot pulse ON D8_2 occurrence interrupt				
		1shot pulse OFF D8_2 occurrence interrupt				
		UDID8_2 down-counter underflow interrupt				
INTATUDSLID83	INTATUDSLID83	OCR1D8_3 compare match interrupt	175	—	—	Selected by ATU_INTSEL_D83[2:0] bits of ATUINTSELD4 register.
		OCR2D8_3 compare match interrupt				
		1shot pulse ON D8_3 occurrence interrupt				
		1shot pulse OFF D8_3 occurrence interrupt				
		UDID8_3 down-counter underflow interrupt				
INTATUECMIE00	INTATUECMIE00	CYLRE0_0 cycle compare match interrupt / DTRE0_0 duty compare match interrupt	176	—	—	—
INTATUECMIE01	INTATUECMIE01	CYLRE0_1 cycle compare match interrupt / DTRE0_1 duty compare match interrupt	177	—	—	—
INTATUECMIE02	INTATUECMIE02	CYLRE0_2 cycle compare match interrupt / DTRE0_2 duty compare match interrupt	178	—	—	—
INTATUECMIE03	INTATUECMIE03	CYLRE0_3 cycle compare match interrupt / DTRE0_3 duty compare match interrupt	179	—	—	—
INTATUECMIE10	INTATUECMIE10	CYLRE1_0 cycle compare match interrupt / DTRE1_0 duty compare match interrupt	180	—	—	—
INTATUECMIE11	INTATUECMIE11	CYLRE1_1 cycle compare match interrupt / DTRE1_1 duty compare match interrupt	181	—	—	—
INTATUECMIE12	INTATUECMIE12	CYLRE1_2 cycle compare match interrupt / DTRE1_2 duty compare match interrupt	182	—	—	—
INTATUECMIE13	INTATUECMIE13	CYLRE1_3 cycle compare match interrupt / DTRE1_3 duty compare match interrupt	183	—	—	—
INTATUECMIE20	INTATUECMIE20	CYLRE2_0 cycle compare match interrupt / DTRE2_0 duty compare match interrupt	184	—	—	—
INTATUECMIE21	INTATUECMIE21	CYLRE2_1 cycle compare match interrupt / DTRE2_1 duty compare match interrupt	185	—	—	—
INTATUECMIE22	INTATUECMIE22	CYLRE2_2 cycle compare match interrupt / DTRE2_2 duty compare match interrupt	186	—	—	—
INTATUECMIE23	INTATUECMIE23	CYLRE2_3 cycle compare match interrupt / DTRE2_3 duty compare match interrupt	187	—	—	—
INTATUECMIE30	INTATUECMIE30	CYLRE3_0 cycle compare match interrupt / DTRE3_0 duty compare match interrupt	188	—	—	—

Table 30.5 Interrupt Requests (11/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUECMIE31	INTATUECMIE31	CYLRE3_1 cycle compare match interrupt / DTRE3_1 duty compare match interrupt	189	—	—	—
INTATUECMIE32	INTATUECMIE32	CYLRE3_2 cycle compare match interrupt / DTRE3_2 duty compare match interrupt	190	—	—	—
INTATUECMIE33	INTATUECMIE33	CYLRE3_3 cycle compare match interrupt / DTRE3_3 duty compare match interrupt	191	—	—	—
INTATUECMIE40	INTATUECMIE40	CYLRE4_0 cycle compare match interrupt / DTRE4_0 duty compare match interrupt	192	—	—	—
INTATUECMIE41	INTATUECMIE41	CYLRE4_1 cycle compare match interrupt / DTRE4_1 duty compare match interrupt	193	—	—	—
INTATUECMIE42	INTATUECMIE42	CYLRE4_2 cycle compare match interrupt / DTRE4_2 duty compare match interrupt	194	—	—	—
INTATUECMIE43	INTATUECMIE43	CYLRE4_3 cycle compare match interrupt / DTRE4_3 duty compare match interrupt	195	—	—	—
INTATUECMIE50	INTATUECMIE50	CYLRE5_0 cycle compare match interrupt / DTRE5_0 duty compare match interrupt	196	—	—	—
INTATUECMIE51	INTATUECMIE51	CYLRE5_1 cycle compare match interrupt / DTRE5_1 duty compare match interrupt	197	—	—	—
INTATUECMIE52	INTATUECMIE52	CYLRE5_2 cycle compare match interrupt / DTRE5_2 duty compare match interrupt	198	—	—	—
INTATUECMIE53	INTATUECMIE53	CYLRE5_3 cycle compare match interrupt / DTRE5_3 duty compare match interrupt	199	—	—	—
INTATUECMIE60	INTATUECMIE60	CYLRE6_0 cycle compare match interrupt / DTRE6_0 duty compare match interrupt	200	—	—	—
INTATUECMIE61	INTATUECMIE61	CYLRE6_1 cycle compare match interrupt / DTRE6_1 duty compare match interrupt	201	—	—	—
INTATUECMIE62	INTATUECMIE62	CYLRE6_2 cycle compare match interrupt / DTRE6_2 duty compare match interrupt	202	—	—	—
INTATUECMIE63	INTATUECMIE63	CYLRE6_3 cycle compare match interrupt / DTRE6_3 duty compare match interrupt	203	—	—	—
INTATUECMIE70	INTATUECMIE70	CYLRE7_0 cycle compare match interrupt / DTRE7_0 duty compare match interrupt	204	—	—	—
INTATUECMIE71	INTATUECMIE71	CYLRE7_1 cycle compare match interrupt / DTRE7_1 duty compare match interrupt	205	—	—	—
INTATUECMIE72	INTATUECMIE72	CYLRE7_2 cycle compare match interrupt / DTRE7_2 duty compare match interrupt	206	—	—	—
INTATUECMIE73	INTATUECMIE73	CYLRE7_3 cycle compare match interrupt / DTRE7_3 duty compare match interrupt	207	—	—	—
INTATUECMIE80	INTATUECMIE80	CYLRE8_0 cycle compare match interrupt / DTRE8_0 duty compare match interrupt	208	—	—	—
INTATUECMIE81	INTATUECMIE81	CYLRE8_1 cycle compare match interrupt / DTRE8_1 duty compare match interrupt	209	—	—	—
INTATUECMIE82	INTATUECMIE82	CYLRE8_2 cycle compare match interrupt / DTRE8_2 duty compare match interrupt	210	—	—	—

Table 30.5 Interrupt Requests (12/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUECMIE83	INTATUECMIE83	CYLRE8_3 cycle compare match interrupt / DTRE8_3 duty compare match interrupt	211	—	—	—
INTATUFICIF0	INTATUFICIF0	Timer F0 input capture interrupt	212	group1-16, group1-144	group2- 16, group2-80	—
INTATUFICIF1	INTATUFICIF1	Timer F1 input capture interrupt	213	group1-17, group1-145	group2- 17, group2-81	—
INTATUFICIF2	INTATUFICIF2	Timer F2 input capture interrupt	214	group1-18, group1-146	group2- 18, group2-82	—
INTATUFICIF3	INTATUFICIF3	Timer F3 input capture interrupt	215	group1-19, group1-147	group2- 19, group2-83	—
INTATUFICIF4	INTATUFICIF4	Timer F4 input capture interrupt	216	group1-20, group1-148	group2- 20, group2-84	—
INTATUFICIF5	INTATUFICIF5	Timer F5 input capture interrupt	217	group1-21, group1-149	group2- 21, group2-85	—
INTATUFICIF6	INTATUFICIF6	Timer F6 input capture interrupt	218	group1-22, group1-150	group2- 22, group2-86	—
INTATUFICIF7	INTATUFICIF7	Timer F7 input capture interrupt	219	group1-23, group1-151	group2- 23, group2-87	—
INTATUFICIF8	INTATUFICIF8	Timer F8 input capture interrupt	220	group1-24, group1-152	group2- 24, group2-88	—
INTATUFICIF9	INTATUFICIF9	Timer F9 input capture interrupt	221	group1-25, group1-153	group2- 25, group2-89	—
INTATUFICIF10	INTATUFICIF10	Timer F10 input capture interrupt	222	group1-26, group1-154	group2- 26, group2-90	—
INTATUFICIF11	INTATUFICIF11	Timer F11 input capture interrupt	223	group1-27, group1-155	group2- 27, group2-91	—
INTATUFICIF12	INTATUFICIF12	Timer F12 input capture interrupt	224	group1-28, group1-156	group2- 28, group2-92	—
INTATUFICIF13	INTATUFICIF13	Timer F13 input capture interrupt	225	group1-29, group1-157	group2- 29, group2-93	—
INTATUFICIF14	INTATUFICIF14	Timer F14 input capture interrupt	226	group1-30, group1-158	group2- 30, group2-94	—
INTATUFICIF15	INTATUFICIF15	Timer F15 input capture interrupt	227	group1-31, group1-159	group2- 31, group2-95	—
INTATUGCMIG0	INTATUGCMIG0	OCR0 compare match interrupt	232	group1-50, group1-178	group2- 50, group2-114	—
INTATUGCMIG1	INTATUGCMIG1	OCR1 compare match interrupt	233	group1-51, group1-179	group2- 51, group2-115	—
INTATUGCMIG2	INTATUGCMIG2	OCR2 compare match interrupt	234	group1-52, group1-180	group2- 52, group2-116	—
INTATUGCMIG3	INTATUGCMIG3	OCR3 compare match interrupt	235	group1-53, group1-181	group2- 53, group2-117	—
INTATUGCMIG4	INTATUGCMIG4	OCR4 compare match interrupt	236	group1-54, group1-182	group2- 54, group2-118	—
INTATUGCMIG5	INTATUGCMIG5	OCR5 compare match interrupt	237	group1-55, group1-183	group2- 55, group2-119	—
INTATUGCMIG6	INTATUGCMIG6	OCR6 compare match interrupt	238	group1-56, group1-184	group2- 56, group2-120	—
INTATUGCMIG7	INTATUGCMIG7	OCR7 compare match interrupt	239	group1-57, group1-185	group2- 57, group2-121	—
INTATUGCMIG8	INTATUGCMIG8	OCR8 compare match interrupt	240	group1-58, group1-186	group2- 58, group2-122	—
INTATUGCMIG9	INTATUGCMIG9	OCR9 compare match interrupt	241	group1-59, group1-187	group2- 59, group2-123	—

Table 30.5 Interrupt Requests (13/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUBSLDB0	INTATUBSLDB0	OCRB0 compare match interrupt	—	group1-46, group1-174	group2- 46, group2-110	Selected by ATU_DMASEL_B0[3:0] bits of ATUDMASELB register.
		OCRB1 compare match interrupt				
		OCRB6 compare match interrupt				
		OCRB10 compare match interrupt				
		OCRB11 compare match interrupt				
		OCRB12 compare match interrupt				
		Comparison between TCNTB6M and ICRB6 condition match interrupt				
		AND/OR condition of CMFB6 and CMFB6M condition match interrupt				
		ICRB0 input capture interrupt				
INTATUBSLDB1	INTATUBSLDB1	OCRB0 compare match interrupt	—	group1-47, group1-175	group2- 47, group2-111	Selected by ATU_DMASEL_B1[3:0] bits of ATUDMASELB register.
		OCRB1 compare match interrupt				
		OCRB6 compare match interrupt				
		OCRB10 compare match interrupt				
		OCRB11 compare match interrupt				
		OCRB12 compare match interrupt				
		Comparison between TCNTB6M and ICRB6 condition match interrupt				
		AND/OR condition of CMFB6 and CMFB6M condition match interrupt				
		ICRB0 input capture interrupt				
INTATUBSLDB2	INTATUBSLDB2	OCRB0 compare match interrupt	—	group1-48, group1-176	group2- 48, group2-112	Selected by ATU_DMASEL_B2[3:0] bits of ATUDMASELB register.
		OCRB1 compare match interrupt				
		OCRB6 compare match interrupt				
		OCRB10 compare match interrupt				
		OCRB11 compare match interrupt				
		OCRB12 compare match interrupt				
		Comparison between TCNTB6M and ICRB6 condition match interrupt				
		AND/OR condition of CMFB6 and CMFB6M condition match interrupt				
		ICRB0 input capture interrupt				
INTATUBSLDB3	INTATUBSLDB3	OCRB0 compare match interrupt	—	group1-49, group1-177	group2- 49, group2-113	Selected by ATU_DMASEL_B3[3:0] bits of ATUDMASELB register.
		OCRB1 compare match interrupt				
		OCRB6 compare match interrupt				
		OCRB10 compare match interrupt				
		OCRB11 compare match interrupt				
		OCRB12 compare match interrupt				
		Comparison between TCNTB6M and ICRB6 condition match interrupt				
		AND/OR condition of CMFB6 and CMFB6M condition match interrupt				
		ICRB0 input capture interrupt				

Table 30.5 Interrupt Requests (14/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUCDSLDCD0	INTATUCDSLDCD0	GRC0_0 input capture/compare match interrupt	—	group1-64, group1-192	group3-0, group3-64	Selected by ATU_DMASEL_CD00[2:0] bits of ATUDMASELCD0 register.
		OCRC0_0 input capture/compare match interrupt				
		OCR1D8_0 compare match interrupt				
		OCR2D8_0 compare match interrupt				
		1shot pulse ON D8_0 occurrence interrupt				
		1shot pulse OFF D8_0 occurrence interrupt				
		UDID8_0 down-counter underflow interrupt				
INTATUCDSLDCD1	INTATUCDSLDCD1	GRC0_1 input capture/compare match interrupt	—	group1-65, group1-193	group3-1, group3-65	Selected by ATU_DMASEL_CD01[2:0] bits of ATUDMASELCD0 register.
		OCRC0_1 input capture/compare match interrupt				
		OCR1D8_1 compare match interrupt				
		OCR2D8_1 compare match interrupt				
		1shot pulse ON D8_1 occurrence interrupt				
		1shot pulse OFF D8_1 occurrence interrupt				
		UDID8_1 down-counter underflow interrupt				
INTATUCDSLDCD2	INTATUCDSLDCD2	GRC0_2 input capture/compare match interrupt	—	group1-66, group1-194	group3-2, group3-66	Selected by ATU_DMASEL_CD02[2:0] bits of ATUDMASELCD0 register.
		OCRC0_2 input capture/compare match interrupt				
		OCR1D8_2 compare match interrupt				
		OCR2D8_2 compare match interrupt				
		1shot pulse ON D8_2 occurrence interrupt				
		1shot pulse OFF D8_2 occurrence interrupt				
		UDID8_2 down-counter underflow interrupt				
INTATUCDSLDCD3	INTATUCDSLDCD3	GRC0_3 input capture/compare match interrupt	—	group1-67, group1-195	group3-3, group3-67	Selected by ATU_DMASEL_CD03[2:0] bits of ATUDMASELCD0 register.
		OCRC0_3 input capture/compare match interrupt				
		OCR1D8_3 compare match interrupt				
		OCR2D8_3 compare match interrupt				
		1shot pulse ON D8_3 occurrence interrupt				
		1shot pulse OFF D8_3 occurrence interrupt				
		UDID8_3 down-counter underflow interrupt				
INTATUCDSLDCD4	INTATUCDSLDCD4	GRC1_0 input capture/compare match interrupt	—	group1-68, group1-196	group3-4, group3-68	Selected by ATU_DMASEL_CD04[2:0] bits of ATUDMASELCD0 register.
		OCRC1_0 input capture/compare match interrupt				
INTATUCDSLDCD5	INTATUCDSLDCD5	GRC1_1 input capture/compare match interrupt	—	group1-69, group1-197	group3-5, group3-69	Selected by ATU_DMASEL_CD05[2:0] bits of ATUDMASELCD0 register.
		OCRC1_1 input capture/compare match interrupt				
INTATUCDSLDCD6	INTATUCDSLDCD6	GRC1_2 input capture/compare match interrupt	—	group1-70, group1-198	group3-6, group3-70	Selected by ATU_DMASEL_CD06[2:0] bits of ATUDMASELCD0 register.
		OCRC1_2 input capture/compare match interrupt				

Table 30.5 Interrupt Requests (15/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUCDSLDCD7	INTATUCDSLDCD7	GRC1_3 input capture/compare match interrupt	—	group1-71, group1-199	group3-7, group3-71	Selected by ATU_DMASEL_CD07[2:0] bits of ATUDMASELCD0 register.
		OCRC1_3 input capture/compare match interrupt				
INTATUCDSLDCD8	INTATUCDSLDCD8	GRC2_0 input capture/compare match interrupt	—	group1-72, group1-200	group3-8, group3-72	Selected by ATU_DMASEL_CD08[2:0] bits of ATUDMASELCD1 register.
		OCRC2_0 input capture/compare match interrupt				
INTATUCDSLDCD9	INTATUCDSLDCD9	GRC2_1 input capture/compare match interrupt	—	group1-73, group1-201	group3-9, group3-73	Selected by ATU_DMASEL_CD09[2:0] bits of ATUDMASELCD1 register.
		OCRC2_1 input capture/compare match interrupt				
INTATUCDSLDCD10	INTATUCDSLDCD10	GRC2_2 input capture/compare match interrupt	—	group1-74, group1-202	group3-10, group3-74	Selected by ATU_DMASEL_CD10[2:0] bits of ATUDMASELCD1 register.
		OCRC2_2 input capture/compare match interrupt				
INTATUCDSLDCD11	INTATUCDSLDCD11	GRC2_3 input capture/compare match interrupt	—	group1-75, group1-203	group3-11, group3-75	Selected by ATU_DMASEL_CD11[2:0] bits of ATUDMASELCD1 register.
		OCRC2_3 input capture/compare match interrupt				
INTATUCDSLDCD12	INTATUCDSLDCD12	GRC3_0 input capture/compare match interrupt	—	group1-76, group1-204	group3-12, group3-76	Selected by ATU_DMASEL_CD12[2:0] bits of ATUDMASELCD1 register.
		OCRC3_0 input capture/compare match interrupt				
INTATUCDSLDCD13	INTATUCDSLDCD13	GRC3_1 input capture/compare match interrupt	—	group1-77, group1-205	group3-13, group3-77	Selected by ATU_DMASEL_CD13[2:0] bits of ATUDMASELCD1 register.
		OCRC3_1 input capture/compare match interrupt				
INTATUCDSLDCD14	INTATUCDSLDCD14	GRC3_2 input capture/compare match interrupt	—	group1-78, group1-206	group3-14, group3-78	Selected by ATU_DMASEL_CD14[2:0] bits of ATUDMASELCD1 register.
		OCRC3_2 input capture/compare match interrupt				
INTATUCDSLDCD15	INTATUCDSLDCD15	GRC3_3 input capture/compare match interrupt	—	group1-79, group1-207	group3-15, group3-79	Selected by ATU_DMASEL_CD15[2:0] bits of ATUDMASELCD1 register.
		OCRC3_3 input capture/compare match interrupt				
INTATUCDSLDCD16	INTATUCDSLDCD16	GRC4_0 input capture/compare match interrupt	—	group1-80, group1-208	group3-16, group3-80	Selected by ATU_DMASEL_CD16[2:0] bits of ATUDMASELCD2 register.
		OCRC4_0 input capture/compare match interrupt				
INTATUCDSLDCD17	INTATUCDSLDCD17	GRC4_1 input capture/compare match interrupt	—	group1-81, group1-209	group3-17, group3-81	Selected by ATU_DMASEL_CD17[2:0] bits of ATUDMASELCD2 register.
		OCRC4_1 input capture/compare match interrupt				
INTATUCDSLDCD18	INTATUCDSLDCD18	GRC4_2 input capture/compare match interrupt	—	group1-82, group1-210	group3-18, group3-82	Selected by ATU_DMASEL_CD18[2:0] bits of ATUDMASELCD2 register.
		OCRC4_2 input capture/compare match interrupt				
INTATUCDSLDCD19	INTATUCDSLDCD19	GRC4_3 input capture/compare match interrupt	—	group1-83, group1-211	group3-19, group3-83	Selected by ATU_DMASEL_CD19[2:0] bits of ATUDMASELCD2 register.
		OCRC4_3 input capture/compare match interrupt				
INTATUCDSLDCD20	INTATUCDSLDCD20	GRC5_0 input capture/compare match interrupt	—	group1-84, group1-212	group3-20, group3-84	Selected by ATU_DMASEL_CD20[2:0] bits of ATUDMASELCD2 register.
		OCRC5_0 input capture/compare match interrupt				
INTATUCDSLDCD21	INTATUCDSLDCD21	GRC5_1 input capture/compare match interrupt	—	group1-85, group1-213	group3-21, group3-85	Selected by ATU_DMASEL_CD21[2:0] bits of ATUDMASELCD2 register.
		OCRC5_1 input capture/compare match interrupt				

Table 30.5 Interrupt Requests (16/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUCDSLDCD22	INTATUCDSLDCD22	GRC5_2 input capture/compare match interrupt	—	group1-86, group1-214	group3-22, group3-86	Selected by ATU_DMASEL_CD22[2:0] bits of ATUDMASELCD2 register.
		OCRC5_2 input capture/compare match interrupt				
INTATUCDSLDCD23	INTATUCDSLDCD23	GRC5_3 input capture/compare match interrupt	—	group1-87, group1-215	group3-23, group3-87	Selected by ATU_DMASEL_CD23[2:0] bits of ATUDMASELCD2 register.
		OCRC5_3 input capture/compare match interrupt				
INTATUCDSLDCD24	INTATUCDSLDCD24	GRC6_0 input capture/compare match interrupt	—	group1-88, group1-216	group3-24, group3-88	Selected by ATU_DMASEL_CD24[2:0] bits of ATUDMASELCD3 register.
		OCRC6_0 input capture/compare match interrupt				
INTATUCDSLDCD25	INTATUCDSLDCD25	GRC6_1 input capture/compare match interrupt	—	group1-89, group1-217	group3-25, group3-89	Selected by ATU_DMASEL_CD25[2:0] bits of ATUDMASELCD3 register.
		OCRC6_1 input capture/compare match interrupt				
INTATUCDSLDCD26	INTATUCDSLDCD26	GRC6_2 input capture/compare match interrupt	—	group1-90, group1-218	group3-26, group3-90	Selected by ATU_DMASEL_CD26[2:0] bits of ATUDMASELCD3 register.
		OCRC6_2 input capture/compare match interrupt				
INTATUCDSLDCD27	INTATUCDSLDCD27	GRC6_3 input capture/compare match interrupt	—	group1-91, group1-219	group3-27, group3-91	Selected by ATU_DMASEL_CD27[2:0] bits of ATUDMASELCD3 register.
		OCRC6_3 input capture/compare match interrupt				
INTATUCDSLDCD28	INTATUCDSLDCD28	GRC7_0 input capture/compare match interrupt	—	group1-92, group1-220	group3-28, group3-92	Selected by ATU_DMASEL_CD28[2:0] bits of ATUDMASELCD3 register.
		OCRC7_0 input capture/compare match interrupt				
INTATUCDSLDCD29	INTATUCDSLDCD29	GRC7_1 input capture/compare match interrupt	—	group1-93, group1-221	group3-29, group3-93	Selected by ATU_DMASEL_CD29[2:0] bits of ATUDMASELCD3 register.
		OCRC7_1 input capture/compare match interrupt				
INTATUCDSLDCD30	INTATUCDSLDCD30	GRC7_2 input capture/compare match interrupt	—	group1-94, group1-222	group3-30, group3-94	Selected by ATU_DMASEL_CD30[2:0] bits of ATUDMASELCD3 register.
		OCRC7_2 input capture/compare match interrupt				
INTATUCDSLDCD31	INTATUCDSLDCD31	GRC7_3 input capture/compare match interrupt	—	group1-95, group1-223	group3-31, group3-95	Selected by ATU_DMASEL_CD31[2:0] bits of ATUDMASELCD3 register.
		OCRC7_3 input capture/compare match interrupt				
INTATUCDSLDCD32	INTATUCDSLDCD32	GRC8_0 input capture/compare match interrupt	—	group1-96, group1-224	group3-32, group3-96	Selected by ATU_DMASEL_CD32[2:0] bits of ATUDMASELCD4 register.
		OCRC8_0 input capture/compare match interrupt				
		OCR1D0_0 compare match interrupt				
		OCR2D0_0 compare match interrupt				
		1shot pulse ON D0_0 occurrence interrupt				
		1shot pulse OFF D0_0 occurrence interrupt				
UDID0_0 down-counter underflow interrupt						

Table 30.5 Interrupt Requests (17/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUCDSLDCD33	INTATUCDSLDCD33	GRC8_1 input capture/compare match interrupt	—	group1-97, group1-225	group3-33, group3-97	Selected by ATU_DMASEL_CD33[2:0] bits of ATUDMASELCD4 register.
		OCRC8_1 input capture/compare match interrupt				
		OCR1D0_1 compare match interrupt				
		OCR2D0_1 compare match interrupt				
		1shot pulse ON D0_1 occurrence interrupt				
		1shot pulse OFF D0_1 occurrence interrupt				
		UDID0_1 down-counter underflow interrupt				
INTATUCDSLDCD34	INTATUCDSLDCD34	GRC8_2 input capture/compare match interrupt	—	group1-98, group1-226	group3-34, group3-98	Selected by ATU_DMASEL_CD34[2:0] bits of ATUDMASELCD4 register.
		OCRC8_2 input capture/compare match interrupt				
		OCR1D0_2 compare match interrupt				
		OCR2D0_2 compare match interrupt				
		1shot pulse ON D0_2 occurrence interrupt				
		1shot pulse OFF D0_2 occurrence interrupt				
		UDID0_2 down-counter underflow interrupt				
INTATUCDSLDCD35	INTATUCDSLDCD35	GRC8_3 input capture/compare match interrupt	—	group1-99, group1-227	group3-35, group3-99	Selected by ATU_DMASEL_CD35[2:0] bits of ATUDMASELCD4 register.
		OCRC8_3 input capture/compare match interrupt				
		OCR1D0_3 compare match interrupt				
		OCR2D0_3 compare match interrupt				
		1shot pulse ON D0_3 occurrence interrupt				
		1shot pulse OFF D0_3 occurrence interrupt				
		UDID0_3 down-counter underflow interrupt				
INTATUCDSLDCD36	INTATUCDSLDCD36	GRC9_0 input capture/compare match interrupt	—	group1-100, group1-228	group3-36, group3-100	Selected by ATU_DMASEL_CD36[2:0] bits of ATUDMASELCD4 register.
		OCRC9_0 input capture/compare match interrupt				
		OCR1D1_0 compare match interrupt				
		OCR2D1_0 compare match interrupt				
		1shot pulse ON D1_0 occurrence interrupt				
		1shot pulse OFF D1_0 occurrence interrupt				
		UDID1_0 down-counter underflow interrupt				
INTATUCDSLDCD37	INTATUCDSLDCD37	GRC9_1 input capture/compare match interrupt	—	group1-101, group1-229	group3-37, group3-101	Selected by ATU_DMASEL_CD37[2:0] bits of ATUDMASELCD4 register.
		OCRC9_1 input capture/compare match interrupt				
		OCR1D1_1 compare match interrupt				
		OCR2D1_1 compare match interrupt				
		1shot pulse ON D1_1 occurrence interrupt				
		1shot pulse OFF D1_1 occurrence interrupt				
		UDID1_1 down-counter underflow interrupt				

Table 30.5 Interrupt Requests (18/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUCDSLDCD38	INTATUCDSLDCD38	GRC9_2 input capture/compare match interrupt	—	group1-102, group1-230	group3-38, group3-102	Selected by ATU_DMASEL_CD38[2:0] bits of ATUDMASELCD4 register.
		OCRC9_2 input capture/compare match interrupt				
		OCR1D1_2 compare match interrupt				
		OCR2D1_2 compare match interrupt				
		1shot pulse ON D1_2 occurrence interrupt				
		1shot pulse OFF D1_2 occurrence interrupt				
		UDID1_2 down-counter underflow interrupt				
INTATUCDSLDCD39	INTATUCDSLDCD39	GRC9_3 input capture/compare match interrupt	—	group1-103, group1-231	group3-39, group3-103	Selected by ATU_DMASEL_CD39[2:0] bits of ATUDMASELCD4 register.
		OCRC9_3 input capture/compare match interrupt				
		OCR1D1_3 compare match interrupt				
		OCR2D1_3 compare match interrupt				
		1shot pulse ON D1_3 occurrence interrupt				
		1shot pulse OFF D1_3 occurrence interrupt				
		UDID1_3 down-counter underflow interrupt				
INTATUCDSLDCD40	INTATUCDSLDCD40	GRC10_0 input capture/compare match interrupt	—	group1-104, group1-232	group3-40, group3-104	Selected by ATU_DMASEL_CD40[2:0] bits of ATUDMASELCD5 register.
		OCRC10_0 input capture/compare match interrupt				
		OCR1D2_0 compare match interrupt				
		OCR2D2_0 compare match interrupt				
		1shot pulse ON D2_0 occurrence interrupt				
		1shot pulse OFF D2_0 occurrence interrupt				
		UDID2_0 down-counter underflow interrupt				
INTATUCDSLDCD41	INTATUCDSLDCD41	GRC10_1 input capture/compare match interrupt	—	group1-105, group1-233	group3-41, group3-105	Selected by ATU_DMASEL_CD41[2:0] bits of ATUDMASELCD5 register.
		OCRC10_1 input capture/compare match interrupt				
		OCR1D2_1 compare match interrupt				
		OCR2D2_1 compare match interrupt				
		1shot pulse ON D2_1 occurrence interrupt				
		1shot pulse OFF D2_1 occurrence interrupt				
		UDID2_1 down-counter underflow interrupt				
INTATUCDSLDCD42	INTATUCDSLDCD42	GRC10_2 input capture/compare match interrupt	—	group1-106, group1-234	group3-42, group3-106	Selected by ATU_DMASEL_CD42[2:0] bits of ATUDMASELCD5 register.
		OCRC10_2 input capture/compare match interrupt				
		OCR1D2_2 compare match interrupt				
		OCR2D2_2 compare match interrupt				
		1shot pulse ON D2_2 occurrence interrupt				
		1shot pulse OFF D2_2 occurrence interrupt				
		UDID2_2 down-counter underflow interrupt				

Table 30.5 Interrupt Requests (19/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUCDSLDCD43	INTATUCDSLDCD43	GRC10_3 input capture/compare match interrupt	—	group1-107, group1-235	group3-43, group3-107	Selected by ATU_DMASEL_CD43[2:0] bits of ATUDMASELCD5 register.
		OCRC10_3 input capture/compare match interrupt				
		OCR1D2_3 compare match interrupt				
		OCR2D2_3 compare match interrupt				
		1shot pulse ON D2_3 occurrence interrupt				
		1shot pulse OFF D2_3 occurrence interrupt				
		UDID2_3 down-counter underflow interrupt				
INTATUCDSLDCD44	INTATUCDSLDCD44	OCR1D3_0 compare match interrupt	—	group1-108, group1-236	group3-44, group3-108	Selected by ATU_DMASEL_CD44[2:0] bits of ATUDMASELCD5 register.
		OCR2D3_0 compare match interrupt				
		1shot pulse ON D3_0 occurrence interrupt				
		1shot pulse OFF D3_0 occurrence interrupt				
		UDID3_0 down-counter underflow interrupt				
INTATUCDSLDCD45	INTATUCDSLDCD45	OCR1D3_1 compare match interrupt	—	group1-109, group1-237	group3-45, group3-109	Selected by ATU_DMASEL_CD45[2:0] bits of ATUDMASELCD5 register.
		OCR2D3_1 compare match interrupt				
		1shot pulse ON D3_1 occurrence interrupt				
		1shot pulse OFF D3_1 occurrence interrupt				
		UDID3_1 down-counter underflow interrupt				
INTATUCDSLDCD46	INTATUCDSLDCD46	OCR1D3_2 compare match interrupt	—	group1-110, group1-238	group3-46, group3-110	Selected by ATU_DMASEL_CD46[2:0] bits of ATUDMASELCD5 register.
		OCR2D3_2 compare match interrupt				
		1shot pulse ON D3_2 occurrence interrupt				
		1shot pulse OFF D3_2 occurrence interrupt				
		UDID3_2 down-counter underflow interrupt				
INTATUCDSLDCD47	INTATUCDSLDCD47	OCR1D3_3 compare match interrupt	—	group1-111, group1-239	group3-47, group3-111	Selected by ATU_DMASEL_CD47[2:0] bits of ATUDMASELCD5 register.
		OCR2D3_3 compare match interrupt				
		1shot pulse ON D3_3 occurrence interrupt				
		1shot pulse OFF D3_3 occurrence interrupt				
		UDID3_3 down-counter underflow interrupt				
INTATUCDSLDCD48	INTATUCDSLDCD48	OCR1D4_0 compare match interrupt	—	group1-112, group1-240	group3-48, group3-112	Selected by ATU_DMASEL_CD48[2:0] bits of ATUDMASELCD6 register.
		OCR2D4_0 compare match interrupt				
		1shot pulse ON D4_0 occurrence interrupt				
		1shot pulse OFF D4_0 occurrence interrupt				
		UDID4_0 down-counter underflow interrupt				

Table 30.5 Interrupt Requests (20/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUCDSLDCD49	INTATUCDSLDCD49	OCR1D4_1 compare match interrupt	—	group1-113, group1-241	group3-49, group3-113	Selected by ATU_DMASEL_CD49[2:0] bits of ATUDMASELCD6 register.
		OCR2D4_1 compare match interrupt				
		1shot pulse ON D4_1 occurrence interrupt				
		1shot pulse OFF D4_1 occurrence interrupt				
		UDID4_1 down-counter underflow interrupt				
INTATUCDSLDCD50	INTATUCDSLDCD50	OCR1D4_2 compare match interrupt	—	group1-114, group1-242	group3-50, group3-114	Selected by ATU_DMASEL_CD50[2:0] bits of ATUDMASELCD6 register.
		OCR2D4_2 compare match interrupt				
		1shot pulse ON D4_2 occurrence interrupt				
		1shot pulse OFF D4_2 occurrence interrupt				
		UDID4_2 down-counter underflow interrupt				
INTATUCDSLDCD51	INTATUCDSLDCD51	OCR1D4_3 compare match interrupt	—	group1-115, group1-243	group3-51, group3-115	Selected by ATU_DMASEL_CD51[2:0] bits of ATUDMASELCD6 register.
		OCR2D4_3 compare match interrupt				
		1shot pulse ON D4_3 occurrence interrupt				
		1shot pulse OFF D4_3 occurrence interrupt				
		UDID4_3 down-counter underflow interrupt				
INTATUCDSLDCD52	INTATUCDSLDCD52	OCR1D5_0 compare match interrupt	—	group1-116, group1-244	group3-52, group3-116	Selected by ATU_DMASEL_CD52[2:0] bits of ATUDMASELCD6 register.
		OCR2D5_0 compare match interrupt				
		1shot pulse ON D5_0 occurrence interrupt				
		1shot pulse OFF D5_0 occurrence interrupt				
		UDID5_0 down-counter underflow interrupt				
INTATUCDSLDCD53	INTATUCDSLDCD53	OCR1D5_1 compare match interrupt	—	group1-117, group1-245	group3-53, group3-117	Selected by ATU_DMASEL_CD53[2:0] bits of ATUDMASELCD6 register.
		OCR2D5_1 compare match interrupt				
		1shot pulse ON D5_1 occurrence interrupt				
		1shot pulse OFF D5_1 occurrence interrupt				
		UDID5_1 down-counter underflow interrupt				
INTATUCDSLDCD54	INTATUCDSLDCD54	OCR1D5_2 compare match interrupt	—	group1-118, group1-246	group3-54, group3-118	Selected by ATU_DMASEL_CD54[2:0] bits of ATUDMASELCD6 register.
		OCR2D5_2 compare match interrupt				
		1shot pulse ON D5_2 occurrence interrupt				
		1shot pulse OFF D5_2 occurrence interrupt				
		UDID5_2 down-counter underflow interrupt				
INTATUCDSLDCD55	INTATUCDSLDCD55	OCR1D5_3 compare match interrupt	—	group1-119, group1-247	group3-55, group3-119	Selected by ATU_DMASEL_CD55[2:0] bits of ATUDMASELCD6 register.
		OCR2D5_3 compare match interrupt				
		1shot pulse ON D5_3 occurrence interrupt				
		1shot pulse OFF D5_3 occurrence interrupt				
		UDID5_3 down-counter underflow interrupt				

Table 30.5 Interrupt Requests (21/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUCDSLDCD56	INTATUCDSLDCD56	OCR1D6_0 compare match interrupt	—	group1-120, group1-248	group3-56, group3-120	Selected by ATU_DMASEL_CD56[2:0] bits of ATUDMASELCD7 register.
		OCR2D6_0 compare match interrupt				
		1shot pulse ON D6_0 occurrence interrupt				
		1shot pulse OFF D6_0 occurrence interrupt				
		UDID6_0 down-counter underflow interrupt				
INTATUCDSLDCD57	INTATUCDSLDCD57	OCR1D6_1 compare match interrupt	—	group1-121, group1-249	group3-57, group3-121	Selected by ATU_DMASEL_CD57[2:0] bits of ATUDMASELCD7 register.
		OCR2D6_1 compare match interrupt				
		1shot pulse ON D6_1 occurrence interrupt				
		1shot pulse OFF D6_1 occurrence interrupt				
		UDID6_1 down-counter underflow interrupt				
INTATUCDSLDCD58	INTATUCDSLDCD58	OCR1D6_2 compare match interrupt	—	group1-122, group1-250	group3-58, group3-122	Selected by ATU_DMASEL_CD58[2:0] bits of ATUDMASELCD7 register.
		OCR2D6_2 compare match interrupt				
		1shot pulse ON D6_2 occurrence interrupt				
		1shot pulse OFF D6_2 occurrence interrupt				
		UDID6_2 down-counter underflow interrupt				
INTATUCDSLDCD59	INTATUCDSLDCD59	OCR1D6_3 compare match interrupt	—	group1-123, group1-251	group3-59, group3-123	Selected by ATU_DMASEL_CD59[2:0] bits of ATUDMASELCD7 register.
		OCR2D6_3 compare match interrupt				
		1shot pulse ON D6_3 occurrence interrupt				
		1shot pulse OFF D6_3 occurrence interrupt				
		UDID6_3 down-counter underflow interrupt				
INTATUCDSLDCD60	INTATUCDSLDCD60	OCR1D7_0 compare match interrupt	—	group1-124, group1-252	group3-60, group3-124	Selected by ATU_DMASEL_CD60[2:0] bits of ATUDMASELCD7 register.
		OCR2D7_0 compare match interrupt				
		1shot pulse ON D7_0 occurrence interrupt				
		1shot pulse OFF D7_0 occurrence interrupt				
		UDID7_0 down-counter underflow interrupt				
INTATUCDSLDCD61	INTATUCDSLDCD61	OCR1D7_1 compare match interrupt	—	group1-125, group1-253	group3-61, group3-125	Selected by ATU_DMASEL_CD61[2:0] bits of ATUDMASELCD7 register.
		OCR2D7_1 compare match interrupt				
		1shot pulse ON D7_1 occurrence interrupt				
		1shot pulse OFF D7_1 occurrence interrupt				
		UDID7_1 down-counter underflow interrupt				
INTATUCDSLDCD62	INTATUCDSLDCD62	OCR1D7_2 compare match interrupt	—	group1-126, group1-254	group3-62, group3-126	Selected by ATU_DMASEL_CD62[2:0] bits of ATUDMASELCD7 register.
		OCR2D7_2 compare match interrupt				
		1shot pulse ON D7_2 occurrence interrupt				
		1shot pulse OFF D7_2 occurrence interrupt				
		UDID7_2 down-counter underflow interrupt				

Table 30.5 Interrupt Requests (22/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUCDSLDCD63	INTATUCDSLDCD63	OCR1D7_3 compare match interrupt	—	group1-127, group1-255	group3-63, group3-127	Selected by ATU_DMASEL_CD63[2:0] bits of ATUDMASELCD7 register.
		OCR2D7_3 compare match interrupt				
		1shot pulse ON D7_3 occurrence interrupt				
		1shot pulse OFF D7_3 occurrence interrupt				
		UDID7_3 down-counter underflow interrupt				
INTATUESLDE0	INTATUESLDE0	CYLRE0_0 cycle compare match interrupt/DTRE0_0 duty compare match interrupt	—	group1-0, group1-128	group2-0, group2-64	Selected by ATU_DMASEL_E0[1:0] bits of ATUDMASELE register.
		CYLRE0_1 cycle compare match interrupt/DTRE0_1 duty compare match interrupt				
		CYLRE0_2 cycle compare match interrupt/DTRE0_2 duty compare match interrupt				
		CYLRE0_3 cycle compare match interrupt/DTRE0_3 duty compare match interrupt				
INTATUESLDE1	INTATUESLDE1	CYLRE1_0 cycle compare match interrupt/DTRE1_0 duty compare match interrupt	—	group1-1, group1-129	group2-1, group2-65	Selected by ATU_DMASEL_E1[1:0] bits of ATUDMASELE register.
		CYLRE1_1 cycle compare match interrupt/DTRE1_1 duty compare match interrupt				
		CYLRE1_2 cycle compare match interrupt/DTRE1_2 duty compare match interrupt				
		CYLRE1_3 cycle compare match interrupt/DTRE1_3 duty compare match interrupt				
INTATUESLDE2	INTATUESLDE2	CYLRE2_0 cycle compare match interrupt/DTRE2_0 duty compare match interrupt	—	group1-2, group1-130	group2-2, group2-66	Selected by ATU_DMASEL_E2[1:0] bits of ATUDMASELE register.
		CYLRE2_1 cycle compare match interrupt/DTRE2_1 duty compare match interrupt				
		CYLRE2_2 cycle compare match interrupt/DTRE2_2 duty compare match interrupt				
		CYLRE2_3 cycle compare match interrupt/DTRE2_3 duty compare match interrupt				
INTATUESLDE3	INTATUESLDE3	CYLRE3_0 cycle compare match interrupt/DTRE3_0 duty compare match interrupt	—	group1-3, group1-131	group2-3, group2-67	Selected by ATU_DMASEL_E3[1:0] bits of ATUDMASELE register.
		CYLRE3_1 cycle compare match interrupt/DTRE3_1 duty compare match interrupt				
		CYLRE3_2 cycle compare match interrupt/DTRE3_2 duty compare match interrupt				
		CYLRE3_3 cycle compare match interrupt/DTRE3_3 duty compare match interrupt				

Table 30.5 Interrupt Requests (23/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUESLDE4	INTATUESLDE4	CYLRE4_0 cycle compare match interrupt/DTRE4_0 duty compare match interrupt	—	group1-4, group1-132	group2-4, group2-68	Selected by ATU_DMASEL_E4[1:0] bits of ATUDMASELE register.
		CYLRE4_1 cycle compare match interrupt/DTRE4_1 duty compare match interrupt				
		CYLRE4_2 cycle compare match interrupt/DTRE4_2 duty compare match interrupt				
		CYLRE4_3 cycle compare match interrupt/DTRE4_3 duty compare match interrupt				
INTATUESLDE5	INTATUESLDE5	CYLRE5_0 cycle compare match interrupt/DTRE5_0 duty compare match interrupt	—	group1-5, group1-133	group2-5, group2-69	Selected by ATU_DMASEL_E5[1:0] bits of ATUDMASELE register.
		CYLRE5_1 cycle compare match interrupt/DTRE5_1 duty compare match interrupt				
		CYLRE5_2 cycle compare match interrupt/DTRE5_2 duty compare match interrupt				
		CYLRE5_3 cycle compare match interrupt/DTRE5_3 duty compare match interrupt				
INTATUESLDE6	INTATUESLDE6	CYLRE6_0 cycle compare match interrupt/DTRE6_0 duty compare match interrupt	—	group1-6, group1-134	group2-6, group2-70	Selected by ATU_DMASEL_E6[1:0] bits of ATUDMASELE register.
		CYLRE6_1 cycle compare match interrupt/DTRE6_1 duty compare match interrupt				
		CYLRE6_2 cycle compare match interrupt/DTRE6_2 duty compare match interrupt				
		CYLRE6_3 cycle compare match interrupt/DTRE6_3 duty compare match interrupt				
INTATUESLDE7	INTATUESLDE7	CYLRE7_0 cycle compare match interrupt/DTRE7_0 duty compare match interrupt	—	group1-7, group1-135	group2-7, group2-71	Selected by ATU_DMASEL_E7[1:0] bits of ATUDMASELE register.
		CYLRE7_1 cycle compare match interrupt/DTRE7_1 duty compare match interrupt				
		CYLRE7_2 cycle compare match interrupt/DTRE7_2 duty compare match interrupt				
		CYLRE7_3 cycle compare match interrupt/DTRE7_3 duty compare match interrupt				
INTATUESLDE8	INTATUESLDE8	CYLRE8_0 cycle compare match interrupt/DTRE8_0 duty compare match interrupt	—	group1-8, group1-136	group2-8, group2-72	Selected by ATU_DMASEL_E8[1:0] bits of ATUDMASELE register.
		CYLRE8_1 cycle compare match interrupt/DTRE8_1 duty compare match interrupt				
		CYLRE8_2 cycle compare match interrupt/DTRE8_2 duty compare match interrupt				
		CYLRE8_3 cycle compare match interrupt/DTRE8_3 duty compare match interrupt				

Table 30.5 Interrupt Requests (24/24)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Note.
INTATUCTRLDMARE Q00	INTATUCTRLDMARE Q00	AD skipping DMA transfer request 0	—	group1-60, group1-188	group2-60, group2-124	—
INTATUCTRLDMARE Q01	INTATUCTRLDMARE Q01	AD skipping DMA transfer request 1	—	group1-61, group1-189	group2-61, group2-125	—
INTATUCTRLDMARE Q10	INTATUCTRLDMARE Q10	AD skipping DMA transfer request 2	—	group1-62, group1-190	group2-62, group2-126	—
INTATUCTRLDMARE Q11	INTATUCTRLDMARE Q11	AD skipping DMA transfer request 3	—	group1-63, group1-191	group2-63, group2-127	—

30.1.5 Reset Sources

ATU-V reset sources are listed in the following table. ATU-V is initialized by these reset sources.

Table 30.6 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
ATU-Vn	All registers	✓	✓	✓	✓	—	—

30.1.6 External Input/Output Signals

External input/output signals of ATU-V are listed below.

Controller

Table 30.7 External Input/Output Signals of Controller

Symbol	I/O	Number of Bits	Function
TCLKA	Input	1	Input pin for the external clock on signal line 4 of the clock bus
TCLKB	Input	1	Input pin for the external clock on signal line 5 of the clock bus

Prescaler

Not provided

Timer A

Table 30.8 External Input/Output Signals Timer A (1/2)

RH850/E2x-FCC1			
Symbol	I/O	Number of Bits	Function
TIA00 to TIA07	Input	8	Input pins for input-capture triggers for timer A channels

Table 30.8 External Input/Output Signals Timer A (2/2)

RH850/E2M			
Symbol	I/O	Number of Bits	Function
TIA00 to TIA07	Input	8	Input pins for input-capture triggers for timer A channels

Timer B

Not provided

Timer C

Table 30.9 External Input/Output Signals Timer C (1/2)

RH850/E2x-FCC1			
Symbol	I/O	Number of Bits	Function
TIOC00 - 03 to TIOC90 - 93	Input/Output	40	Output pins for input-capture triggers and output pins for output compare signals of timer C (one each for channels 0 to 3 in sub blocks C0 to C9)

Table 30.9 External Input/Output Signals Timer C (2/2)

RH850/E2M			
Symbol	I/O	Number of Bits	Function
TIOC00 - 03 to TIOC90 - 93	Input/Output	40	Output pins for input-capture triggers and output pins for output compare signals of timer C (one each for channels 0 to 3 in sub blocks C0 to C9)

Timer D

Table 30.10 External Input/Output Signals Timer D (1/2)

RH850/E2x-FCC1			
Symbol	I/O	Number of Bits	Function
TOD00A - 03A to TOD80A - 83A	Output	36	Output pins for compare-match signals of timer D (one each for channels 0 to 3 in subblocks D0 to D8)
TOD00B - 03B to TOD80B - 83B	Output	36	Output pins for one-shot pulses of timer D (one each for channels 0 to 3 in sub blocks D0 to D8)

Table 30.10 External Input/Output Signals Timer D (2/2)

RH850/E2M			
Symbol	I/O	Number of Bits	Function
TOD00A - 03A to TOD80A - 83A	Output	36	Output pins for compare-match signals of timer D (one each for channels 0 to 3 in subblocks D0 to D8)
TOD00B - 03B to TOD80B - 83B	Output	36	Output pins for one-shot pulses of timer D (one each for channels 0 to 3 in sub blocks D0 to D8)

Timer E

Table 30.11 External Input/Output Signals Timer E (1/2)

RH850/E2x-FCC1			
Symbol	I/O	Number of Bits	Function
TOE00 - 03 to TOE80 - 83	Output	36	Output pins for PWM signals of timer E (one each for channels 0 to 3 in sub blocks E0 to E8)

Table 30.11 External Input/Output Signals Timer E (2/2)

RH850/E2M			
Symbol	I/O	Number of Bits	Function
TOE00 - 03 to TOE80 - 83	Output	36	Output pins for PWM signals of timer E (one each for channels 0 to 3 in sub blocks E0 to E8)

Timer F

Table 30.12 External Input/Output Signals Timer F (1/2)

RH850/E2x-FCC1			
Symbol	I/O	Number of Bits	Function
TIF0A to TIF2A TIF3 to TIF15	Input	16	Input pins for event signals for sub blocks F0 to F15 of timer F TIF0A to TIF2A: Input pins for sub blocks F0 to F2 TIF3 to TIF15: Input pins for sub blocks F3 to F15
TIF0B to TIF2B	Input	3	Input pins for event signals for subblocks F0 to F2 of timer F

Table 30.12 External Input/Output Signals Timer F (2/2)

RH850/E2M			
Symbol	I/O	Number of Bits	Function
TIF0A to TIF2A TIF3 to TIF15	Input	16	Input pins for event signals for sub blocks F0 to F15 of timer F TIF0A to TIF2A: Input pins for sub blocks F0 to F2 TIF3 to TIF15: Input pins for sub blocks F3 to F15
TIF0B to TIF2B	Input	3	Input pins for event signals for subblocks F0 to F2 of timer F

Timer G

Not provided

30.2 Overview

The ATU-V has the following features.

- ATU-V consists of 7 timer blocks (timer A to timer G), prescalers, and a controller. The timer blocks have different functions and each can operate independently; timer blocks can also be linked via the clock bus. Each timer block consists of one or more timer subblocks, and each subblock has one or more channels. The Numbers of subblocks/channels are shown in “**Table 30.13, Block Configuration**”, please refer.
 - Timer A has a free run counter, and can capture by an assertion of the external event inputs.
 - Timer B can generate a multiplied-and-corrected clock signal based on an external event input. It can also supply the generated signal to other timers.
 - Timer C is a general-purpose timer, input capture or output compare is selectable.
 - Timer D can output one-shot pulses.
 - Timer E is a PWM output timer.
 - Timer F can be used as a measurement timer for external inputs. 7 operation modes are available.
 - Timer G has a counter and can use as an interval timer.
- Equipped with a function to output pulses to A/D and DFE
- On-chip 4-channel prescaler provided, which generates four types of clocks by dividing on-chip peripheral clock (PCLK) by 1/1 to 1/1024
- Each channel for a timer can select a count source from among four divided clocks generated by prescaler, two external clocks, and angle clock generated by timer B.
- Pin output values of timers C, D, and E can be selected by the RHSB cross bar and then output as data to the RHSB.

<<Timer A>>

Timer A has a 32-bit free-run counter and SB_A 32-bit input capture registers. Its features are shown below.

- Detection by rising edges, falling edges, or both edges
- Activating the DMAC at the capture timing
- Noise canceling function for each external input pin with maximum length of 0.21 s or noise cancelling function using the angle clock
- Selectable noise canceling mode per channel
- The level of each external input pin can be read.
- A capture interrupt request and a counter overflow interrupt request can be generated.
- Signals after noise cancellation can be sent to external input pins of timer F.

<<Timer B>>

Timer B consists of three subblocks: an edge-interval measuring block, frequency-multiplied clock generator, and frequency-multiplied clock signal corrector.

Edge-Interval Measuring Block

The edge-interval measuring block is provided with a 32-bit input edge-interval measuring timer, three output compare and input capture registers, six edge-interval time measured value recording registers, seven edge-interval time measured value history registers, 8-bit event counter, and output compare register. This provides the following operations:

- Capture by edges of external event input (rising edge, falling edge, or both edges are selectable)
- Capture by event compare match of external event input
- Automatic event counter clear by event compare match
- Capture interrupt and compare match interrupt (edge-interval compare match, event compare match)
- Captures event counter values individually by detection of SB_A external event inputs from timer A.
- Generates reference data for down-count values used in the frequency-multiplied clock generator, from 24 sequencer states.

Frequency-Multiplied Clock Generator

The frequency-multiplied clock generator is provided with 24-bit reloadable counter, reload register, 20-bit multiplied clock counter, 26-bit multiplied clock counter, capture register, and output compare register. This provides the following operations:

- Reloadable counting of values captured by edge-interval measuring block with arbitrary number (1 to 4095)
- Reloadable down-count values can be modified by event compare match of the edge-interval measuring block.
- Internal clock generated by the underflow of reloadable counter can be used as input of 20-bit multiplied clock counter and 26-bit multiplied clock counter.
- The 20-bit multiplied clock counter value can be captured by an edge input of external input event.
- An interrupt request for compare match between 20-bit multiplied clock counter and output compare register and an interrupt request for compare match between 26-bit multiplied clock counter and capture register can be generated.

Frequency-Multiplied Clock Signal Corrector

Frequency-multiplied clock signal corrector is provided with 20-bit correcting event counter, 20-bit correcting multiplied clock counter, 20-bit multiplied-and-corrected clock generating counter, and correcting counter clearing register. This provides the following operations:

- Frequency-multiplied correcting clock that serves as the count source for other timers can be generated based on the read count in frequency-multiplied clock generator.
- The counter of timer D can be cleared by compare match (enabling or disabling this function selectable) between multiplied-and-corrected clock generating counter and correcting counter clearing register.
- The correcting event counter can be automatically cleared at the event compare match timing due to an external input event of the edge-interval measuring block.

<<Timer C>>

Timer C consists of SB_C subblocks that have the same functions. Each subblock is provided with a 32-bit counter, four 32-bit general registers, four 32-bit compare match registers, four range comparison value setting registers, four 32-bit general mirror registers, four 32-bit output comparison mirror registers, and a 32-bit upper-limit value setting register. This provides the following operations:

- Choice of rising edge, falling edge, or both edge sensing as the edge of input capture trigger signal.
- When timer C is used for input capture, capturing is possible at the timing of event output 1, 2A, and 2B from timer A.
- Noise canceling function for each external input pin with maximum length of 0.21 s or noise cancelling function using the angle clock
- Selectable noise canceling mode per subblock or channel
- 1, 0, toggle, or one-shot pulse waveform can be output by a compare match.
- Three PWM waveforms can be output for each subblock in PWM mode.
- Input capture/compare match interrupt and overflow interrupt can be generated. Input capture/compare match interrupts activate DMAC.
- One-shot pulse mode is possible for output.
- Upper-limit value setting function for 32-bit counters
- The range compare value setting register enables the range compare function for compare match output by compare match register.

<<Timer D>>

Timer D consists of SB_D subblocks that have the same function. Each subblock is provided with two 32-bit counters, an offset base register, an output value register, and an output setting register, and four channels. Each channel is provided with two 32-bit output compare registers, two range comparison value setting registers, two input capture registers, a 32-bit down-counter for outputting one-shot pulse, and two 32-bit upper-limit value setting registers. This provides the following operations:

- Enables to start down-counter by software. One-shot pulse can also be generated.
- Compare match in compare match register can be used as the start trigger for down-counter. One-shot pulse with offset can also be generated.
- Compare match in compare match register can stop down-counter and forcibly cut off one-shot pulse output.
- Compare match in compare match register can be output.
- Compare match range can be set by the range comparison value setting register (range compare function).
- Compare match in compare match register can be used as a trigger for capturing the count value into input capture register.
- Counter value can be captured triggered by timer A.
- Provided with counter clearing function from timer B
- Generation of $8 \times SB_D$ compare match interrupts, $2 \times SB_D$ counter overflow interrupts, and $4 \times SB_D$ underflow interrupts. The $4 \times SB_D$ underflow interrupts correspond to DMAC activation.
- Output waveform can be inverted.
- The value set in the output value register can be output from each pin by the output setting register.

- Upper-limit value setting function for 32-bit counters
- The MIN guard function which guarantees the MIN width for the “the MIN width of ON waveform”, “the MIN width of the OFF waveform”, and “ON to next ON” is loaded to the output waveform.

<<Timer E>>

Timer E consists of SB_E subblocks that have the same function. Each subblock consists of four channels. Each channel is provided with a 24-bit counter, a 24-bit duty cycle setting register, a 24-bit cycle setting register, a 24-bit duty cycle reload register, and a 24-bit cycle reload register. This provides the following operations:

- PWM output with programmable cycle time and duty cycle ranging from 0 to 100% is possible by setting the cycle setting register and duty cycle setting register.
- The duty mode can be switched between on-state and off-state by setting the output control register.
- Values in duty cycle reload register and cycle reload register can be transferred to duty cycle setting register and cycle setting register at every cycle, respectively. The reload function can be enabled or disabled.
- Writing 000000_H to the counter can forcibly finish the current PWM cycle and start another PWM cycle.
- PWM output can be blocked by shutoff input.
- $4 \times 2 \times SB_E$ interrupt requests can be output for cycle match and duty match. Also, SB_E DMAC activation requests can be output by cycle match in channel 0 of each subblock.
- Compare match between TCNTExy and CYLRExy can forcibly be generated by setting the forced compare match bit.

<<Timer F>>

Timer F consists of SB_F subblocks. Each subblock is provided with two 32-bit counters, a 16-bit counter, three 32-bit general registers, a 16-bit general register, three 32-bit backup registers, and input processing block. This provides the following operations:

- Detection level can be selected from rising edge, falling edge or both edges
- TIA00 to TIA0SB_A-1 signals of timer A after noise cancellation can be used as external input signals.
- Noise canceling function for each external input pin with maximum length of 0.21 s or noise cancelling function using the angle clock
- Selectable noise canceling mode per subblock or channel
- Seven operation modes: edge counting in a specified period, effective edge input interval counting, measurement of time during high/low input, measurement of PWM input waveform timing, rotation speed/pulse measurement (subblocks 3 to SB_F-1), up/down count (subblocks 0 to 2), and four-time multiplication event count (subblocks 0 to 2).
- Activates DMAC by input capture interrupt request
- Generates an interrupt upon overflow, underflow, or capturing.
- A interrupt request can be generated on compare match in specified-period edge count mode, effective edge input interval measurement mode, input high/low level measurement mode, PWM input waveform measurement mode, up/down count mode, and four-time multiplication event count mode.

<<Timer G>>

Timer G consists of SB_G subblocks that have the same function. Each channel is provided with a 32-bit timer counter and compare match register. This provides the following operations:

- An event can be output by using compare match as a trigger, which is used as A/D activation/interrupt request trigger.
- Activates DMAC by compare match interrupt request

30.2.1 Configuration of ATU-V

ATU-V is a macro for engine control that operates by using multiple timer block in combination.

ATU-V consists of seven timer blocks from timer A to timer G, prescalers, and a common controller unit. The timer blocks have different functions and each can operate independently; timer blocks can also be linked via the clock bus. Each timer block consists of one or more timer subblocks that have the same function, and each subblock has one or more channels (see **Table 30.13, Block Configuration**).

Table 30.13 Block Configuration

Modules	Remarks	
Blocks	Total_ch: Total number of channels	
Subblock	SB: Number of subblocks SB_ch: Number of channels per subblock	
ATU-V	RH850/E2x-FCC1	RH850/E2M
Common controller unit	—	—
Prescaler	Total_ch: 4	Total_ch: 4
Channel 0		
...		
Channel 3		
Timer A	Total_ch: $SB_A = 8$	Total_ch: $SB_A = 8$
Channel 0		
...		
Channel SB_A-1		
Timer B	Total_ch: 1	Total_ch: 1
Timer C	Total_ch: $4 \times SB_C$	Total_ch: $4 \times SB_C$
Timer C0	SB: $SB_C = 11$	SB: $SB_C = 11$
Channel 0	SB_ch: 4	SB_ch: 4
...		
Channel 3		
Timers C1 to CSB_C-1		
Timer D	Total_ch: $4 \times SB_D$	Total_ch: $4 \times SB_D$
Timer D0	SB: $SB_D = 9$	SB: $SB_D = 9$
Channel 0	SB_ch: 4	SB_ch: 4
...		
Channel 3		
Timers D1 to DSB_D-1		
Timer E	Total_ch: $4 \times SB_E$	Total_ch: $4 \times SB_E$
Timer E0	SB: $SB_E = 9$	SB: $SB_E = 9$
Channel 0	SB_ch: 4	SB_ch: 4
...		
Channel 3		
Timers E1 to ESB_E-1		
Timer F	Total_ch: SB_F	Total_ch: SB_F
Timer F0	SB: $SB_F = 16$	SB: $SB_F = 16$
...	SB_ch: 1	SB_ch: 1
Timer FSB_F-1		
Timer G	Total_ch: SB_G	Total_ch: SB_G
Timer G0	SB: $SB_G = 10$	SB: $SB_G = 10$
...		
Timer GSB_G-1		
Automatic Switching of DMA and A/D Requests	—	—
Trigger select function	—	—

30.2.2 ATU-V Registers

Addresses of the ATU-V registers are shown below.

Common Controller Registers (FFE6 0000_H to FFE6 007F_H)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 0000 _H	NCMR			CBCNT			(Prohibited area)			ATUENR			ATU-V
FFE6 0004 _H to FFE6 007C _H	(Prohibited area)												

Prescaler Registers (FFE6 0080_H to FFE6 00BF_H)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 0080 _H	(Prohibited area)						PSCR0						PSC
FFE6 0084 _H	(Prohibited area)						PSCR1						
FFE6 0088 _H	(Prohibited area)						PSCR2						
FFE6 008C _H	(Prohibited area)						PSCR3						
FFE6 0090 _H to FFE6 00BC _H	(Prohibited area)												

DMA/AD Requests Automatic Switching Registers (FFE6 00C0_H to FFE6 00FF_H)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 00C0 _H	(Prohibited area)									TRGSRDMA0			PICSU
FFE6 00C4 _H	(Prohibited area)			TRGSELDMA01			(Prohibited area)			TRGSELDMA00			
FFE6 00C8 _H	(Prohibited area)									TRGSELAD			
FFE6 00CC _H	(Prohibited area)												
FFE6 00D0 _H	(Prohibited area)									TRGSRDMA1			
FFE6 00D4 _H	(Prohibited area)			TRGSELDMA11			(Prohibited area)			TRGSELDMA10			
FFE6 00D8 _H to FFE6 00FC _H	(Prohibited area)												

Timer A Registers (FFE6 0200_H to FFE6 03FF_H) (1/2)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 0200 _H	TCR2A					(Prohibited area)			TCR1A			Common to A	
FFE6 0204 _H	TCR4A					TCR3A							
FFE6 0208 _H	TSCRA					TSRA							
FFE6 020C _H	(Prohibited area)		NCMCR2A			(Prohibited area)		NCMCR1A					
FFE6 0210 _H	(Prohibited area)					TIOR1A							
FFE6 0214 _H	TIOR2A												
FFE6 0218 _H	(Prohibited area)					(Prohibited area)			TILRA				
FFE6 021C _H	(Prohibited area)										TCR5A		
FFE6 0220 _H	TCNTA												
FFE6 0224 _H	(Prohibited area)		TCR2AS2			(Prohibited area)		TCR2AS1					
FFE6 0228 _H	(Prohibited area)												
FFE6 022C _H	(Prohibited area)					TIERA							
FFE6 0230 _H to FFE6 023C _H	(Prohibited area)												
FFE6 0240 _H	NCRA0					NCNTA0					A0		
FFE6 0244 _H	ICRA0												
FFE6 0248 _H	ICRA20												
FFE6 024C _H to FFE6 025C _H	(Prohibited area)												
FFE6 0260 _H	NCRA1					NCNTA1					A1		
FFE6 0264 _H	ICRA1												
FFE6 0268 _H	ICRA21												
FFE6 026C _H to FFE6 027C _H	(Prohibited area)												
FFE6 0280 _H	NCRA2					NCNTA2					A2		
FFE6 0284 _H	ICRA2												
FFE6 0288 _H	ICRA22												
FFE6 028C _H to FFE6 029C _H	(Prohibited area)												
FFE6 02A0 _H	NCRA3					NCNTA3					A3		
FFE6 02A4 _H	ICRA3												
FFE6 02A8 _H	ICRA23												
FFE6 02AC _H to FFE6 02BC _H	(Prohibited area)												

Timer A Registers (FFE6 0200_H to FFE6 03FF_H) (2/2)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 02C0 _H	NCRA4						NCNTA4						A4
FFE6 02C4 _H	ICRA4												
FFE6 02C8 _H	ICRA24												
FFE6 02CC _H to FFE6 02DC _H	(Prohibited area)												
FFE6 02E0 _H	NCRA5						NCNTA5						A5
FFE6 02E4 _H	ICRA5												
FFE6 02E8 _H	ICRA25												
FFE6 02EC _H to FFE6 02FC _H	(Prohibited area)												
FFE6 0300 _H	NCRA6						NCNTA6						A6
FFE6 0304 _H	ICRA6												
FFE6 0308 _H	ICRA26												
FFE6 030C _H to FFE6 031C _H	(Prohibited area)												
FFE6 0320 _H	NCRA7						NCNTA7						A7
FFE6 0324 _H	ICRA7												
FFE6 0328 _H	ICRA27												
FFE6 032C _H to FFE6 03FC _H	(Prohibited area)												

Timer B Registers (FFE6 0400_H to FFE6 05FF_H) (1/3)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 0400 _H	(Prohibited area)			TIORB			(Prohibited area)			TCRB			Common to B
FFE6 0404 _H	TSCRb						TSRB						
FFE6 0408 _H	TIERB						(Prohibited area)			TICRB			
FFE6 040C _H	(Prohibited area)			TCNT4CRB			(Prohibited area)			CHSELBR			
FFE6 0410 _H	(Prohibited area)									TCRBS1			
FFE6 0414 _H	(Prohibited area)									TCRBS2			
FFE6 0418 _H to FFE6 043C _H	(Prohibited area)												
FFE6 0440 _H	TCNTB0S1												
FFE6 0444 _H	ICRB0S1												
FFE6 0448 _H	OCRB0S1												
FFE6 044C _H	(Prohibited area)			OCRB1S1			(Prohibited area)			TCNTB1S1			
FFE6 0450 _H	(Prohibited area)			OCRB11S1			(Prohibited area)			OCRB10S1			
FFE6 0454 _H	(Prohibited area)									OCRB12S1			
FFE6 0458 _H	ICRB1S1												
FFE6 045C _H	ICRB2S1												
FFE6 0460 _H	TCNTB0S2												
FFE6 0464 _H	ICRB0S2												
FFE6 0468 _H	OCRB0S2												
FFE6 046C _H	(Prohibited area)			OCRB1S2			(Prohibited area)			TCNTB1S2			
FFE6 0470 _H	(Prohibited area)			OCRB11S2			(Prohibited area)			OCRB10S2			
FFE6 0474 _H	(Prohibited area)									OCRB12S2			
FFE6 0478 _H	ICRB1S2												
FFE6 047C _H	ICRB2S2												
FFE6 0480 _H	TCNTB0												
FFE6 0484 _H	ICRB0												
FFE6 0488 _H	RECRB1												
FFE6 048C _H	RECRB2												
FFE6 0490 _H	RECRB3												
FFE6 0494 _H	RECRB4												
FFE6 0498 _H	RECRB5												
FFE6 049C _H	RECRB6												
FFE6 04A0 _H	RBURB0												
FFE6 04A4 _H	RBURB1												
FFE6 04A8 _H	RBURB2												
FFE6 04AC _H	RBURB3												
FFE6 04B0 _H	RBURB4												
FFE6 04B4 _H	RBURB5												
FFE6 04B8 _H	RBURB6												
FFE6 04BC _H	OCRB0												

Timer B Registers (FFE6 0400_H to FFE6 05FF_H) (2/3)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 04C0 _H	MIICRB0												
FFE6 04C4 _H	(Prohibited area)			OCRB1			(Prohibited area)			TCNTB1			
FFE6 04C8 _H	(Prohibited area)			OCRB11			(Prohibited area)			OCRB10			
FFE6 04CC _H	(Prohibited area)									OCRB12			
FFE6 04D0 _H	ICRB1												
FFE6 04D4 _H	ICRB2												
FFE6 04D8 _H	ICRB33			ICRB32			ICRB31			ICRB30			
FFE6 04DC _H	ICRB37			ICRB36			ICRB35			ICRB34			
FFE6 04E0 _H	OCRB23			OCRB22			OCRB21			OCRB20			
FFE6 04E4 _H	OCRB27			OCRB26			OCRB25			OCRB24			
FFE6 04E8 _H	OCRB31			OCRB30			OCRB29			OCRB28			
FFE6 04EC _H	OCRB35			OCRB34			OCRB33			OCRB32			
FFE6 04F0 _H	OCRB39			OCRB38			OCRB37			OCRB36			
FFE6 04F4 _H	OCRB43			OCRB42			OCRB41			OCRB40			
FFE6 04F8 _H	(Prohibited area)												
FFE6 04FC _H	(Prohibited area)												
FFE6 0500 _H	LDB												
FFE6 0504 _H	RLDB												
FFE6 0508 _H	TCNTB2												
FFE6 050C _H	PIMR2						PIMR1						
FFE6 0510 _H	TCNTB6												
FFE6 0514 _H	ICRB6												
FFE6 0518 _H	(Prohibited area)									RARB6			
FFE6 051C _H	TCNTB6M												
FFE6 0520 _H	OCRB6												
FFE6 0524 _H	OCRB7												
FFE6 0528 _H	(Prohibited area)												
to													
FFE6 053C _H													
FFE6 0540 _H	(Prohibited area)			TSEQRB			(Prohibited area)			TSEQCRB			
FFE6 0544 _H	(Prohibited area)			TSEQENB2			TSEQENB1			TSEQENB0			
FFE6 0548 _H	TEPCFB1						TEPCFB0						
FFE6 054C _H	TEPCFB3						TEPCFB2						
FFE6 0550 _H	TEPCFB5						TEPCFB4						
FFE6 0554 _H	TEPCFB7						TEPCFB6						
FFE6 0558 _H	TEPCFB9						TEPCFB8						
FFE6 055C _H	TEPCFB11						TEPCFB10						
FFE6 0560 _H	TEPCFB13						TEPCFB12						
FFE6 0564 _H	TEPCFB15						TEPCFB14						
FFE6 0568 _H	TEPCFB17						TEPCFB16						
FFE6 056C _H	TEPCFB19						TEPCFB18						
FFE6 0570 _H	TEPCFB21						TEPCFB20						
FFE6 0574 _H	TEPCFB23						TEPCFB22						

Timer B Registers (FFE6 0400_H to FFE6 05FF_H) (3/3)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 0578 _H	(Prohibited area)		TEPCFENB2			TEPCFENB1			TEPCFENB0				
FFE6 057C _H	(Prohibited area)												
FFE6 0580 _H	TCNTB3												
FFE6 0584 _H	TCNTB4												
FFE6 0588 _H	TCNTB5												
FFE6 058C _H	(Prohibited area)		(Prohibited area)			(Prohibited area)			TCCLFRB				
FFE6 0590 _H	TCCLR8												
FFE6 0594 _H	OCR8												
FFE6 0598 _H	(Prohibited area)												
to													
FFE6 05BC _H													
FFE6 05C0 _H	DICR8												
FFE6 05C4 _H	DRECR1												
FFE6 05C8 _H	TEPCRECR1												
FFE6 05CC _H	TEPCVALR8												
FFE6 05D0 _H	(Prohibited area)												
to													
FFE6 05FC _H													

Timer C Registers (FFE6 0800_H to FFE6 0FFF_H) (1/8)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 0800 _H	(Prohibited area)						TSTRC						Common to C
FFE6 0804 _H	NCMCR2C						NCMCR1C						
FFE6 0808 _H	NCCRC3			NCCRC2			NCCRC1			NCCRC0			
FFE6 080C _H	NCCRC7			NCCRC6			NCCRC5			NCCRC4			
FFE6 0810 _H	(Prohibited area)			NCCRC10			NCCRC9			NCCRC8			
FFE6 0814 _H	(Prohibited area)												
FFE6 0818 _H	(Prohibited area)												
to													
FFE6 0820 _H	TSCRC0						(Prohibited area)						C0
FFE6 0824 _H	TSRC0						TCRC0						
FFE6 0828 _H	(Prohibited area)						TIORC0						
FFE6 082C _H	(Prohibited area)												
FFE6 0830 _H	(Prohibited area)												
FFE6 0834 _H	(Prohibited area)						TIERC0						
FFE6 0838 _H	TCNTC0												
FFE6 083C _H	CUCRC0												
FFE6 0840 _H	GRC00												
FFE6 0844 _H	GRC01												
FFE6 0848 _H	GRC02												
FFE6 084C _H	GRC03												
FFE6 0850 _H	OCRC00												
FFE6 0854 _H	OCRC01												
FFE6 0858 _H	OCRC02												
FFE6 085C _H	OCRC03												
FFE6 0860 _H	(Prohibited area)			RCR2C0			(Prohibited area)			RCR1C0			
to	(Prohibited area)												
FFE6 0870 _H	NCRC00						NCNTC00						
FFE6 0874 _H	NCRC01						NCNTC01						
FFE6 0878 _H	NCRC02						NCNTC02						
FFE6 087C _H	NCRC03						NCNTC03						
FFE6 0880 _H	OCMRC00												
FFE6 0884 _H	GMRC00												
FFE6 0888 _H	OCMRC01												
FFE6 088C _H	GMRC01												
FFE6 0890 _H	OCMRC02												
FFE6 0894 _H	GMRC02												
FFE6 0898 _H	OCMRC03												
FFE6 089C _H	GMRC03												

Timer C Registers (FFE6 0800_H to FFE6 0FFF_H) (2/8)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 08A0 _H	TSCRC1						(Prohibited area)						C1
FFE6 08A4 _H	TSRC1						TCRC1						
FFE6 08A8 _H	(Prohibited area)						TIORC1						
FFE6 08AC _H	(Prohibited area)												
FFE6 08B0 _H	(Prohibited area)												
FFE6 08B4 _H	(Prohibited area)						TIERC1						
FFE6 08B8 _H	TCNTC1												
FFE6 08BC _H	CUCRC1												
FFE6 08C0 _H	GRC10												
FFE6 08C4 _H	GRC11												
FFE6 08C8 _H	GRC12												
FFE6 08CC _H	GRC13												
FFE6 08D0 _H	OCRC10												
FFE6 08D4 _H	OCRC11												
FFE6 08D8 _H	OCRC12												
FFE6 08DC _H	OCRC13												
FFE6 08E0 _H	(Prohibited area)			RCR2C1			(Prohibited area)			RCR1C1			
to	(Prohibited area)												
FFE6 08F0 _H	NCRC10						NCNTC10						
FFE6 08F4 _H	NCRC11						NCNTC11						
FFE6 08F8 _H	NCRC12						NCNTC12						
FFE6 08FC _H	NCRC13						NCNTC13						
FFE6 0900 _H	OCMRC10												
FFE6 0904 _H	GMRC10												
FFE6 0908 _H	OCMRC11												
FFE6 090C _H	GMRC11												
FFE6 0910 _H	OCMRC12												
FFE6 0914 _H	GMRC12												
FFE6 0918 _H	OCMRC13												
FFE6 091C _H	GMRC13												
FFE6 0920 _H	TSCRC2						(Prohibited area)						C2
FFE6 0924 _H	TSRC2						TCRC2						
FFE6 0928 _H	(Prohibited area)						TIORC2						
FFE6 092C _H	(Prohibited area)												
FFE6 0930 _H	(Prohibited area)												
FFE6 0934 _H	(Prohibited area)						TIERC2						
FFE6 0938 _H	TCNTC2												
FFE6 093C _H	CUCRC2												
FFE6 0940 _H	GRC20												
FFE6 0944 _H	GRC21												
FFE6 0948 _H	GRC22												
FFE6 094C _H	GRC23												

Timer C Registers (FFE6 0800_H to FFE6 0FFF_H) (3/8)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 0950 _H	OCRC20												
FFE6 0954 _H	OCRC21												
FFE6 0958 _H	OCRC22												
FFE6 095C _H	OCRC23												
FFE6 0860 _H	(Prohibited area)			RCR2C2			(Prohibited area)			RCR1C2			
to	(Prohibited area)												
FFE6 0970 _H	NCRC20						NCNTC20						
FFE6 0974 _H	NCRC21						NCNTC21						
FFE6 0978 _H	NCRC22						NCNTC22						
FFE6 097C _H	NCRC23						NCNTC23						
FFE6 0980 _H	OCMRC20												
FFE6 0984 _H	GMRC20												
FFE6 0988 _H	OCMRC21												
FFE6 098C _H	GMRC21												
FFE6 0990 _H	OCMRC22												
FFE6 0994 _H	GMRC22												
FFE6 0998 _H	OCMRC23												
FFE6 099C _H	GMRC23												
FFE6 09A0 _H	TSCRC3						(Prohibited area)						C3
FFE6 09A4 _H	TSRC3						TCRC3						
FFE6 09A8 _H	(Prohibited area)						TIORC3						
FFE6 09AC _H	(Prohibited area)												
FFE6 09B0 _H	(Prohibited area)												
FFE6 09B4 _H	(Prohibited area)						TIERC3						
FFE6 09B8 _H	TCNTC3												
FFE6 09BC _H	CUCRC3												
FFE6 09C0 _H	GRC30												
FFE6 09C4 _H	GRC31												
FFE6 09C8 _H	GRC32												
FFE6 09CC _H	GRC33												
FFE6 09D0 _H	OCRC30												
FFE6 09D4 _H	OCRC31												
FFE6 09D8 _H	OCRC32												
FFE6 09DC _H	OCRC33												
FFE6 09E0 _H	(Prohibited area)			RCR2C3			(Prohibited area)			RCR1C3			
to	(Prohibited area)												
FFE6 09F0 _H	NCRC30						NCNTC30						
FFE6 09F4 _H	NCRC31						NCNTC31						
FFE6 09F8 _H	NCRC32						NCNTC32						
FFE6 09FC _H	NCRC33						NCNTC33						
FFE6 0A00 _H	OCMRC30												
FFE6 0A04 _H	GMRC30												
FFE6 0A08 _H	OCMRC31												

Timer C Registers (FFE6 0800_H to FFE6 0FFF_H) (4/8)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 0A0C _H	GMRC31												
FFE6 0A10 _H	OCMRC32												
FFE6 0A14 _H	GMRC32												
FFE6 0A18 _H	OCMRC33												
FFE6 0A1C _H	GMRC33												
FFE6 0A20 _H	TSCRC4						(Prohibited area)						C4
FFE6 0A24 _H	TSRC4						TCRC4						
FFE6 0A28 _H	(Prohibited area)						TIORC4						
FFE6 0A2C _H	(Prohibited area)												
FFE6 0A30 _H	(Prohibited area)												
FFE6 0A34 _H	(Prohibited area)						TIERC4						
FFE6 0A38 _H	TCNTC4												
FFE6 0A3C _H	CUCRC4												
FFE6 0A40 _H	GRC40												
FFE6 0A44 _H	GRC41												
FFE6 0A48 _H	GRC42												
FFE6 0A4C _H	GRC43												
FFE6 0A50 _H	OCRC40												
FFE6 0A54 _H	OCRC41												
FFE6 0A58 _H	OCRC42												
FFE6 0A5C _H	OCRC43												
FFE6 0A60 _H	(Prohibited area)			RCR2C4			(Prohibited area)			RCR1C4			
to	(Prohibited area)												
FFE6 0A70 _H	NCRC40						NCNTC40						
FFE6 0A74 _H	NCRC41						NCNTC41						
FFE6 0A78 _H	NCRC42						NCNTC42						
FFE6 0A7C _H	NCRC43						NCNTC43						
FFE6 0A80 _H	OCMRC40												
FFE6 0A84 _H	GMRC40												
FFE6 0A88 _H	OCMRC41												
FFE6 0A8C _H	GMRC41												
FFE6 0A90 _H	OCMRC42												
FFE6 0A94 _H	GMRC42												
FFE6 0A98 _H	OCMRC43												
FFE6 0A9C _H	GMRC43												
FFE6 0AA0 _H	TSCRC5						(Prohibited area)						C5
FFE6 0AA4 _H	TSRC5						TCRC5						
FFE6 0AA8 _H	(Prohibited area)						TIORC5						
FFE6 0AAC _H	(Prohibited area)												
FFE6 0AB0 _H	(Prohibited area)												
FFE6 0AB4 _H	(Prohibited area)						TIERC5						
FFE6 0AB8 _H	TCNTC5												
FFE6 0ABC _H	CUCRC5												

Timer C Registers (FFE6 0800_H to FFE6 0FFF_H) (5/8)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 0AC0 _H	GRC50												
FFE6 0AC4 _H	GRC51												
FFE6 0AC8 _H	GRC52												
FFE6 0ACC _H	GRC53												
FFE6 0AD0 _H	OCRC50												
FFE6 0AD4 _H	OCRC51												
FFE6 0AD8 _H	OCRC52												
FFE6 0ADC _H	OCRC53												
FFE6 0AE0 _H	(Prohibited area)			RCR2C5			(Prohibited area)			RCR1C5			
to	(Prohibited area)												
FFE6 0AF0 _H	NCRC50						NCNTC50						
FFE6 0AF4 _H	NCRC51						NCNTC51						
FFE6 0AF8 _H	NCRC52						NCNTC52						
FFE6 0AFC _H	NCRC53						NCNTC53						
FFE6 0B00 _H	OCMRC50												
FFE6 0B04 _H	GMRC50												
FFE6 0B08 _H	OCMRC51												
FFE6 0B0C _H	GMRC51												
FFE6 0B10 _H	OCMRC52												
FFE6 0B14 _H	GMRC52												
FFE6 0B18 _H	OCMRC53												
FFE6 0B1C _H	GMRC53												
FFE6 0B20 _H	TSCRC6						(Prohibited area)						C6
FFE6 0B24 _H	TSRC6						TCRC6						
FFE6 0B28 _H	(Prohibited area)						TIORC6						
FFE6 0B2C _H	(Prohibited area)												
FFE6 0B30 _H	(Prohibited area)												
FFE6 0B34 _H	(Prohibited area)						TIERC6						
FFE6 0B38 _H	TCNTC6												
FFE6 0B3C _H	CUCRC6												
FFE6 0B40 _H	GRC60												
FFE6 0B44 _H	GRC61												
FFE6 0B48 _H	GRC62												
FFE6 0B4C _H	GRC63												
FFE6 0B50 _H	OCRC60												
FFE6 0B54 _H	OCRC61												
FFE6 0B58 _H	OCRC62												
FFE6 0B5C _H	OCRC63												
FFE6 0B60 _H	(Prohibited area)			RCR2C6			(Prohibited area)			RCR1C6			
to	(Prohibited area)												
FFE6 0B70 _H	NCRC60						NCNTC60						
FFE6 0B74 _H	NCRC61						NCNTC61						
FFE6 0B78 _H	NCRC62						NCNTC62						
FFE6 0B7C _H	NCRC63						NCNTC63						

Timer C Registers (FFE6 0800_H to FFE6 0FFF_H) (6/8)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 0B80 _H	OCMRC60												
FFE6 0B84 _H	GMRC60												
FFE6 0B88 _H	OCMRC61												
FFE6 0B8C _H	GMRC61												
FFE6 0B90 _H	OCMRC62												
FFE6 0B94 _H	GMRC62												
FFE6 0B98 _H	OCMRC63												
FFE6 0B9C _H	GMRC63												
FFE6 0BA0 _H	TSCRC7						(Prohibited area)						C7
FFE6 0BA4 _H	TSRC7						TCRC7						
FFE6 0BA8 _H	(Prohibited area)						TIORC7						
FFE6 0BAC _H	(Prohibited area)												
FFE6 0BB0 _H	(Prohibited area)												
FFE6 0BB4 _H	(Prohibited area)						TIERC7						
FFE6 0BB8 _H	TCNTC7												
FFE6 0BBC _H	CUCRC7												
FFE6 0BC0 _H	GRC70												
FFE6 0BC4 _H	GRC71												
FFE6 0BC8 _H	GRC72												
FFE6 0BCC _H	GRC73												
FFE6 0BD0 _H	OCRC70												
FFE6 0BD4 _H	OCRC71												
FFE6 0BD8 _H	OCRC72												
FFE6 0BDC _H	OCRC73												
FFE6 0BE0 _H	(Prohibited area)			RCR2C7			(Prohibited area)			RCR1C7			
to	(Prohibited area)												
FFE6 0BF0 _H	NCRC70						NCNTC70						
FFE6 0BF4 _H	NCRC71						NCNTC71						
FFE6 0BF8 _H	NCRC72						NCNTC72						
FFE6 0BFC _H	NCRC73						NCNTC73						
FFE6 0C00 _H	OCMRC70												
FFE6 0C04 _H	GMRC70												
FFE6 0C08 _H	OCMRC71												
FFE6 0C0C _H	GMRC71												
FFE6 0C10 _H	OCMRC72												
FFE6 0C14 _H	GMRC72												
FFE6 0C18 _H	OCMRC73												
FFE6 0C1C _H	GMRC73												
FFE6 0C20 _H	TSCRC8						(Prohibited area)						C8
FFE6 0C24 _H	TSRC8						TCRC8						
FFE6 0C28 _H	(Prohibited area)						TIORC8						
FFE6 0C2C _H	(Prohibited area)												
FFE6 0C30 _H	(Prohibited area)												
FFE6 0C34 _H	(Prohibited area)						TIERC8						

Timer C Registers (FFE6 0800_H to FFE6 0FFF_H) (7/8)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 0C38 _H	TCNTC8												
FFE6 0C3C _H	CUCRC8												
FFE6 0C40 _H	GRC80												
FFE6 0C44 _H	GRC81												
FFE6 0C48 _H	GRC82												
FFE6 0C4C _H	GRC83												
FFE6 0C50 _H	OCRC80												
FFE6 0C54 _H	OCRC81												
FFE6 0C58 _H	OCRC82												
FFE6 0C5C _H	OCRC83												
FFE6 0C60 _H	(Prohibited area)		RCR2C8				(Prohibited area)			RCR1C8			
to	(Prohibited area)												
FFE6 0C70 _H	NCRC80					NCNTC80							
FFE6 0C74 _H	NCRC81					NCNTC81							
FFE6 0C78 _H	NCRC82					NCNTC82							
FFE6 0C7C _H	NCRC83					NCNTC83							
FFE6 0C80 _H	OCMRC80												
FFE6 0C84 _H	GMRC80												
FFE6 0C88 _H	OCMRC81												
FFE6 0C8C _H	GMRC81												
FFE6 0C90 _H	OCMRC82												
FFE6 0C94 _H	GMRC82												
FFE6 0C98 _H	OCMRC83												
FFE6 0C9C _H	GMRC83												
FFE6 0CA0 _H	TSCRC9					(Prohibited area)							C9
FFE6 0CA4 _H	TSRC9					TCRC9							
FFE6 0CA8 _H	(Prohibited area)					TIORC9							
FFE6 0CAC _H	(Prohibited area)												
FFE6 0CB0 _H	(Prohibited area)												
FFE6 0CB4 _H	(Prohibited area)					TIERC9							
FFE6 0CB8 _H	TCNTC9												
FFE6 0CBC _H	CUCRC9												
FFE6 0CC0 _H	GRC90												
FFE6 0CC4 _H	GRC91												
FFE6 0CC8 _H	GRC92												
FFE6 0CCC _H	GRC93												
FFE6 0CD0 _H	OCRC90												
FFE6 0CD4 _H	OCRC91												
FFE6 0CD8 _H	OCRC92												
FFE6 0CDC _H	OCRC93												
FFE6 0CE0 _H	(Prohibited area)		RCR2C9				(Prohibited area)			RCR1C9			
to	(Prohibited area)												
FFE6 0CF0 _H	NCRC90					NCNTC90							
FFE6 0CF4 _H	NCRC91					NCNTC91							

Timer C Registers (FFE6 0800_H to FFE6 0FFF_H) (8/8)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 0CF8 _H	NCRC92						NCNTC92						
FFE6 0CFC _H	NCRC93						NCNTC93						
FFE6 0D00 _H	OCMRC90												
FFE6 0D04 _H	GMRC90												
FFE6 0D08 _H	OCMRC91												
FFE6 0D0C _H	GMRC91												
FFE6 0D10 _H	OCMRC92												
FFE6 0D14 _H	GMRC92												
FFE6 0D18 _H	OCMRC93												
FFE6 0D1C _H	GMRC93												
FFE6 0D20 _H	TSCRC10						(Prohibited area)						C10
FFE6 0D24 _H	TSRC10						TCRC10						
FFE6 0D28 _H	(Prohibited area)						TIORC10						
FFE6 0D2C _H	(Prohibited area)												
FFE6 0D30 _H	(Prohibited area)												
FFE6 0D34 _H	(Prohibited area)						TIERC10						
FFE6 0D38 _H	TCNTC10												
FFE6 0D3C _H	CUCRC10												
FFE6 0D40 _H	GRC100												
FFE6 0D44 _H	GRC101												
FFE6 0D48 _H	GRC102												
FFE6 0D4C _H	GRC103												
FFE6 0D50 _H	OCRC100												
FFE6 0D54 _H	OCRC101												
FFE6 0D58 _H	OCRC102												
FFE6 0D5C _H	OCRC103												
FFE6 0D60 _H	(Prohibited area)			RCR2C10			(Prohibited area)			RCR1C10			
to	(Prohibited area)												
FFE6 0D70 _H	NCRC100						NCNTC100						
FFE6 0D74 _H	NCRC101						NCNTC101						
FFE6 0D78 _H	NCRC102						NCNTC102						
FFE6 0D7C _H	NCRC103						NCNTC103						
FFE6 0D80 _H	OCMRC100												
FFE6 0D84 _H	GMRC100												
FFE6 0D88 _H	OCMRC101												
FFE6 0D8C _H	GMRC101												
FFE6 0D90 _H	OCMRC102												
FFE6 0D94 _H	GMRC102												
FFE6 0D98 _H	OCMRC103												
FFE6 0D9C _H	GMRC103												
FFE6 0DA0 _H	(Prohibited area)												
to													
FFE6 0FFC _H													

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (1/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 2000 _H	(Prohibited area)						TSTRD						Common to D
FFE6 2004 _H	(Prohibited area)									CCAPSELD			
FFE6 2008 _H to FFE6 201C _H	(Prohibited area)												
FFE6 2020 _H	CICRD0												
FFE6 2024 _H	CICRD1												
FFE6 2028 _H	CICRD2												
FFE6 202C _H	CICRD3												
FFE6 2030 _H	CICRD4												
FFE6 2034 _H	CICRD5												
FFE6 2038 _H	CICRD6												
FFE6 203C _H	CICRD7												
FFE6 2040 _H to FFE6 21FC _H	(Prohibited area)												
FFE6 2200 _H	DCRD0						TCRD0						D0
FFE6 2204 _H	TIOR2D0						TIOR1D0						
FFE6 2208 _H	DSR2D0			DSR1D0			DSCRD0			DSTRD0			
FFE6 220C _H	TSCRD0						TSRD0						
FFE6 2210 _H	(Prohibited area)			ODRD0			OSELRD0			TOCRD0			
FFE6 2214 _H	MIGSELD0			MIGCRD0			(Prohibited area)			TICTSELDO			
FFE6 2218 _H	OSBRD0												
FFE6 221C _H	(Prohibited area)						TCCRLRD0						
FFE6 2220 _H	TCNT1D0												
FFE6 2224 _H	TCNT2D0												
FFE6 2228 _H	CUCR1D0												
FFE6 222C _H	CUCR2D0												
FFE6 2230 _H	(Prohibited area)									RCR1D0			
FFE6 2234 _H	(Prohibited area)									RCR2D0			
FFE6 2238 _H	(Prohibited area)												
FFE6 223C _H	TIORD0						TCMPED0			TOCCR0			
FFE6 2240 _H	OCR1D00												
FFE6 2244 _H	(Prohibited area)												
FFE6 2248 _H	OCR2D00												
FFE6 224C _H	(Prohibited area)												
FFE6 2250 _H	ICR1D00												
FFE6 2254 _H	ICR2D00												
FFE6 2258 _H	DCNTD00												
FFE6 225C _H	(Prohibited area)												
FFE6 2260 _H	OCR1D01												
FFE6 2264 _H	(Prohibited area)												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (2/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 2268 _H	OCR2D01												
FFE6 226C _H	(Prohibited area)												
FFE6 2270 _H	ICR1D01												
FFE6 2274 _H	ICR2D01												
FFE6 2278 _H	DCNTD01												
FFE6 227C _H	(Prohibited area)												
FFE6 2280 _H	OCR1D02												
FFE6 2284 _H	(Prohibited area)												
FFE6 2288 _H	OCR2D02												
FFE6 228C _H	(Prohibited area)												
FFE6 2290 _H	ICR1D02												
FFE6 2294 _H	ICR2D02												
FFE6 2298 _H	DCNTD02												
FFE6 229C _H	(Prohibited area)												
FFE6 22A0 _H	OCR1D03												
FFE6 22A4 _H	(Prohibited area)												
FFE6 22A8 _H	OCR2D03												
FFE6 22AC _H	(Prohibited area)												
FFE6 22B0 _H	ICR1D03												
FFE6 22B4 _H	ICR2D03												
FFE6 22B8 _H	DCNTD03												
FFE6 22BC _H	(Prohibited area)												
FFE6 22C0 _H	(Prohibited area)									TIER1D0			
FFE6 22C4 _H	TIER2D0												
FFE6 22C8 _H	(Prohibited area)						TIER3D0						
FFE6 22CC _H	(Prohibited area)												
FFE6 22D0 _H	(Prohibited area)						TSR2D0						
FFE6 22D4 _H	(Prohibited area)						TSCR2D0						
FFE6 22D8 _H	(Prohibited area)												
to													
FFE6 22FC _H													
FFE6 2300 _H	OFMICNTD00												
FFE6 2304 _H	ONMICNTD00												
FFE6 2308 _H	OTOMICNTD00												
FFE6 230C _H	(Prohibited area)												
FFE6 2310 _H	OFMIND00												
FFE6 2314 _H	ONMIND00												
FFE6 2318 _H	OTOMIND00												
FFE6 231C _H	(Prohibited area)												
FFE6 2320 _H	ONCAP1D00												
FFE6 2324 _H	ONCAP2D00(ICR2D00)												
FFE6 2328 _H	(Prohibited area)												
FFE6 232C _H	(Prohibited area)												
FFE6 2330 _H	OFCAP1D00(ICR1D00)												
FFE6 2334 _H	OFCAP2D00												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (3/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 2338 _H	(Prohibited area)												
FFE6 233C _H	(Prohibited area)												
FFE6 2340 _H	OFMICNTD01												
FFE6 2344 _H	ONMICNTD01												
FFE6 2348 _H	OTOMICNTD01												
FFE6 234C _H	(Prohibited area)												
FFE6 2350 _H	OFMIND01												
FFE6 2354 _H	ONMIND01												
FFE6 2358 _H	OTOMIND01												
FFE6 235C _H	(Prohibited area)												
FFE6 2360 _H	ONCAP1D01												
FFE6 2364 _H	ONCAP2D01(ICR2D01)												
FFE6 2368 _H	(Prohibited area)												
FFE6 236C _H	(Prohibited area)												
FFE6 2370 _H	OFCAP1D01(ICR1D01)												
FFE6 2374 _H	OFCAP2D01												
FFE6 2378 _H	(Prohibited area)												
FFE6 237C _H	(Prohibited area)												
FFE6 2380 _H	OFMICNTD02												
FFE6 2384 _H	ONMICNTD02												
FFE6 2388 _H	OTOMICNTD02												
FFE6 238C _H	(Prohibited area)												
FFE6 2390 _H	OFMIND02												
FFE6 2394 _H	ONMIND02												
FFE6 2398 _H	OTOMIND02												
FFE6 239C _H	(Prohibited area)												
FFE6 23A0 _H	ONCAP1D02												
FFE6 23A4 _H	ONCAP2D02(ICR2D02)												
FFE6 23A8 _H	(Prohibited area)												
FFE6 23AC _H	(Prohibited area)												
FFE6 23B0 _H	OFCAP1D02(ICR1D02)												
FFE6 23B4 _H	OFCAP2D02												
FFE6 23B8 _H	(Prohibited area)												
FFE6 23BC _H	(Prohibited area)												
FFE6 23C0 _H	OFMICNTD03												
FFE6 23C4 _H	ONMICNTD03												
FFE6 23C8 _H	OTOMICNTD03												
FFE6 23CC _H	(Prohibited area)												
FFE6 23D0 _H	OFMIND03												
FFE6 23D4 _H	ONMIND03												
FFE6 23D8 _H	OTOMIND03												
FFE6 23DC _H	(Prohibited area)												
FFE6 23E0 _H	ONCAP1D03												
FFE6 23E4 _H	ONCAP2D03(ICR2D03)												
FFE6 23E8 _H	(Prohibited area)												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (4/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 23EC _H	(Prohibited area)												
FFE6 23F0 _H	OFCAP1D03(ICR1D03)												
FFE6 23F4 _H	OFCAP2D03												
FFE6 23F8 _H	(Prohibited area)												
FFE6 23FC _H	(Prohibited area)												
FFE6 2400 _H	DCRD1						TCRD1						D1
FFE6 2404 _H	TIOR2D1						TIOR1D1						
FFE6 2408 _H	DSR2D1			DSR1D1			DSCRD1			DSTRD1			
FFE6 240C _H	TSCRD1						TSRD1						
FFE6 2410 _H	(Prohibited area)			ODRD1			OSELRD1			TOCRD1			
FFE6 2414 _H	MIGSELD1			MIGCRD1			(Prohibited area)			TICTSELD1			
FFE6 2418 _H	OSBRD1												
FFE6 241C _H	(Prohibited area)						TCCRLRD1						
FFE6 2420 _H	TCNT1D1												
FFE6 2424 _H	TCNT2D1												
FFE6 2428 _H	CUCR1D1												
FFE6 242C _H	CUCR2D1												
FFE6 2430 _H	(Prohibited area)									RCR1D1			
FFE6 2434 _H	(Prohibited area)									RCR2D1			
FFE6 2438 _H	(Prohibited area)												
FFE6 243C _H	TIORD1						TCMPED1			TOCCRD1			
FFE6 2440 _H	OCR1D10												
FFE6 2444 _H	(Prohibited area)												
FFE6 2448 _H	OCR2D10												
FFE6 244C _H	(Prohibited area)												
FFE6 2450 _H	ICR1D10												
FFE6 2454 _H	ICR2D10												
FFE6 2458 _H	DCNTD10												
FFE6 245C _H	(Prohibited area)												
FFE6 2460 _H	OCR1D11												
FFE6 2464 _H	(Prohibited area)												
FFE6 2468 _H	OCR2D11												
FFE6 246C _H	(Prohibited area)												
FFE6 2470 _H	ICR1D11												
FFE6 2474 _H	ICR2D11												
FFE6 2478 _H	DCNTD11												
FFE6 247C _H	(Prohibited area)												
FFE6 2480 _H	OCR1D12												
FFE6 2484 _H	(Prohibited area)												
FFE6 2488 _H	OCR2D12												
FFE6 248C _H	(Prohibited area)												
FFE6 2490 _H	ICR1D12												
FFE6 2494 _H	ICR2D12												
FFE6 2498 _H	DCNTD12												
FFE6 249C _H	(Prohibited area)												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (5/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 24A0 _H	OCR1D13												
FFE6 24A4 _H	(Prohibited area)												
FFE6 24A8 _H	OCR2D13												
FFE6 24AC _H	(Prohibited area)												
FFE6 24B0 _H	ICR1D13												
FFE6 24B4 _H	ICR2D13												
FFE6 24B8 _H	DCNTD13												
FFE6 24BC _H	(Prohibited area)												
FFE6 24C0 _H	(Prohibited area)									TIER1D1			
FFE6 24C4 _H	TIER2D1												
FFE6 24C8 _H	(Prohibited area)						TIER3D1						
FFE6 24CC _H	(Prohibited area)												
FFE6 24D0 _H	(Prohibited area)						TSR2D0						
FFE6 24D4 _H	(Prohibited area)						TSCR2D0						
FFE6 24D8 _H	(Prohibited area)												
to													
FFE6 24FC _H													
FFE6 2500 _H	OFMICNTD10												
FFE6 2504 _H	ONMICNTD10												
FFE6 2508 _H	OTOMICNTD10												
FFE6 250C _H	(Prohibited area)												
FFE6 2510 _H	OFMIND10												
FFE6 2514 _H	ONMIND10												
FFE6 2518 _H	OTOMIND10												
FFE6 251C _H	(Prohibited area)												
FFE6 2520 _H	ONCAP1D10												
FFE6 2524 _H	ONCAP2D10(ICR2D10)												
FFE6 2528 _H	(Prohibited area)												
FFE6 252C _H	(Prohibited area)												
FFE6 2530 _H	OFCAP1D10(ICR1D10)												
FFE6 2534 _H	OFCAP2D10												
FFE6 2538 _H	(Prohibited area)												
FFE6 253C _H	(Prohibited area)												
FFE6 2540 _H	OFMICNTD11												
FFE6 2544 _H	ONMICNTD11												
FFE6 2548 _H	OTOMICNTD11												
FFE6 254C _H	(Prohibited area)												
FFE6 2550 _H	OFMIND11												
FFE6 2554 _H	ONMIND11												
FFE6 2558 _H	OTOMIND11												
FFE6 255C _H	(Prohibited area)												
FFE6 2560 _H	ONCAP1D11												
FFE6 2564 _H	ONCAP2D11(ICR2D11)												
FFE6 2568 _H	(Prohibited area)												
FFE6 256C _H	(Prohibited area)												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (6/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 2570 _H	OFCAP1D11(ICR1D11)												
FFE6 2574 _H	OFCAP2D11												
FFE6 2578 _H	(Prohibited area)												
FFE6 257C _H	(Prohibited area)												
FFE6 2580 _H	OFMICNTD12												
FFE6 2584 _H	ONMICNTD12												
FFE6 2588 _H	OTOMICNTD12												
FFE6 258C _H	(Prohibited area)												
FFE6 2590 _H	OFMIND12												
FFE6 2594 _H	ONMIND12												
FFE6 2598 _H	OTOMIND12												
FFE6 259C _H	(Prohibited area)												
FFE6 25A0 _H	ONCAP1D12												
FFE6 25A4 _H	ONCAP2D12(ICR2D12)												
FFE6 25A8 _H	(Prohibited area)												
FFE6 25AC _H	(Prohibited area)												
FFE6 25B0 _H	OFCAP1D12(ICR1D12)												
FFE6 25B4 _H	OFCAP2D12												
FFE6 25B8 _H	(Prohibited area)												
FFE6 25BC _H	(Prohibited area)												
FFE6 25C0 _H	OFMICNTD13												
FFE6 25C4 _H	ONMICNTD13												
FFE6 25C8 _H	OTOMICNTD13												
FFE6 25CC _H	(Prohibited area)												
FFE6 25D0 _H	OFMIND13												
FFE6 25D4 _H	ONMIND13												
FFE6 25D8 _H	OTOMIND13												
FFE6 25DC _H	(Prohibited area)												
FFE6 25E0 _H	ONCAP1D13												
FFE6 25E4 _H	ONCAP2D13(ICR2D13)												
FFE6 25E8 _H	(Prohibited area)												
FFE6 25EC _H	(Prohibited area)												
FFE6 25F0 _H	OFCAP1D13(ICR1D13)												
FFE6 25F4 _H	OFCAP2D13												
FFE6 25F8 _H	(Prohibited area)												
FFE6 25FC _H	(Prohibited area)												
FFE6 2600 _H	DCRD2						TCRD2						D2
FFE6 2604 _H	TIOR2D2						TIOR1D2						
FFE6 2608 _H	DSR2D2			DSR1D2			DSCRD2			DSTRD2			
FFE6 260C _H	TSCRD2						TSRD2						
FFE6 2610 _H	(Prohibited area)			ODRD2			OSELRD2			TOCRD2			
FFE6 2614 _H	MIGSELD2			MIGCRD2			(Prohibited area)			TICTSELD2			
FFE6 2618 _H	OSBRD2												
FFE6 261C _H	(Prohibited area)						TCCLRD2						
FFE6 2620 _H	TCNT1D2												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (7/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page	
FFE6 2624 _H	TCNT2D2													
FFE6 2628 _H	CUCR1D2													
FFE6 262C _H	CUCR2D2													
FFE6 2630 _H	(Prohibited area)									RCR1D2				
FFE6 2634 _H	(Prohibited area)									RCR2D2				
FFE6 2638 _H	(Prohibited area)													
FFE6 263C _H	TIORD2					TCMPED2				TOCCRD2				
FFE6 2640 _H	OCR1D20													
FFE6 2644 _H	(Prohibited area)													
FFE6 2648 _H	OCR2D20													
FFE6 264C _H	(Prohibited area)													
FFE6 2650 _H	ICR1D20													
FFE6 2654 _H	ICR2D20													
FFE6 2658 _H	DCNTD20													
FFE6 265C _H	(Prohibited area)													
FFE6 2660 _H	OCR1D21													
FFE6 2664 _H	(Prohibited area)													
FFE6 2668 _H	OCR2D21													
FFE6 266C _H	(Prohibited area)													
FFE6 2670 _H	ICR1D21													
FFE6 2674 _H	ICR2D21													
FFE6 2678 _H	DCNTD21													
FFE6 267C _H	(Prohibited area)													
FFE6 2680 _H	OCR1D22													
FFE6 2684 _H	(Prohibited area)													
FFE6 2688 _H	OCR2D22													
FFE6 268C _H	(Prohibited area)													
FFE6 2690 _H	ICR1D22													
FFE6 2694 _H	ICR2D22													
FFE6 2698 _H	DCNTD22													
FFE6 269C _H	(Prohibited area)													
FFE6 26A0 _H	OCR1D23													
FFE6 26A4 _H	(Prohibited area)													
FFE6 26A8 _H	OCR2D23													
FFE6 26AC _H	(Prohibited area)													
FFE6 26B0 _H	ICR1D23													
FFE6 26B4 _H	ICR2D23													
FFE6 26B8 _H	DCNTD23													
FFE6 26BC _H	(Prohibited area)													
FFE6 26C0 _H	(Prohibited area)									TIER1D2				
FFE6 26C4 _H	TIER2D2													
FFE6 26C8 _H	(Prohibited area)						TIER3D2							
FFE6 26CC _H	(Prohibited area)													
FFE6 26D0 _H	(Prohibited area)						TSR2D2							
FFE6 26D4 _H	(Prohibited area)						TSCR2D2							

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (8/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 26D8 _H	(Prohibited area)												
to													
FFE6 26FC _H													
FFE6 2700 _H	OFMICNTD20												
FFE6 2704 _H	ONMICNTD20												
FFE6 2708 _H	OTOMICNTD20												
FFE6 270C _H	(Prohibited area)												
FFE6 2710 _H	OFMIND20												
FFE6 2714 _H	ONMIND20												
FFE6 2718 _H	OTOMIND20												
FFE6 271C _H	(Prohibited area)												
FFE6 2720 _H	ONCAP1D20												
FFE6 2724 _H	ONCAP2D20(ICR2D20)												
FFE6 2728 _H	(Prohibited area)												
FFE6 272C _H	(Prohibited area)												
FFE6 2730 _H	OFCAP1D20(ICR1D20)												
FFE6 2734 _H	OFCAP2D20												
FFE6 2738 _H	(Prohibited area)												
FFE6 273C _H	(Prohibited area)												
FFE6 2740 _H	OFMICNTD21												
FFE6 2744 _H	ONMICNTD21												
FFE6 2748 _H	OTOMICNTD21												
FFE6 274C _H	(Prohibited area)												
FFE6 2750 _H	OFMIND21												
FFE6 2754 _H	ONMIND21												
FFE6 2758 _H	OTOMIND21												
FFE6 275C _H	(Prohibited area)												
FFE6 2760 _H	ONCAP1D21												
FFE6 2764 _H	ONCAP2D21(ICR2D21)												
FFE6 2768 _H	(Prohibited area)												
FFE6 276C _H	(Prohibited area)												
FFE6 2770 _H	OFCAP1D21(ICR1D21)												
FFE6 2774 _H	OFCAP2D21												
FFE6 2778 _H	(Prohibited area)												
FFE6 277C _H	(Prohibited area)												
FFE6 2780 _H	OFMICNTD22												
FFE6 2784 _H	ONMICNTD22												
FFE6 2788 _H	OTOMICNTD22												
FFE6 278C _H	(Prohibited area)												
FFE6 2790 _H	OFMIND22												
FFE6 2794 _H	ONMIND22												
FFE6 2798 _H	OTOMIND22												
FFE6 279C _H	(Prohibited area)												
FFE6 27A0 _H	ONCAP1D22												
FFE6 27A4 _H	ONCAP2D22(ICR2D22)												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (9/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 27A8 _H	(Prohibited area)												
FFE6 27AC _H	(Prohibited area)												
FFE6 27B0 _H	OFCAP1D22(ICR1D22)												
FFE6 27B4 _H	OFCAP2D22												
FFE6 27B8 _H	(Prohibited area)												
FFE6 27BC _H	(Prohibited area)												
FFE6 27C0 _H	OFMICNTD23												
FFE6 27C4 _H	ONMICNTD23												
FFE6 27C8 _H	OTOMICNTD23												
FFE6 27CC _H	(Prohibited area)												
FFE6 27D0 _H	OFMIND23												
FFE6 27D4 _H	ONMIND23												
FFE6 27D8 _H	OTOMIND23												
FFE6 27DC _H	(Prohibited area)												
FFE6 27E0 _H	ONCAP1D23												
FFE6 27E4 _H	ONCAP2D23(ICR2D23)												
FFE6 27E8 _H	(Prohibited area)												
FFE6 27EC _H	(Prohibited area)												
FFE6 27F0 _H	OFCAP1D23(ICR1D23)												
FFE6 27F4 _H	OFCAP2D23												
FFE6 27F8 _H	(Prohibited area)												
FFE6 27FC _H	(Prohibited area)												
FFE6 2800 _H	DCRD3						TCRD3						D3
FFE6 2804 _H	TIOR2D3						TIOR1D3						
FFE6 2808 _H	DSR2D3			DSR1D3			DSCRD3			DSTRD3			
FFE6 280C _H	TSCRD3						TSRD3						
FFE6 2810 _H	(Prohibited area)			ODRD3			OSELRD3			TOCRD3			
FFE6 2814 _H	MIGSELD3			MIGCRD3			(Prohibited area)			TICTSELD3			
FFE6 2818 _H	OSBRD3												
FFE6 281C _H	(Prohibited area)						TCCRLRD3						
FFE6 2820 _H	TCNT1D3												
FFE6 2824 _H	TCNT2D3												
FFE6 2828 _H	CUCR1D3												
FFE6 282C _H	CUCR2D3												
FFE6 2830 _H	(Prohibited area)									RCR1D3			
FFE6 2834 _H	(Prohibited area)									RCR2D3			
FFE6 2838 _H	(Prohibited area)												
FFE6 283C _H	TIORD3						TCMPED3			TOCCRD3			
FFE6 2840 _H	OCR1D30												
FFE6 2844 _H	(Prohibited area)												
FFE6 2848 _H	OCR2D30												
FFE6 284C _H	(Prohibited area)												
FFE6 2850 _H	ICR1D30												
FFE6 2854 _H	ICR2D30												
FFE6 2858 _H	DCNTD30												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (10/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 285C _H	(Prohibited area)												
FFE6 2860 _H	OCR1D31												
FFE6 2864 _H	(Prohibited area)												
FFE6 2868 _H	OCR2D31												
FFE6 286C _H	(Prohibited area)												
FFE6 2870 _H	ICR1D31												
FFE6 2874 _H	ICR2D31												
FFE6 2878 _H	DCNTD31												
FFE6 287C _H	(Prohibited area)												
FFE6 2880 _H	OCR1D32												
FFE6 2884 _H	(Prohibited area)												
FFE6 2888 _H	OCR2D32												
FFE6 288C _H	(Prohibited area)												
FFE6 2890 _H	ICR1D32												
FFE6 2894 _H	ICR2D32												
FFE6 2898 _H	DCNTD32												
FFE6 289C _H	(Prohibited area)												
FFE6 28A0 _H	OCR1D33												
FFE6 28A4 _H	(Prohibited area)												
FFE6 28A8 _H	OCR2D33												
FFE6 28AC _H	(Prohibited area)												
FFE6 28B0 _H	ICR1D33												
FFE6 28B4 _H	ICR2D33												
FFE6 28B8 _H	DCNTD33												
FFE6 28BC _H	(Prohibited area)												
FFE6 28C0 _H	(Prohibited area)									TIER1D3			
FFE6 28C4 _H	TIER2D3												
FFE6 28C8 _H	(Prohibited area)						TIER3D3						
FFE6 28CC _H	(Prohibited area)												
FFE6 28D0 _H	(Prohibited area)						TSR2D3						
FFE6 28D4 _H	(Prohibited area)						TSCR2D3						
FFE6 28D8 _H	(Prohibited area)												
to													
FFE6 28FC _H													
FFE6 2900 _H	OFMICNTD30												
FFE6 2904 _H	ONMICNTD30												
FFE6 2908 _H	OTOMICNTD30												
FFE6 290C _H	(Prohibited area)												
FFE6 2910 _H	OFMIND30												
FFE6 2914 _H	ONMIND30												
FFE6 2918 _H	OTOMIND30												
FFE6 291C _H	(Prohibited area)												
FFE6 2920 _H	ONCAP1D30												
FFE6 2924 _H	ONCAP2D30(ICR2D30)												
FFE6 2928 _H	(Prohibited area)												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (11/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 292C _H	(Prohibited area)												
FFE6 2930 _H	OFCAP1D30(ICR1D30)												
FFE6 2934 _H	OFCAP2D30												
FFE6 2938 _H	(Prohibited area)												
FFE6 293C _H	(Prohibited area)												
FFE6 2940 _H	OFMICNTD31												
FFE6 2944 _H	ONMICNTD31												
FFE6 2948 _H	OTOMICNTD31												
FFE6 294C _H	(Prohibited area)												
FFE6 2950 _H	OFMIND31												
FFE6 2954 _H	ONMIND31												
FFE6 2958 _H	OTOMIND31												
FFE6 295C _H	(Prohibited area)												
FFE6 2960 _H	ONCAP1D31												
FFE6 2964 _H	ONCAP2D31(ICR2D31)												
FFE6 2968 _H	(Prohibited area)												
FFE6 296C _H	(Prohibited area)												
FFE6 2970 _H	OFCAP1D31(ICR1D31)												
FFE6 2974 _H	OFCAP2D31												
FFE6 2978 _H	(Prohibited area)												
FFE6 297C _H	(Prohibited area)												
FFE6 2980 _H	OFMICNTD32												
FFE6 2984 _H	ONMICNTD32												
FFE6 2988 _H	OTOMICNTD32												
FFE6 298C _H	(Prohibited area)												
FFE6 2990 _H	OFMIND32												
FFE6 2994 _H	ONMIND32												
FFE6 2998 _H	OTOMIND32												
FFE6 299C _H	(Prohibited area)												
FFE6 29A0 _H	ONCAP1D32												
FFE6 29A4 _H	ONCAP2D32(ICR2D32)												
FFE6 29A8 _H	(Prohibited area)												
FFE6 29AC _H	(Prohibited area)												
FFE6 29B0 _H	OFCAP1D32(ICR1D32)												
FFE6 29B4 _H	OFCAP2D32												
FFE6 29B8 _H	(Prohibited area)												
FFE6 29BC _H	(Prohibited area)												
FFE6 29C0 _H	OFMICNTD33												
FFE6 29C4 _H	ONMICNTD33												
FFE6 29C8 _H	OTOMICNTD33												
FFE6 29CC _H	(Prohibited area)												
FFE6 29D0 _H	OFMIND33												
FFE6 29D4 _H	ONMIND33												
FFE6 29D8 _H	OTOMIND33												
FFE6 29DC _H	(Prohibited area)												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (12/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 29E0 _H	ONCAP1D33												
FFE6 29E4 _H	ONCAP2D33(ICR2D33)												
FFE6 29E8 _H	(Prohibited area)												
FFE6 29EC _H	(Prohibited area)												
FFE6 29F0 _H	OFCAP1D33(ICR1D33)												
FFE6 29F4 _H	OFCAP2D33												
FFE6 29F8 _H	(Prohibited area)												
FFE6 29FC _H	(Prohibited area)												
FFE6 2A00 _H	DCRD4						TCRD4						D4
FFE6 2A04 _H	TIOR2D4						TIOR1D4						
FFE6 2A08 _H	DSR2D4			DSR1D4			DSCRD4			DSTRD4			
FFE6 2A0C _H	TSCRD4						TSRD4						
FFE6 2A10 _H	(Prohibited area)			ODRD4			OSELRD4			TOCRD4			
FFE6 2A14 _H	MIGSELD4			MIGCRD4			(Prohibited area)			TICTSELD4			
FFE6 2A18 _H	OSBRD4												
FFE6 2A1C _H	(Prohibited area)						TCCRLRD4						
FFE6 2A20 _H	TCNT1D4												
FFE6 2A24 _H	TCNT2D4												
FFE6 2A28 _H	CUCR1D4												
FFE6 2A2C _H	CUCR2D4												
FFE6 2A30 _H	(Prohibited area)									RCR1D4			
FFE6 2A34 _H	(Prohibited area)									RCR2D4			
FFE6 2A38 _H	(Prohibited area)												
FFE6 2A3C _H	TIORD4						TCMPED4			TOCCRD4			
FFE6 2A40 _H	OCR1D40												
FFE6 2A44 _H	(Prohibited area)												
FFE6 2A48 _H	OCR2D40												
FFE6 2A4C _H	(Prohibited area)												
FFE6 2A50 _H	ICR1D40												
FFE6 2A54 _H	ICR2D40												
FFE6 2A58 _H	DCNTD40												
FFE6 2A5C _H	(Prohibited area)												
FFE6 2A60 _H	OCR1D41												
FFE6 2A64 _H	(Prohibited area)												
FFE6 2A68 _H	OCR2D41												
FFE6 2A6C _H	(Prohibited area)												
FFE6 2A70 _H	ICR1D41												
FFE6 2A74 _H	ICR2D41												
FFE6 2A78 _H	DCNTD41												
FFE6 2A7C _H	(Prohibited area)												
FFE6 2A80 _H	OCR1D42												
FFE6 2A84 _H	(Prohibited area)												
FFE6 2A88 _H	OCR2D42												
FFE6 2A8C _H	(Prohibited area)												
FFE6 2A90 _H	ICR1D42												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (13/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page	
FFE6 2A94 _H	ICR2D42													
FFE6 2A98 _H	DCNTD42													
FFE6 2A9C _H	(Prohibited area)													
FFE6 2AA0 _H	OCR1D43													
FFE6 2AA4 _H	(Prohibited area)													
FFE6 2AA8 _H	OCR2D43													
FFE6 2AAC _H	(Prohibited area)													
FFE6 2AB0 _H	ICR1D43													
FFE6 2AB4 _H	ICR2D43													
FFE6 2AB8 _H	DCNTD43													
FFE6 2ABC _H	(Prohibited area)													
FFE6 2AC0 _H	(Prohibited area)									TIER1D4				
FFE6 2AC4 _H	TIER2D4													
FFE6 2AC8 _H	(Prohibited area)						TIER3D4							
FFE6 2ACC _H	(Prohibited area)													
FFE6 2AD0 _H	(Prohibited area)						TSR2D4							
FFE6 2AD4 _H	(Prohibited area)						TSCR2D4							
FFE6 2AD8 _H to	(Prohibited area)													
FFE6 2AFC _H														
FFE6 2B00 _H	OFMICNTD40													
FFE6 2B04 _H	ONMICNTD40													
FFE6 2B08 _H	OTOMICNTD40													
FFE6 2B0C _H	(Prohibited area)													
FFE6 2B10 _H	OFMIND40													
FFE6 2B14 _H	ONMIND40													
FFE6 2B18 _H	OTOMIND40													
FFE6 2B1C _H	(Prohibited area)													
FFE6 2B20 _H	ONCAP1D40													
FFE6 2B24 _H	ONCAP2D40(ICR2D40)													
FFE6 2B28 _H	(Prohibited area)													
FFE6 2B2C _H	(Prohibited area)													
FFE6 2B30 _H	OFCAP1D40(ICR1D40)													
FFE6 2B34 _H	OFCAP2D40													
FFE6 2B38 _H	(Prohibited area)													
FFE6 2B3C _H	(Prohibited area)													
FFE6 2B40 _H	OFMICNTD41													
FFE6 2B44 _H	ONMICNTD41													
FFE6 2B48 _H	OTOMICNTD41													
FFE6 2B4C _H	(Prohibited area)													
FFE6 2B50 _H	OFMIND41													
FFE6 2B54 _H	ONMIND41													
FFE6 2B58 _H	OTOMIND41													
FFE6 2B5C _H	(Prohibited area)													
FFE6 2B60 _H	ONCAP1D41													

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (14/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 2B64 _H	ONCAP2D41(ICR2D41)												
FFE6 2B68 _H	(Prohibited area)												
FFE6 2B6C _H	(Prohibited area)												
FFE6 2B70 _H	OFCAP1D41(ICR1D41)												
FFE6 2B74 _H	OFCAP2D41												
FFE6 2B78 _H	(Prohibited area)												
FFE6 2B7C _H	(Prohibited area)												
FFE6 2B80 _H	OFMICNTD42												
FFE6 2B84 _H	ONMICNTD42												
FFE6 2B88 _H	OTOMICNTD42												
FFE6 2B8C _H	(Prohibited area)												
FFE6 2B90 _H	OFMIND42												
FFE6 2B94 _H	ONMIND42												
FFE6 2B98 _H	OTOMIND42												
FFE6 2B9C _H	(Prohibited area)												
FFE6 2BA0 _H	ONCAP1D42												
FFE6 2BA4 _H	ONCAP2D42(ICR2D42)												
FFE6 2BA8 _H	(Prohibited area)												
FFE6 2BAC _H	(Prohibited area)												
FFE6 2BB0 _H	OFCAP1D42(ICR1D42)												
FFE6 2BB4 _H	OFCAP2D42												
FFE6 2BB8 _H	(Prohibited area)												
FFE6 2BBC _H	(Prohibited area)												
FFE6 2BC0 _H	OFMICNTD43												
FFE6 2BC4 _H	ONMICNTD43												
FFE6 2BC8 _H	OTOMICNTD43												
FFE6 2BCC _H	(Prohibited area)												
FFE6 2BD0 _H	OFMIND43												
FFE6 2BD4 _H	ONMIND43												
FFE6 2BD8 _H	OTOMIND43												
FFE6 2BDC _H	(Prohibited area)												
FFE6 2BE0 _H	ONCAP1D43												
FFE6 2BE4 _H	ONCAP2D43(ICR2D43)												
FFE6 2BE8 _H	(Prohibited area)												
FFE6 2BEC _H	(Prohibited area)												
FFE6 2BF0 _H	OFCAP1D43(ICR1D43)												
FFE6 2BF4 _H	OFCAP2D43												
FFE6 2BF8 _H	(Prohibited area)												
FFE6 2BFC _H	(Prohibited area)												
FFE6 2C00 _H	DCRD5						TCRD5						D5
FFE6 2C04 _H	TIOR2D5						TIOR1D5						
FFE6 2C08 _H	DSR2D5			DSR1D5			DSCRD5			DSTRD5			
FFE6 2C0C _H	TSCRD5						TSRD5						
FFE6 2C10 _H	(Prohibited area)			ODRD5			OSELRD5			TOCRD5			
FFE6 2C14 _H	MIGSELD5			MIGCRD5			(Prohibited area)			TICTSELD5			

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (15/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page	
FFE6 2C18 _H	OSBRD5													
FFE6 2C1C _H	(Prohibited area)						TCCRLRD5							
FFE6 2C20 _H	TCNT1D5													
FFE6 2C24 _H	TCNT2D5													
FFE6 2C28 _H	CUCR1D5													
FFE6 2C2C _H	CUCR2D5													
FFE6 2C30 _H	(Prohibited area)									RCR1D5				
FFE6 2C34 _H	(Prohibited area)									RCR2D5				
FFE6 2C38 _H	(Prohibited area)													
FFE6 2C3C _H	TIORD5						TCMPED5			TOCCRD5				
FFE6 2C40 _H	OCR1D50													
FFE6 2C44 _H	(Prohibited area)													
FFE6 2C48 _H	OCR2D50													
FFE6 2C4C _H	(Prohibited area)													
FFE6 2C50 _H	ICR1D50													
FFE6 2C54 _H	ICR2D50													
FFE6 2C58 _H	DCNTD50													
FFE6 2C5C _H	(Prohibited area)													
FFE6 2C60 _H	OCR1D51													
FFE6 2C64 _H	(Prohibited area)													
FFE6 2C68 _H	OCR2D51													
FFE6 2C6C _H	(Prohibited area)													
FFE6 2C70 _H	ICR1D51													
FFE6 2C74 _H	ICR2D51													
FFE6 2C78 _H	DCNTD51													
FFE6 2C7C _H	(Prohibited area)													
FFE6 2C80 _H	OCR1D52													
FFE6 2C84 _H	(Prohibited area)													
FFE6 2C88 _H	OCR2D52													
FFE6 2C8C _H	(Prohibited area)													
FFE6 2C90 _H	ICR1D52													
FFE6 2C94 _H	ICR2D52													
FFE6 2C98 _H	DCNTD52													
FFE6 2C9C _H	(Prohibited area)													
FFE6 2CA0 _H	OCR1D53													
FFE6 2CA4 _H	(Prohibited area)													
FFE6 2CA8 _H	OCR2D53													
FFE6 2CAC _H	(Prohibited area)													
FFE6 2CB0 _H	ICR1D53													
FFE6 2CB4 _H	ICR2D53													
FFE6 2CB8 _H	DCNTD53													
FFE6 2CBC _H	(Prohibited area)													
FFE6 2CC0 _H	(Prohibited area)									TIER1D5				
FFE6 2CC4 _H	TIER2D5													
FFE6 2CC8 _H	(Prohibited area)						TIER3D5							

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (16/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 2CCC _H	(Prohibited area)												
FFE6 2CD0 _H	(Prohibited area)						TSR2D5						
FFE6 2CD4 _H	(Prohibited area)						TSCR2D5						
FFE6 2CD8 _H to	(Prohibited area)												
FFE6 2CFC _H													
FFE6 2D00 _H	OFMICNTD50												
FFE6 2D04 _H	ONMICNTD50												
FFE6 2D08 _H	OTOMICNTD50												
FFE6 2D0C _H	(Prohibited area)												
FFE6 2D10 _H	OFMIND50												
FFE6 2D14 _H	ONMIND50												
FFE6 2D18 _H	OTOMIND50												
FFE6 2D1C _H	(Prohibited area)												
FFE6 2D20 _H	ONCAP1D50												
FFE6 2D24 _H	ONCAP2D50(ICR2D50)												
FFE6 2D28 _H	(Prohibited area)												
FFE6 2D2C _H	(Prohibited area)												
FFE6 2D30 _H	OFCAP1D50(ICR1D50)												
FFE6 2D34 _H	OFCAP2D50												
FFE6 2D38 _H	(Prohibited area)												
FFE6 2D3C _H	(Prohibited area)												
FFE6 2D40 _H	OFMICNTD51												
FFE6 2D44 _H	ONMICNTD51												
FFE6 2D48 _H	OTOMICNTD51												
FFE6 2D4C _H	(Prohibited area)												
FFE6 2D50 _H	OFMIND51												
FFE6 2D54 _H	ONMIND51												
FFE6 2D58 _H	OTOMIND51												
FFE6 2D5C _H	(Prohibited area)												
FFE6 2D60 _H	ONCAP1D51												
FFE6 2D64 _H	ONCAP2D51(ICR2D51)												
FFE6 2D68 _H	(Prohibited area)												
FFE6 2D6C _H	(Prohibited area)												
FFE6 2D70 _H	OFCAP1D51(ICR1D51)												
FFE6 2D74 _H	OFCAP2D51												
FFE6 2D78 _H	(Prohibited area)												
FFE6 2D7C _H	(Prohibited area)												
FFE6 2D80 _H	OFMICNTD52												
FFE6 2D84 _H	ONMICNTD52												
FFE6 2D88 _H	OTOMICNTD52												
FFE6 2D8C _H	(Prohibited area)												
FFE6 2D90 _H	OFMIND52												
FFE6 2D94 _H	ONMIND52												
FFE6 2D98 _H	OTOMIND52												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (17/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 2D9C _H	(Prohibited area)												
FFE6 2DA0 _H	ONCAP1D52												
FFE6 2DA4 _H	ONCAP2D52(ICR2D52)												
FFE6 2DA8 _H	(Prohibited area)												
FFE6 2DAC _H	(Prohibited area)												
FFE6 2DB0 _H	OFCAP1D52(ICR1D52)												
FFE6 2DB4 _H	OFCAP2D52												
FFE6 2DB8 _H	(Prohibited area)												
FFE6 2DBC _H	(Prohibited area)												
FFE6 2DC0 _H	OFMICNTD53												
FFE6 2DC4 _H	ONMICNTD53												
FFE6 2DC8 _H	OTOMICNTD53												
FFE6 2DCC _H	(Prohibited area)												
FFE6 2DD0 _H	OFMIND53												
FFE6 2DD4 _H	ONMIND53												
FFE6 2DD8 _H	OTOMIND53												
FFE6 2DDC _H	(Prohibited area)												
FFE6 2DE0 _H	ONCAP1D53												
FFE6 2DE4 _H	ONCAP2D53(ICR2D53)												
FFE6 2DE8 _H	(Prohibited area)												
FFE6 2DEC _H	(Prohibited area)												
FFE6 2DF0 _H	OFCAP1D53(ICR1D53)												
FFE6 2DF4 _H	OFCAP2D53												
FFE6 2DF8 _H	(Prohibited area)												
FFE6 2DFC _H	(Prohibited area)												
FFE6 2E00 _H	DCRD6						TCRD6						D6
FFE6 2E04 _H	TIOR2D6						TIOR1D6						
FFE6 2E08 _H	DSR2D6			DSR1D6			DSCRD6			DSTRD6			
FFE6 2E0C _H	TSCRD6						TSRD6						
FFE6 2E10 _H	(Prohibited area)			ODRD6			OSELRD6			TOCRD6			
FFE6 2E14 _H	MIGSELD6			MIGCRD6			(Prohibited area)			TICTSELD6			
FFE6 2E18 _H	OSBRD6												
FFE6 2E1C _H	(Prohibited area)						TCCRLRD6						
FFE6 2E20 _H	TCNT1D6												
FFE6 2E24 _H	TCNT2D6												
FFE6 2E28 _H	CUCR1D6												
FFE6 2E2C _H	CUCR2D6												
FFE6 2E30 _H	(Prohibited area)									RCR1D6			
FFE6 2E34 _H	(Prohibited area)									RCR2D6			
FFE6 2E38 _H	(Prohibited area)												
FFE6 2E3C _H	TIORD6						TCMPED6			TOCCRD6			
FFE6 2E40 _H	OCR1D60												
FFE6 2E44 _H	(Prohibited area)												
FFE6 2E48 _H	OCR2D60												
FFE6 2E4C _H	(Prohibited area)												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (18/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 2E50 _H	ICR1D60												
FFE6 2E54 _H	ICR2D60												
FFE6 2E58 _H	DCNTD60												
FFE6 2E5C _H	(Prohibited area)												
FFE6 2E60 _H	OCR1D61												
FFE6 2E64 _H	(Prohibited area)												
FFE6 2E68 _H	OCR2D61												
FFE6 2E6C _H	(Prohibited area)												
FFE6 2E70 _H	ICR1D61												
FFE6 2E74 _H	ICR2D61												
FFE6 2E78 _H	DCNTD61												
FFE6 2E7C _H	(Prohibited area)												
FFE6 2E80 _H	OCR1D62												
FFE6 2E84 _H	(Prohibited area)												
FFE6 2E88 _H	OCR2D62												
FFE6 2E8C _H	(Prohibited area)												
FFE6 2E90 _H	ICR1D62												
FFE6 2E94 _H	ICR2D62												
FFE6 2E98 _H	DCNTD62												
FFE6 2E9C _H	(Prohibited area)												
FFE6 2EA0 _H	OCR1D63												
FFE6 2EA4 _H	(Prohibited area)												
FFE6 2EA8 _H	OCR2D63												
FFE6 2EAC _H	(Prohibited area)												
FFE6 2EB0 _H	ICR1D63												
FFE6 2EB4 _H	ICR2D63												
FFE6 2EB8 _H	DCNTD63												
FFE6 2EBC _H	(Prohibited area)												
FFE6 2EC0 _H	(Prohibited area)									TIER1D6			
FFE6 2EC4 _H	TIER2D6												
FFE6 2EC8 _H	(Prohibited area)						TIER3D6						
FFE6 2ECC _H	(Prohibited area)												
FFE6 2ED0 _H	(Prohibited area)						TSR2D6						
FFE6 2ED4 _H	(Prohibited area)						TSCR2D6						
FFE6 2ED8 _H	(Prohibited area)												
to													
FFE6 2EFC _H													
FFE6 2F00 _H	OFMICNTD60												
FFE6 2F04 _H	ONMICNTD60												
FFE6 2F08 _H	OTOMICNTD60												
FFE6 2F0C _H	(Prohibited area)												
FFE6 2F10 _H	OFMIND60												
FFE6 2F14 _H	ONMIND60												
FFE6 2F18 _H	OTOMIND60												
FFE6 2F1C _H	(Prohibited area)												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (19/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 2F20 _H	ONCAP1D60												
FFE6 2F24 _H	ONCAP2D60(ICR2D60)												
FFE6 2F28 _H	(Prohibited area)												
FFE6 2F2C _H	(Prohibited area)												
FFE6 2F30 _H	OFCAP1D60(ICR1D60)												
FFE6 2F34 _H	OFCAP2D60												
FFE6 2F38 _H	(Prohibited area)												
FFE6 2F3C _H	(Prohibited area)												
FFE6 2F40 _H	OFMICNTD61												
FFE6 2F44 _H	ONMICNTD61												
FFE6 2F48 _H	OTOMICNTD61												
FFE6 2F4C _H	(Prohibited area)												
FFE6 2F50 _H	OFMIND61												
FFE6 2F54 _H	ONMIND61												
FFE6 2F58 _H	OTOMIND61												
FFE6 2F5C _H	(Prohibited area)												
FFE6 2F60 _H	ONCAP1D61												
FFE6 2F64 _H	ONCAP2D61(ICR2D61)												
FFE6 2F68 _H	(Prohibited area)												
FFE6 2F6C _H	(Prohibited area)												
FFE6 2F70 _H	OFCAP1D61(ICR1D61)												
FFE6 2F74 _H	OFCAP2D61												
FFE6 2F78 _H	(Prohibited area)												
FFE6 2F7C _H	(Prohibited area)												
FFE6 2F80 _H	OFMICNTD62												
FFE6 2F84 _H	ONMICNTD62												
FFE6 2F88 _H	OTOMICNTD62												
FFE6 2F8C _H	(Prohibited area)												
FFE6 2F90 _H	OFMIND62												
FFE6 2F94 _H	ONMIND62												
FFE6 2F98 _H	OTOMIND62												
FFE6 2F9C _H	(Prohibited area)												
FFE6 2FA0 _H	ONCAP1D62												
FFE6 2FA4 _H	ONCAP2D62(ICR2D62)												
FFE6 2FA8 _H	(Prohibited area)												
FFE6 2FAC _H	(Prohibited area)												
FFE6 2FB0 _H	OFCAP1D62(ICR1D62)												
FFE6 2FB4 _H	OFCAP2D62												
FFE6 2FB8 _H	(Prohibited area)												
FFE6 2FBC _H	(Prohibited area)												
FFE6 2FC0 _H	OFMICNTD63												
FFE6 2FC4 _H	ONMICNTD63												
FFE6 2FC8 _H	OTOMICNTD63												
FFE6 2FCC _H	(Prohibited area)												
FFE6 2FD0 _H	OFMIND63												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (20/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 2FD4 _H	ONMIND63												
FFE6 2FD8 _H	OTOMIND63												
FFE6 2FDC _H	(Prohibited area)												
FFE6 2FE0 _H	ONCAP1D63												
FFE6 2FE4 _H	ONCAP2D63(ICR2D63)												
FFE6 2FE8 _H	(Prohibited area)												
FFE6 2FEC _H	(Prohibited area)												
FFE6 2FF0 _H	OFCAP1D63(ICR1D63)												
FFE6 2FF4 _H	OFCAP2D63												
FFE6 2FF8 _H	(Prohibited area)												
FFE6 2FFC _H	(Prohibited area)												
FFE6 3000 _H	DCRD7						TCRD7						D7
FFE6 3004 _H	TIOR2D7						TIOR1D7						
FFE6 3008 _H	DSR2D7			DSR1D7			DSCRD7			DSTRD7			
FFE6 300C _H	TSCRD7						TSRD7						
FFE6 3010 _H	(Prohibited area)			ODRD7			OSELRD7			TOCRD7			
FFE6 3014 _H	MIGSELD7			MIGCRD7			(Prohibited area)			TICTSELD7			
FFE6 3018 _H	OSBRD7												
FFE6 301C _H	(Prohibited area)						TCCRLRD7						
FFE6 3020 _H	TCNT1D7												
FFE6 3024 _H	TCNT2D7												
FFE6 3028 _H	CUCR1D7												
FFE6 302C _H	CUCR2D7												
FFE6 3030 _H	(Prohibited area)									RCR1D7			
FFE6 3034 _H	(Prohibited area)									RCR2D7			
FFE6 3038 _H	(Prohibited area)												
FFE6 303C _H	TIORD7						TCMPED7			TOCCRD7			
FFE6 3040 _H	OCR1D70												
FFE6 3044 _H	(Prohibited area)												
FFE6 3048 _H	OCR2D70												
FFE6 304C _H	(Prohibited area)												
FFE6 3050 _H	ICR1D70												
FFE6 3054 _H	ICR2D70												
FFE6 3058 _H	DCNTD70												
FFE6 305C _H	(Prohibited area)												
FFE6 3060 _H	OCR1D71												
FFE6 3064 _H	(Prohibited area)												
FFE6 3068 _H	OCR2D71												
FFE6 306C _H	(Prohibited area)												
FFE6 3070 _H	ICR1D71												
FFE6 3074 _H	ICR2D71												
FFE6 3078 _H	DCNTD71												
FFE6 307C _H	(Prohibited area)												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (21/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page	
FFE6 3080 _H	OCR1D72													
FFE6 3084 _H	(Prohibited area)													
FFE6 3088 _H	OCR2D72													
FFE6 308C _H	(Prohibited area)													
FFE6 3090 _H	ICR1D72													
FFE6 3094 _H	ICR2D72													
FFE6 3098 _H	DCNTD72													
FFE6 309C _H	(Prohibited area)													
FFE6 30A0 _H	OCR1D73													
FFE6 30A4 _H	(Prohibited area)													
FFE6 30A8 _H	OCR2D73													
FFE6 30AC _H	(Prohibited area)													
FFE6 30B0 _H	ICR1D73													
FFE6 30B4 _H	ICR2D73													
FFE6 30B8 _H	DCNTD73													
FFE6 30BC _H	(Prohibited area)													
FFE6 30C0 _H	(Prohibited area)									TIER1D7				
FFE6 30C4 _H	TIER2D7													
FFE6 30C8 _H	(Prohibited area)						TIER3D7							
FFE6 30CC _H	(Prohibited area)													
FFE6 30D0 _H	(Prohibited area)						TSR2D7							
FFE6 30D4 _H	(Prohibited area)						TSCR2D7							
FFE6 30D8 _H	(Prohibited area)													
to														
FFE6 30FC _H														
FFE6 3100 _H	OFMICNTD70													
FFE6 3104 _H	ONMICNTD70													
FFE6 3108 _H	OTOMICNTD70													
FFE6 310C _H	(Prohibited area)													
FFE6 3110 _H	OFMIND70													
FFE6 3114 _H	ONMIND70													
FFE6 3118 _H	OTOMIND70													
FFE6 311C _H	(Prohibited area)													
FFE6 3120 _H	ONCAP1D70													
FFE6 3124 _H	ONCAP2D70(ICR2D70)													
FFE6 3128 _H	(Prohibited area)													
FFE6 312C _H	(Prohibited area)													
FFE6 3130 _H	OFCAP1D70(ICR1D70)													
FFE6 3134 _H	OFCAP2D70													
FFE6 3138 _H	(Prohibited area)													
FFE6 313C _H	(Prohibited area)													
FFE6 3140 _H	OFMICNTD71													
FFE6 3144 _H	ONMICNTD71													
FFE6 3148 _H	OTOMICNTD71													
FFE6 314C _H	(Prohibited area)													

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (22/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 3150 _H													
FFE6 3154 _H													
FFE6 3158 _H													
FFE6 315C _H													
FFE6 3160 _H													
FFE6 3164 _H													
FFE6 3168 _H													
FFE6 316C _H													
FFE6 3170 _H													
FFE6 3174 _H													
FFE6 3178 _H													
FFE6 317C _H													
FFE6 3180 _H													
FFE6 3184 _H													
FFE6 3188 _H													
FFE6 318C _H													
FFE6 3190 _H													
FFE6 3194 _H													
FFE6 3198 _H													
FFE6 319C _H													
FFE6 31A0 _H													
FFE6 31A4 _H													
FFE6 31A8 _H													
FFE6 31AC _H													
FFE6 31B0 _H													
FFE6 31B4 _H													
FFE6 31B8 _H													
FFE6 31BC _H													
FFE6 31C0 _H													
FFE6 31C4 _H													
FFE6 31C8 _H													
FFE6 31CC _H													
FFE6 31D0 _H													
FFE6 31D4 _H													
FFE6 31D8 _H													
FFE6 31DC _H													
FFE6 31E0 _H													
FFE6 31E4 _H													
FFE6 31E8 _H													
FFE6 31EC _H													
FFE6 31F0 _H													
FFE6 31F4 _H													
FFE6 31F8 _H													
FFE6 31FC _H													

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (23/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 3200 _H	DCRD8						TCRD8						D8
FFE6 3204 _H	TIOR2D8						TIOR1D8						
FFE6 3208 _H	DSR2D8			DSR1D8			DSCRD8			DSTRD8			
FFE6 320C _H	TSCRD8						TSRD8						
FFE6 3210 _H	(Prohibited area)			ODRD8			OSELRD8			TOCRD8			
FFE6 3214 _H	MIGSELD8			MIGCRD8			(Prohibited area)			TICTSELD8			
FFE6 3218 _H	OSBRD8												
FFE6 321C _H	(Prohibited area)						TCCRLRD8						
FFE6 3220 _H	TCNT1D8												
FFE6 3224 _H	TCNT2D8												
FFE6 3228 _H	CUCR1D8												
FFE6 322C _H	CUCR2D8												
FFE6 3230 _H	(Prohibited area)									RCR1D8			
FFE6 3234 _H	(Prohibited area)									RCR2D8			
FFE6 3238 _H	(Prohibited area)												
FFE6 323C _H	TIORD8						TCMPED8			TOCCRD8			
FFE6 3240 _H	OCR1D80												
FFE6 3244 _H	(Prohibited area)												
FFE6 3248 _H	OCR2D80												
FFE6 324C _H	(Prohibited area)												
FFE6 3250 _H	ICR1D80												
FFE6 3254 _H	ICR2D80												
FFE6 3258 _H	DCNTD80												
FFE6 325C _H	(Prohibited area)												
FFE6 3260 _H	OCR1D81												
FFE6 3264 _H	(Prohibited area)												
FFE6 3268 _H	OCR2D81												
FFE6 326C _H	(Prohibited area)												
FFE6 3270 _H	ICR1D81												
FFE6 3274 _H	ICR2D81												
FFE6 3278 _H	DCNTD81												
FFE6 327C _H	(Prohibited area)												
FFE6 3280 _H	OCR1D82												
FFE6 3284 _H	(Prohibited area)												
FFE6 3288 _H	OCR2D82												
FFE6 328C _H	(Prohibited area)												
FFE6 3290 _H	ICR1D82												
FFE6 3294 _H	ICR2D82												
FFE6 3298 _H	DCNTD82												
FFE6 329C _H	(Prohibited area)												
FFE6 32A0 _H	OCR1D83												
FFE6 32A4 _H	(Prohibited area)												
FFE6 32A8 _H	OCR2D83												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (24/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 32AC _H	(Prohibited area)												
FFE6 32B0 _H	ICR1D83												
FFE6 32B4 _H	ICR2D83												
FFE6 32B8 _H	DCNTD83												
FFE6 32BC _H	(Prohibited area)												
FFE6 32C0 _H	(Prohibited area)									TIER1D8			
FFE6 32C4 _H	TIER2D8												
FFE6 32C8 _H	(Prohibited area)						TIER3D8						
FFE6 32CC _H	(Prohibited area)												
FFE6 32D0 _H	(Prohibited area)						TSR2D8						
FFE6 32D4 _H	(Prohibited area)						TSCR2D8						
FFE6 32D8 _H	(Prohibited area)												
to													
FFE6 32FC _H													
FFE6 3300 _H	OFMICNTD80												
FFE6 3304 _H	ONMICNTD80												
FFE6 3308 _H	OTOMICNTD80												
FFE6 330C _H	(Prohibited area)												
FFE6 3310 _H	OFMIND80												
FFE6 3314 _H	ONMIND80												
FFE6 3318 _H	OTOMIND80												
FFE6 331C _H	(Prohibited area)												
FFE6 3320 _H	ONCAP1D80												
FFE6 3324 _H	ONCAP2D80(ICR2D80)												
FFE6 3328 _H	(Prohibited area)												
FFE6 332C _H	(Prohibited area)												
FFE6 3330 _H	OFCAP1D80(ICR1D80)												
FFE6 3334 _H	OFCAP2D80												
FFE6 3338 _H	(Prohibited area)												
FFE6 333C _H	(Prohibited area)												
FFE6 3340 _H	OFMICNTD81												
FFE6 3344 _H	ONMICNTD81												
FFE6 3348 _H	OTOMICNTD81												
FFE6 334C _H	(Prohibited area)												
FFE6 3350 _H	OFMIND81												
FFE6 3354 _H	ONMIND81												
FFE6 3358 _H	OTOMIND81												
FFE6 335C _H	(Prohibited area)												
FFE6 3360 _H	ONCAP1D81												
FFE6 3364 _H	ONCAP2D81(ICR2D81)												
FFE6 3368 _H	(Prohibited area)												
FFE6 336C _H	(Prohibited area)												
FFE6 3370 _H	OFCAP1D81(ICR1D81)												
FFE6 3374 _H	OFCAP2D81												
FFE6 3378 _H	(Prohibited area)												

Timer D Registers (FFE6 2000_H to FFE6 3FFF_H) (25/25)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 337C _H	(Prohibited area)												
FFE6 3380 _H	OFMICNTD82												
FFE6 3384 _H	ONMICNTD82												
FFE6 3388 _H	OTOMICNTD82												
FFE6 338C _H	(Prohibited area)												
FFE6 3390 _H	OFMIND82												
FFE6 3394 _H	ONMIND82												
FFE6 3398 _H	OTOMIND82												
FFE6 339C _H	(Prohibited area)												
FFE6 33A0 _H	ONCAP1D82												
FFE6 33A4 _H	ONCAP2D82(ICR2D82)												
FFE6 33A8 _H	(Prohibited area)												
FFE6 33AC _H	(Prohibited area)												
FFE6 33B0 _H	OFCAP1D82(ICR1D82)												
FFE6 33B4 _H	OFCAP2D82												
FFE6 33B8 _H	(Prohibited area)												
FFE6 33BC _H	(Prohibited area)												
FFE6 33C0 _H	OFMICNTD83												
FFE6 33C4 _H	ONMICNTD83												
FFE6 33C8 _H	OTOMICNTD83												
FFE6 33CC _H	(Prohibited area)												
FFE6 33D0 _H	OFMIND83												
FFE6 33D4 _H	ONMIND83												
FFE6 33D8 _H	OTOMIND83												
FFE6 33DC _H	(Prohibited area)												
FFE6 33E0 _H	ONCAP1D83												
FFE6 33E4 _H	ONCAP2D83(ICR2D83)												
FFE6 33E8 _H	(Prohibited area)												
FFE6 33EC _H	(Prohibited area)												
FFE6 33F0 _H	OFCAP1D83(ICR1D83)												
FFE6 33F4 _H	OFCAP2D83												
FFE6 33F8 _H	(Prohibited area)												
FFE6 33FC _H	(Prohibited area)												
FFE6 3400 _H	(Prohibited area)												
to													
FFE6 3FFC _H													

Timer E Registers (FFE6 4000_H to FFE6 4FFF_H) (1/9)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 4000 _H	(Prohibited area)						TSTRE						Common to E
FFE6 4004 _H	(Prohibited area)						SBRLNE						
FFE6 4008 _H to FFE6 40FC _H	(Prohibited area)												
FFE6 4100 _H	(Prohibited area)									SSTRE0			E0
FFE6 4104 _H	(Prohibited area)									PSCRE0			
FFE6 4108 _H	(Prohibited area)			RLDCRE0			(Prohibited area)			TCRE0			
FFE6 410C _H	(Prohibited area)			SOLVLE0			POECRE0						
FFE6 4110 _H	TSCRE0						TSRE0						
FFE6 4114 _H	TIERE0						(Prohibited area)			TOCRE0			
FFE6 4118 _H	PSCCRE03			PSCCRE02			PSCCRE01			PSCCRE00			
FFE6 411C _H	(Prohibited area)									FCTRGE0			
FFE6 4120 _H	(Prohibited area)												
FFE6 4124 _H	TCNTE00												
FFE6 4128 _H	CYLRE00												
FFE6 412C _H	DTRE00												
FFE6 4130 _H	CRLDE00												
FFE6 4134 _H	DRLDE00												
FFE6 4138 _H	(Prohibited area)												
FFE6 413C _H	(Prohibited area)												
FFE6 4140 _H	(Prohibited area)												
FFE6 4144 _H	TCNTE01												
FFE6 4148 _H	CYLRE01												
FFE6 414C _H	DTRE01												
FFE6 4150 _H	CRLDE01												
FFE6 4154 _H	DRLDE01												
FFE6 4158 _H	(Prohibited area)												
FFE6 415C _H	(Prohibited area)												
FFE6 4160 _H	(Prohibited area)												
FFE6 4164 _H	TCNTE02												
FFE6 4168 _H	CYLRE02												
FFE6 416C _H	DTRE02												
FFE6 4170 _H	CRLDE02												
FFE6 4174 _H	DRLDE02												
FFE6 4178 _H	(Prohibited area)												
FFE6 417C _H	(Prohibited area)												
FFE6 4180 _H	(Prohibited area)												
FFE6 4184 _H	TCNTE03												
FFE6 4188 _H	CYLRE03												
FFE6 418C _H	DTRE03												
FFE6 4190 _H	CRLDE03												
FFE6 4194 _H	DRLDE03												
FFE6 4198 _H	(Prohibited area)												

Timer E Registers (FFE6 4000_H to FFE6 4FFF_H) (2/9)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 419C _H	(Prohibited area)												
FFE6 41A0 _H	(Prohibited area)												
to													
FFE6 41FC _H													
FFE6 4200 _H	(Prohibited area)									SSTRE1		E1	
FFE6 4204 _H	(Prohibited area)									PSCRE1			
FFE6 4208 _H	(Prohibited area)			RLDCRE1			(Prohibited area)			TCRE1			
FFE6 420C _H	(Prohibited area)			SOLVLE1			POECRE1						
FFE6 4210 _H	TSCRE1						TSRE1						
FFE6 4214 _H	TIERE1						(Prohibited area)			TOCRE1			
FFE6 4218 _H	PSCCRE13			PSCCRE12			PSCCRE11			PSCCRE10			
FFE6 421C _H	(Prohibited area)									FCTRGE1			
FFE6 4220 _H	(Prohibited area)												
FFE6 4224 _H	TCNTE10												
FFE6 4228 _H	CYLRE10												
FFE6 422C _H	DTRE10												
FFE6 4230 _H	CRLDE10												
FFE6 4234 _H	DRLDE10												
FFE6 4238 _H	(Prohibited area)												
FFE6 423C _H	(Prohibited area)												
FFE6 4240 _H	(Prohibited area)												
FFE6 4244 _H	TCNTE11												
FFE6 4248 _H	CYLRE11												
FFE6 424C _H	DTRE11												
FFE6 4250 _H	CRLDE11												
FFE6 4254 _H	DRLDE11												
FFE6 4258 _H	(Prohibited area)												
FFE6 425C _H	(Prohibited area)												
FFE6 4260 _H	(Prohibited area)												
FFE6 4264 _H	TCNTE12												
FFE6 4268 _H	CYLRE12												
FFE6 426C _H	DTRE12												
FFE6 4270 _H	CRLDE12												
FFE6 4274 _H	DRLDE12												
FFE6 4278 _H	(Prohibited area)												
FFE6 427C _H	(Prohibited area)												
FFE6 4280 _H	(Prohibited area)												
FFE6 4284 _H	TCNTE13												
FFE6 4288 _H	CYLRE13												
FFE6 428C _H	DTRE13												
FFE6 4290 _H	CRLDE13												
FFE6 4294 _H	DRLDE13												

Timer E Registers (FFE6 4000_H to FFE6 4FFF_H) (3/9)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 4298 _H	(Prohibited area)												
to													
FFE6 42FC _H													
FFE6 4300 _H	(Prohibited area)									SSTRE2		E2	
FFE6 4304 _H	(Prohibited area)									PSCRE2			
FFE6 4308 _H	(Prohibited area)			RLDCRE2			(Prohibited area)			TCRE2			
FFE6 430C _H	(Prohibited area)			SOLVLE2			POECRE2						
FFE6 4310 _H	TSCRE2						TSRE2						
FFE6 4314 _H	TIERE2						(Prohibited area)			TOCRE2			
FFE6 4318 _H	PSCCRE23			PSCCRE22			PSCCRE21			PSCCRE20			
FFE6 431C _H	(Prohibited area)									FCTRGE2			
FFE6 4320 _H	(Prohibited area)												
FFE6 4324 _H	TCNTE20												
FFE6 4328 _H	CYLRE20												
FFE6 432C _H	DTRE20												
FFE6 4330 _H	CRLDE20												
FFE6 4334 _H	DRLDE20												
FFE6 4338 _H	(Prohibited area)												
FFE6 433C _H	(Prohibited area)												
FFE6 4340 _H	(Prohibited area)												
FFE6 4344 _H	TCNTE21												
FFE6 4348 _H	CYLRE21												
FFE6 434C _H	DTRE21												
FFE6 4350 _H	CRLDE21												
FFE6 4354 _H	DRLDE21												
FFE6 4358 _H	(Prohibited area)												
FFE6 435C _H	(Prohibited area)												
FFE6 4360 _H	(Prohibited area)												
FFE6 4364 _H	TCNTE22												
FFE6 4368 _H	CYLRE22												
FFE6 436C _H	DTRE22												
FFE6 4370 _H	CRLDE22												
FFE6 4374 _H	DRLDE22												
FFE6 4378 _H	(Prohibited area)												
FFE6 437C _H	(Prohibited area)												
FFE6 4380 _H	(Prohibited area)												
FFE6 4384 _H	TCNTE23												
FFE6 4388 _H	CYLRE23												
FFE6 438C _H	DTRE23												
FFE6 4390 _H	CRLDE23												
FFE6 4394 _H	DRLDE23												
FFE6 4398 _H	(Prohibited area)												
FFE6 439C _H	(Prohibited area)												

Timer E Registers (FFE6 4000_H to FFE6 4FFF_H) (4/9)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 43A0 _H	(Prohibited area)												
to													
FFE6 43FC _H													
FFE6 4400 _H	(Prohibited area)									SSTRE3		E3	
FFE6 4404 _H	(Prohibited area)									PSCRE3			
FFE6 4408 _H	(Prohibited area)			RLDCRE3			(Prohibited area)			TCRE3			
FFE6 440C _H	(Prohibited area)			SOLVLE3			POECRE3						
FFE6 4410 _H	TSCRE3						TSRE3						
FFE6 4414 _H	TIERE3						(Prohibited area)			TOCRE3			
FFE6 4418 _H	PSCCRE33			PSCCRE32			PSCCRE31			PSCCRE30			
FFE6 441C _H	(Prohibited area)									FCTRGE3			
FFE6 4420 _H	(Prohibited area)												
FFE6 4424 _H	TCNTE30												
FFE6 4428 _H	CYLRE30												
FFE6 442C _H	DTRE30												
FFE6 4430 _H	CRLDE30												
FFE6 4434 _H	DRLDE30												
FFE6 4438 _H	(Prohibited area)												
FFE6 443C _H	(Prohibited area)												
FFE6 4440 _H	(Prohibited area)												
FFE6 4444 _H	TCNTE31												
FFE6 4448 _H	CYLRE31												
FFE6 444C _H	DTRE31												
FFE6 4450 _H	CRLDE31												
FFE6 4454 _H	DRLDE31												
FFE6 4458 _H	(Prohibited area)												
FFE6 445C _H	(Prohibited area)												
FFE6 4460 _H	(Prohibited area)												
FFE6 4464 _H	TCNTE32												
FFE6 4468 _H	CYLRE32												
FFE6 446C _H	DTRE32												
FFE6 4470 _H	CRLDE32												
FFE6 4474 _H	DRLDE32												
FFE6 4478 _H	(Prohibited area)												
FFE6 447C _H	(Prohibited area)												
FFE6 4480 _H	(Prohibited area)												
FFE6 4484 _H	TCNTE33												
FFE6 4488 _H	CYLRE33												
FFE6 448C _H	DTRE33												
FFE6 4490 _H	CRLDE33												
FFE6 4494 _H	DRLDE33												
FFE6 4498 _H	(Prohibited area)												
FFE6 449C _H	(Prohibited area)												

Timer E Registers (FFE6 4000_H to FFE6 4FFF_H) (5/9)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 44A0 _H	(Prohibited area)												
to													
FFE6 44FC _H													
FFE6 4500 _H	(Prohibited area)									SSTRE4		E4	
FFE6 4504 _H	(Prohibited area)									PSCRE4			
FFE6 4508 _H	(Prohibited area)			RLDCRE4			(Prohibited area)			TCRE4			
FFE6 450C _H	(Prohibited area)			SOLVLE4			POECRE4						
FFE6 4510 _H	TSCRE4						TSRE4						
FFE6 4514 _H	TIERE4						(Prohibited area)			TOCRE4			
FFE6 4518 _H	PSCCRE43			PSCCRE42			PSCCRE41			PSCCRE40			
FFE6 451C _H	(Prohibited area)									FCTRGE4			
FFE6 451C _H	(Prohibited area)												
FFE6 4520 _H	(Prohibited area)												
FFE6 4524 _H	TCNTE40												
FFE6 4528 _H	CYLRE40												
FFE6 452C _H	DTRE40												
FFE6 4530 _H	CRLDE40												
FFE6 4534 _H	DRLDE40												
FFE6 4538 _H	(Prohibited area)												
FFE6 453C _H	(Prohibited area)												
FFE6 4540 _H	(Prohibited area)												
FFE6 4544 _H	TCNTE41												
FFE6 4548 _H	CYLRE41												
FFE6 454C _H	DTRE41												
FFE6 4550 _H	CRLDE41												
FFE6 4554 _H	DRLDE41												
FFE6 4558 _H	(Prohibited area)												
FFE6 455C _H	(Prohibited area)												
FFE6 4560 _H	(Prohibited area)												
FFE6 4564 _H	TCNTE42												
FFE6 4568 _H	CYLRE42												
FFE6 456C _H	DTRE42												
FFE6 4570 _H	CRLDE42												
FFE6 4574 _H	DRLDE42												
FFE6 4578 _H	(Prohibited area)												
FFE6 457C _H	(Prohibited area)												
FFE6 4580 _H	(Prohibited area)												
FFE6 4584 _H	TCNTE43												
FFE6 4588 _H	CYLRE43												
FFE6 458C _H	DTRE43												
FFE6 4590 _H	CRLDE43												
FFE6 4594 _H	DRLDE43												
FFE6 4598 _H	(Prohibited area)												
FFE6 459C _H	(Prohibited area)												

Timer E Registers (FFE6 4000_H to FFE6 4FFF_H) (6/9)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 45A0 _H	(Prohibited area)												
to													
FFE6 45FC _H													
FFE6 4600 _H	(Prohibited area)									SSTRE5		E5	
FFE6 4604 _H	(Prohibited area)									PSCRE5			
FFE6 4608 _H	(Prohibited area)			RLDCRE5			(Prohibited area)			TCRE5			
FFE6 460C _H	(Prohibited area)			SOLVLE5			POECRE5						
FFE6 4610 _H	TSCRE5						TSRE5						
FFE6 4614 _H	TIERE5						(Prohibited area)			TOCRE5			
FFE6 4618 _H	PSCCRE53			PSCCRE52			PSCCRE51			PSCCRE50			
FFE6 461C _H	(Prohibited area)									FCTRGE5			
FFE6 4620 _H	(Prohibited area)												
FFE6 4624 _H	TCNTE50												
FFE6 4628 _H	CYLRE50												
FFE6 462C _H	DTRE50												
FFE6 4630 _H	CRLDE50												
FFE6 4634 _H	DRLDE50												
FFE6 4638 _H	(Prohibited area)												
FFE6 463C _H	(Prohibited area)												
FFE6 4640 _H	(Prohibited area)												
FFE6 4644 _H	TCNTE51												
FFE6 4648 _H	CYLRE51												
FFE6 464C _H	DTRE51												
FFE6 4650 _H	CRLDE51												
FFE6 4654 _H	DRLDE51												
FFE6 4658 _H	(Prohibited area)												
FFE6 465C _H	(Prohibited area)												
FFE6 4660 _H	(Prohibited area)												
FFE6 4664 _H	TCNTE52												
FFE6 4668 _H	CYLRE52												
FFE6 466C _H	DTRE52												
FFE6 4670 _H	CRLDE52												
FFE6 4674 _H	DRLDE52												
FFE6 4678 _H	(Prohibited area)												
FFE6 467C _H	(Prohibited area)												
FFE6 4680 _H	(Prohibited area)												
FFE6 4684 _H	TCNTE53												
FFE6 4688 _H	CYLRE53												
FFE6 468C _H	DTRE53												
FFE6 4690 _H	CRLDE53												
FFE6 4694 _H	DRLDE53												
FFE6 4698 _H	(Prohibited area)												
FFE6 469C _H	(Prohibited area)												

Timer E Registers (FFE6 4000_H to FFE6 4FFF_H) (7/9)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 46A0 _H	(Prohibited area)												
to													
FFE6 46FC _H													
FFE6 4700 _H	(Prohibited area)									SSTRE6		E6	
FFE6 4704 _H	(Prohibited area)									PSCRE6			
FFE6 4708 _H	(Prohibited area)			RLDCRE6			(Prohibited area)			TCRE6			
FFE6 470C _H	(Prohibited area)												
FFE6 4710 _H	TSCRE6						TSRE6						
FFE6 4714 _H	TIERE6						(Prohibited area)			TOCRE6			
FFE6 4718 _H	PSCCRE63			PSCCRE62			PSCCRE61			PSCCRE60			
FFE6 471C _H	(Prohibited area)									FCTRGE6			
FFE6 4720 _H	(Prohibited area)												
FFE6 4724 _H	TCNTE60												
FFE6 4728 _H	CYLRE60												
FFE6 472C _H	DTRE60												
FFE6 4730 _H	CRLDE60												
FFE6 4734 _H	DRLDE60												
FFE6 4738 _H	(Prohibited area)												
FFE6 473C _H	(Prohibited area)												
FFE6 4740 _H	(Prohibited area)												
FFE6 4744 _H	TCNTE61												
FFE6 4748 _H	CYLRE61												
FFE6 474C _H	DTRE61												
FFE6 4750 _H	CRLDE61												
FFE6 4754 _H	DRLDE61												
FFE6 4758 _H	(Prohibited area)												
FFE6 475C _H	(Prohibited area)												
FFE6 4760 _H	(Prohibited area)												
FFE6 4764 _H	TCNTE62												
FFE6 4768 _H	CYLRE62												
FFE6 476C _H	DTRE62												
FFE6 4770 _H	CRLDE62												
FFE6 4774 _H	DRLDE62												
FFE6 4778 _H	(Prohibited area)												
FFE6 477C _H	(Prohibited area)												
FFE6 4780 _H	(Prohibited area)												
FFE6 4784 _H	TCNTE63												
FFE6 4788 _H	CYLRE63												
FFE6 478C _H	DTRE63												
FFE6 4790 _H	CRLDE63												
FFE6 4794 _H	DRLDE63												
FFE6 4798 _H	(Prohibited area)												
FFE6 479C _H	(Prohibited area)												

Timer E Registers (FFE6 4000_H to FFE6 4FFF_H) (8/9)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 47A0 _H	(Prohibited area)												
to													
FFE6 47FC _H													
FFE6 4800 _H	(Prohibited area)									SSTRE7		E7	
FFE6 4804 _H	(Prohibited area)									PSCRE7			
FFE6 4808 _H	(Prohibited area)			RLDCRE7			(Prohibited area)			TCRE7			
FFE6 480C _H	(Prohibited area)												
FFE6 4810 _H	TSCRE7						TSRE7						
FFE6 4814 _H	TIERE7						(Prohibited area)			TOCRE7			
FFE6 4818 _H	PSCCRE73			PSCCRE72			PSCCRE71			PSCCRE70			
FFE6 481C _H	(Prohibited area)									FCTRGE7			
FFE6 4820 _H	(Prohibited area)												
FFE6 4824 _H	TCNTE70												
FFE6 4828 _H	CYLRE70												
FFE6 482C _H	DTRE70												
FFE6 4830 _H	CRLDE70												
FFE6 4834 _H	DRLDE70												
FFE6 4838 _H	(Prohibited area)												
FFE6 483C _H	(Prohibited area)												
FFE6 4840 _H	(Prohibited area)												
FFE6 4844 _H	TCNTE71												
FFE6 4848 _H	CYLRE71												
FFE6 484C _H	DTRE71												
FFE6 4850 _H	CRLDE71												
FFE6 4854 _H	DRLDE71												
FFE6 4858 _H	(Prohibited area)												
FFE6 485C _H	(Prohibited area)												
FFE6 4860 _H	(Prohibited area)												
FFE6 4864 _H	TCNTE72												
FFE6 4868 _H	CYLRE72												
FFE6 486C _H	DTRE72												
FFE6 4870 _H	CRLDE72												
FFE6 4874 _H	DRLDE72												
FFE6 4878 _H	(Prohibited area)												
FFE6 487C _H	(Prohibited area)												
FFE6 4880 _H	(Prohibited area)												
FFE6 4884 _H	TCNTE73												
FFE6 4888 _H	CYLRE73												
FFE6 488C _H	DTRE73												
FFE6 4890 _H	CRLDE73												
FFE6 4894 _H	DRLDE73												
FFE6 4898 _H	(Prohibited area)												
FFE6 489C _H	(Prohibited area)												

Timer E Registers (FFE6 4000_H to FFE6 4FFF_H) (9/9)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 48A0 _H	(Prohibited area)												
to													
FFE6 48FC _H													
FFE6 4900 _H	(Prohibited area)									SSTRE8		E8	
FFE6 4904 _H	(Prohibited area)									PSCRE8			
FFE6 4908 _H	(Prohibited area)			RLDCRE8			(Prohibited area)			TCRE8			
FFE6 490C _H	(Prohibited area)												
FFE6 4910 _H	TSCRE8						TSRE8						
FFE6 4914 _H	TIERE8						(Prohibited area)			TOCRE8			
FFE6 4918 _H	PSCCRE83			PSCCRE82			PSCCRE81			PSCCRE80			
FFE6 491C _H	(Prohibited area)									FCTRGE8			
FFE6 4920 _H	(Prohibited area)												
FFE6 4924 _H	TCNTE80												
FFE6 4928 _H	CYLRE80												
FFE6 492C _H	DTRE80												
FFE6 4930 _H	CRLDE80												
FFE6 4934 _H	DRLDE80												
FFE6 4938 _H	(Prohibited area)												
FFE6 493C _H	(Prohibited area)												
FFE6 4940 _H	(Prohibited area)												
FFE6 4944 _H	TCNTE81												
FFE6 4948 _H	CYLRE81												
FFE6 494C _H	DTRE81												
FFE6 4950 _H	CRLDE81												
FFE6 4954 _H	DRLDE81												
FFE6 4958 _H	(Prohibited area)												
FFE6 495C _H	(Prohibited area)												
FFE6 4960 _H	(Prohibited area)												
FFE6 4964 _H	TCNTE82												
FFE6 4968 _H	CYLRE82												
FFE6 496C _H	DTRE82												
FFE6 4970 _H	CRLDE82												
FFE6 4974 _H	DRLDE82												
FFE6 4978 _H	(Prohibited area)												
FFE6 497C _H	(Prohibited area)												
FFE6 4980 _H	(Prohibited area)												
FFE6 4984 _H	TCNTE83												
FFE6 4988 _H	CYLRE83												
FFE6 498C _H	DTRE83												
FFE6 4990 _H	CRLDE83												
FFE6 4994 _H	DRLDE83												
FFE6 4998 _H	(Prohibited area)												
to													
FFE6 4FFC _H													

Timer F Registers (FFE6 5000_H to FFE6 59FF_H) (1/7)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 5000 _H	TSTRF												Common to F
FFE6 5004 _H	NCMCR1F												
FFE6 5008 _H	NCMCR2F												
FFE6 500C _H	NCCRF												
FFE6 5010 _H	(Prohibited area)						PVFCRF						
FFE6 5014 _H	(Prohibited area)												
to													
FFE6 503C _H													
FFE6 5040 _H	(Prohibited area)			TCR2F0			(Prohibited area)			TCR1F0			F0
FFE6 5044 _H	TIERF0			TSCRF0			TSRF0			BKCRF0			
FFE6 5048 _H	NCRFA0						NCNTFA0						
FFE6 504C _H	NCRFB0						NCNTFB0						
FFE6 5050 _H	ECNTAF0												
FFE6 5054 _H	GRAF0												
FFE6 5058 _H	(Prohibited area)						ECNTBF0						
FFE6 505C _H	(Prohibited area)						GRBF0						
FFE6 5060 _H	ECNTCF0												
FFE6 5064 _H	GRCF0 (when ARSWCF0 = "0") BGRCF0 (when ARSWCF0 = "1")												
FFE6 5068 _H	GRDF0												
FFE6 506C _H	CDRF0												
FFE6 5070 _H	(Prohibited area)												
FFE6 5074 _H	(Prohibited area)												
FFE6 5078 _H	(Prohibited area)												
FFE6 507C _H	(Prohibited area)												
FFE6 5080 _H	(Prohibited area)			TCR2F1			(Prohibited area)			TCR1F1			F1
FFE6 5084 _H	TIERF1			TSCRF1			TSRF1			BKCRF1			
FFE6 5088 _H	NCRFA1						NCNTFA1						
FFE6 508C _H	NCRFB1						NCNTFB1						
FFE6 5090 _H	ECNTAF1												
FFE6 5094 _H	GRAF1												
FFE6 5098 _H	(Prohibited area)						ECNTBF1						
FFE6 509C _H	(Prohibited area)						GRBF1						
FFE6 50A0 _H	ECNTCF1												
FFE6 50A4 _H	GRCF1 (when ARSWCF1 = "0") BGRCF1 (when ARSWCF1 = "1")												
FFE6 50A8 _H	GRDF1												
FFE6 50AC _H	CDRF1												
FFE6 50B0 _H	(Prohibited area)												
FFE6 50B4 _H	(Prohibited area)												
FFE6 50B8 _H	(Prohibited area)												
FFE6 50BC _H	(Prohibited area)												
FFE6 50C0 _H	(Prohibited area)			TCR2F2			(Prohibited area)			TCR1F2			F2
FFE6 50C4 _H	TIERF2			TSCRF2			TSRF2			BKCRF2			

Timer F Registers (FFE6 5000_H to FFE6 59FF_H) (2/7)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 50C8 _H	NCRFA2						NCNTFA2						
FFE6 50CC _H	NCRFB2						NCNTFB2						
FFE6 50D0 _H	ECNTAF2												
FFE6 50D4 _H	GRAF2												
FFE6 50D8 _H	(Prohibited area)						ECNTBF2						
FFE6 50DC _H	(Prohibited area)						GRBF2						
FFE6 50E0 _H	ECNTCF2												
FFE6 50E4 _H	GRCF2 (when ARSWCF2 = "0") BGRCF2 (when ARSWCF2 = "1")												
FFE6 50E8 _H	GRDF2												
FFE6 50EC _H	CDRF2												
FFE6 50F0 _H	(Prohibited area)												
FFE6 50F4 _H	(Prohibited area)												
FFE6 50F8 _H	(Prohibited area)												
FFE6 50FC _H	(Prohibited area)												
FFE6 5100 _H	(Prohibited area)			TCR2F3			(Prohibited area)			TCR1F3			F3
FFE6 5104 _H	TIERF3			TSCRF3			TSRF3			BKCRF3			
FFE6 5108 _H	NCRFA3						NCNTFA3						
FFE6 510C _H	(Prohibited area)												
FFE6 5110 _H	ECNTAF3												
FFE6 5114 _H	GRAF3 (when ARSWAF3 = "0") BGRAF3 (when ARSWAF3 = "1")												
FFE6 5118 _H	(Prohibited area)						ECNTBF3						
FFE6 511C _H	(Prohibited area)						GRBF3						
FFE6 5120 _H	ECNTCF3												
FFE6 5124 _H	GRCF3 (when ARSWCF3 = "0") BGRCF3 (when ARSWCF3 = "1")												
FFE6 5128 _H	GRDF3 (when ARSWDF3 = "0") BGRDF3 (when ARSWDF3 = "1")												
FFE6 512C _H	CDRF3												
FFE6 5130 _H	(Prohibited area)												
FFE6 5134 _H	(Prohibited area)												
FFE6 5138 _H	(Prohibited area)												
FFE6 513C _H	(Prohibited area)												
FFE6 5140 _H	(Prohibited area)			TCR2F4			(Prohibited area)			TCR1F4			F4
FFE6 5144 _H	TIERF4			TSCRF4			TSRF4			BKCRF4			
FFE6 5148 _H	NCRFA4						NCNTFA4						
FFE6 514C _H	(Prohibited area)												
FFE6 5150 _H	ECNTAF4												
FFE6 5154 _H	GRAF4 (when ARSWAF4 = "0") BGRAF4 (when ARSWAF4 = "1")												
FFE6 5158 _H	(Prohibited area)						ECNTBF4						
FFE6 515C _H	(Prohibited area)						GRBF4						
FFE6 5160 _H	ECNTCF4												

Timer F Registers (FFE6 5000_H to FFE6 59FF_H) (3/7)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 5164 _H	GRCF4 (when ARSWCF4 = "0") BGRCF4 (when ARSWCF4 = "1")												
FFE6 5168 _H	GRDF4 (when ARSWDF4 = "0") BGRDF4 (when ARSWDF4 = "1")												
FFE6 516C _H	CDRF4												
FFE6 5170 _H	(Prohibited area)												
FFE6 5174 _H	(Prohibited area)												
FFE6 5178 _H	(Prohibited area)												
FFE6 517C _H	(Prohibited area)												
FFE6 5180 _H	(Prohibited area)			TCR2F5			(Prohibited area)			TCR1F5			F5
FFE6 5184 _H	TIERF5			TSCRF5			TSRF5			BKCRF5			
FFE6 5188 _H	NCRFA5						NCNTFA5						
FFE6 518C _H	(Prohibited area)												
FFE6 5190 _H	ECNTAF5												
FFE6 5194 _H	GRAF5 (when ARSWAF5 = "0") BGRAF5 (when ARSWAF5 = "1")												
FFE6 5198 _H	(Prohibited area)						ECNTBF5						
FFE6 519C _H	(Prohibited area)						GRBF5						
FFE6 51A0 _H	ECNTCF5												
FFE6 51A4 _H	GRCF5 (when ARSWCF5 = "0") BGRCF5 (when ARSWCF5 = "1")												
FFE6 51A8 _H	GRDF5 (when ARSWDF5 = "0") BGRDF5 (when ARSWDF5 = "1")												
FFE6 51AC _H	CDRF5												
FFE6 51B0 _H	(Prohibited area)												
FFE6 51B4 _H	(Prohibited area)												
FFE6 51B8 _H	(Prohibited area)												
FFE6 51BC _H	(Prohibited area)												
FFE6 51C0 _H	(Prohibited area)			TCR2F6			(Prohibited area)			TCR1F6			F6
FFE6 51C4 _H	TIERF6			TSCRF6			TSRF6			BKCRF6			
FFE6 51C8 _H	NCRFA6						NCNTFA6						
FFE6 51CC _H	(Prohibited area)												
FFE6 51D0 _H	ECNTAF6												
FFE6 51D4 _H	GRAF6 (when ARSWAF6 = "0") BGRAF6 (when ARSWAF6 = "1")												
FFE6 51D8 _H	(Prohibited area)						ECNTBF6						
FFE6 51DC _H	(Prohibited area)						GRBF6						
FFE6 51E0 _H	ECNTCF6												
FFE6 51E4 _H	GRCF6 (when ARSWCF6 = "0") BGRCF6 (when ARSWCF6 = "1")												
FFE6 51E8 _H	GRDF6 (when ARSWDF6 = "0") BGRDF6 (when ARSWDF6 = "1")												
FFE6 51EC _H	CDRF6												
FFE6 51F0 _H	(Prohibited area)												
FFE6 51F4 _H	(Prohibited area)												

Timer F Registers (FFE6 5000_H to FFE6 59FF_H) (4/7)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 51F8 _H	(Prohibited area)												
FFE6 51FC _H	(Prohibited area)												
FFE6 5200 _H	(Prohibited area)			TCR2F7			(Prohibited area)			TCR1F7			F7
FFE6 5204 _H	TIERF7			TSCRF7			TSRF7			BKCRF7			
FFE6 5208 _H	NCRFA7						NCNTFA7						
FFE6 520C _H	(Prohibited area)												
FFE6 5210 _H	ECNTAF7												
FFE6 5214 _H	GRAF7 (when ARSWAF7 = "0") BGRAF7 (when ARSWAF7 = "1")												
FFE6 5218 _H	(Prohibited area)						ECNTBF7						
FFE6 521C _H	(Prohibited area)						GRBF7						
FFE6 5220 _H	ECNTCF7												
FFE6 5224 _H	GRCF7 (when ARSWCF7 = "0") BGRCF7 (when ARSWCF7 = "1")												
FFE6 5228 _H	GRDF7 (when ARSWDF7 = "0") BGRDF7 (when ARSWDF7 = "1")												
FFE6 522C _H	CDRF7												
FFE6 5230 _H	(Prohibited area)												
FFE6 5234 _H	(Prohibited area)												
FFE6 5238 _H	(Prohibited area)												
FFE6 523C _H	(Prohibited area)												
FFE6 5240 _H	(Prohibited area)			TCR2F8			(Prohibited area)			TCR1F8			F8
FFE6 5244 _H	TIERF8			TSCRF8			TSRF8			BKCRF8			
FFE6 5248 _H	NCRFA8						NCNTFA8						
FFE6 524C _H	(Prohibited area)												
FFE6 5250 _H	ECNTAF8												
FFE6 5254 _H	GRAF8 (when ARSWAF8 = "0") BGRAF8 (when ARSWAF8 = "1")												
FFE6 5258 _H	(Prohibited area)						ECNTBF8						
FFE6 525C _H	(Prohibited area)						GRBF8						
FFE6 5260 _H	ECNTCF8												
FFE6 5264 _H	GRCF8 (when ARSWCF8 = "0") BGRCF8 when ARSWCF8 = "1")												
FFE6 5268 _H	GRDF8 (when ARSWDF8 = "0") BGRDF8 (when ARSWDF8 = "1")												
FFE6 526C _H	CDRF8												
FFE6 5270 _H	(Prohibited area)												
FFE6 5274 _H	(Prohibited area)												
FFE6 5278 _H	(Prohibited area)												
FFE6 527C _H	(Prohibited area)												
FFE6 5280 _H	(Prohibited area)			TCR2F9			(Prohibited area)			TCR1F9			F9
FFE6 5284 _H	TIERF9			TSCRF9			TSRF9			BKCRF9			
FFE6 5288 _H	NCRFA9						NCNTFA9						
FFE6 528C _H	(Prohibited area)												

Timer F Registers (FFE6 5000_H to FFE6 59FF_H) (5/7)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 5290 _H	ECNTAF9												
FFE6 5294 _H	GRAF9 (when ARSWAF9 = "0") BGRAF9 (when ARSWAF9 = "1")												
FFE6 5298 _H	(Prohibited area)						ECNTBF9						
FFE6 529C _H	(Prohibited area)						GRBF9						
FFE6 52A0 _H	ECNTCF9												
FFE6 52A4 _H	GRCF9 (when ARSWCF9 = "0") BGRCF9 (when ARSWCF9 = "1")												
FFE6 52A8 _H	GRDF9 (when ARSWDF9 = "0") BGRDF9 (when ARSWDF9 = "1")												
FFE6 52AC _H	CDRF9												
FFE6 52B0 _H	(Prohibited area)												
FFE6 52B4 _H	(Prohibited area)												
FFE6 52B8 _H	(Prohibited area)												
FFE6 52BC _H	(Prohibited area)												
FFE6 52C0 _H	(Prohibited area)	TCR2F10				(Prohibited area)	TCR1F10				F10		
FFE6 52C4 _H	TIERF10		TSCRF10			TSRF10			BKCRF10				
FFE6 52C8 _H	NCRFA10						NCNTFA10						
FFE6 52CC _H	(Prohibited area)												
FFE6 52D0 _H	ECNTAF10												
FFE6 52D4 _H	GRAF10 (when ARSWAF10 = "0") BGRAF10 (when ARSWAF10 = "1")												
FFE6 52D8 _H	(Prohibited area)						ECNTBF10						
FFE6 52DC _H	(Prohibited area)						GRBF10						
FFE6 52E0 _H	ECNTCF10												
FFE6 52E4 _H	GRCF10 (when ARSWCF10 = "0") BGRCF10 (when ARSWCF10 = "1")												
FFE6 52E8 _H	GRDF10 (when ARSWDF10 = "0") BGRDF10 (when ARSWDF10 = "1")												
FFE6 52EC _H	CDRF10												
FFE6 52F0 _H	(Prohibited area)												
FFE6 52F4 _H	(Prohibited area)												
FFE6 52F8 _H	(Prohibited area)												
FFE6 52FC _H	(Prohibited area)												
FFE6 5300 _H	(Prohibited area)	TCR2F11				(Prohibited area)	TCR1F11				F11		
FFE6 5304 _H	TIERF11		TSCRF11			TSRF11			BKCRF11				
FFE6 5308 _H	NCRFA11						NCNTFA11						
FFE6 530C _H	(Prohibited area)												
FFE6 5310 _H	ECNTAF11												
FFE6 5314 _H	GRAF11 (when ARSWAF11 = "0") BGRAF11 (when ARSWAF11 = "1")												
FFE6 5318 _H	(Prohibited area)						ECNTBF11						
FFE6 531C _H	(Prohibited area)						GRBF11						
FFE6 5320 _H	ECNTCF11												

Timer F Registers (FFE6 5000_H to FFE6 59FF_H) (6/7)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 5324 _H	GRCF11 (when ARSWCF11 = "0") BGRCF11 (when ARSWCF11 = "1")												
FFE6 5328 _H	GRDF11 (when ARSWDF11 = "0") BGRDF11 (when ARSWDF11 = "1")												
FFE6 532C _H	CDRF11												
FFE6 5330 _H	(Prohibited area)												
FFE6 5334 _H	(Prohibited area)												
FFE6 5338 _H	(Prohibited area)												
FFE6 533C _H	(Prohibited area)												
FFE6 5340 _H	(Prohibited area)			TCR2F12			(Prohibited area)			TCR1F12			F12
FFE6 5344 _H	TIERF12			TSCRF12			TSRF12			BKCRF12			
FFE6 5348 _H	NCRFA12						NCNTFA12						
FFE6 534C _H	(Prohibited area)												
FFE6 5350 _H	ECNTAF12												
FFE6 5354 _H	GRAF12 (when ARSWAF12 = "0") BGRAF12 (when ARSWAF12 = "1")												
FFE6 5358 _H	(Prohibited area)						ECNTBF12						
FFE6 535C _H	(Prohibited area)						GRBF12						
FFE6 5360 _H	ECNTCF12												
FFE6 5364 _H	GRCF12 (when ARSWCF12 = "0") BGRCF12 (when ARSWCF12 = "1")												
FFE6 5368 _H	GRDF12 (when ARSWDF12 = "0") BGRDF12 (when ARSWDF12 = "1")												
FFE6 536C _H	CDRF12												
FFE6 5370 _H	(Prohibited area)												
FFE6 5374 _H	(Prohibited area)												
FFE6 5378 _H	(Prohibited area)												
FFE6 537C _H	(Prohibited area)												
FFE6 5380 _H	(Prohibited area)			TCR2F13			(Prohibited area)			TCR1F13			F13
FFE6 5384 _H	TIERF13			TSCRF13			TSRF13			BKCRF13			
FFE6 5388 _H	NCRFA13						NCNTFA13						
FFE6 538C _H	(Prohibited area)												
FFE6 5390 _H	ECNTAF13												
FFE6 5394 _H	GRAF13 (when ARSWAF13 = "0") BGRAF13 (when ARSWAF13 = "1")												
FFE6 5398 _H	(Prohibited area)						ECNTBF13						
FFE6 539C _H	(Prohibited area)						GRBF13						
FFE6 53A0 _H	ECNTCF13												
FFE6 53A4 _H	GRCF13 (when ARSWCF13 = "0") BGRCF13 (when ARSWCF13 = "1")												
FFE6 53A8 _H	GRDF13 (when ARSWDF13 = "0") BGRDF13 (when ARSWDF13 = "1")												
FFE6 53AC _H	CDRF13												
FFE6 53B0 _H	(Prohibited area)												
FFE6 53B4 _H	(Prohibited area)												

Timer F Registers (FFE6 5000_H to FFE6 59FF_H) (7/7)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 53B8 _H	(Prohibited area)												
FFE6 53BC _H	(Prohibited area)												
FFE6 53C0 _H	(Prohibited area)			TCR2F14			(Prohibited area)			TCR1F14			F14
FFE6 53C4 _H	TIERF14			TSCRF14			TSRF14			BKCRF14			
FFE6 53C8 _H	NCRFA14						NCNTFA14						
FFE6 53CC _H	(Prohibited area)												
FFE6 53D0 _H	ECNTAF14												
FFE6 53D4 _H	GRAF14 (when ARSWAF14 = "0") BGRAF14 (when ARSWAF14 = "1")												
FFE6 53D8 _H	(Prohibited area)						ECNTBF14						
FFE6 53DC _H	(Prohibited area)						GRBF14						
FFE6 53E0 _H	ECNTCF14												
FFE6 53E4 _H	GRCF14 (when ARSWCF14 = "0") BGRCF14 (when ARSWCF14 = "1")												
FFE6 53E8 _H	GRDF14 (when ARSWDF14 = "0") BGRDF14 (when ARSWDF14 = "1")												
FFE6 53EC _H	CDRF14												
FFE6 53F0 _H	(Prohibited area)												
FFE6 53F4 _H	(Prohibited area)												
FFE6 53F8 _H	(Prohibited area)												
FFE6 53FC _H	(Prohibited area)												
FFE6 5400 _H	(Prohibited area)			TCR2F15			(Prohibited area)			TCR1F15			F15
FFE6 5404 _H	TIERF15			TSCRF15			TSRF15			BKCRF15			
FFE6 5408 _H	NCRFA15						NCNTFA15						
FFE6 540C _H	(Prohibited area)												
FFE6 5410 _H	ECNTAF15												
FFE6 5414 _H	GRAF15 (when ARSWAF15 = "0") BGRAF15 (when ARSWAF15 = "1")												
FFE6 5418 _H	(Prohibited area)						ECNTBF15						
FFE6 541C _H	(Prohibited area)						GRBF15						
FFE6 5420 _H	ECNTCF15												
FFE6 5424 _H	GRCF15 (when ARSWCF15 = "0") BGRCF15 (when ARSWCF15 = "1")												
FFE6 5428 _H	GRDF15 (when ARSWDF15 = "0") BGRDF15 (when ARSWDF15 = "1")												
FFE6 542C _H	CDRF15												
FFE6 5430 _H	(Prohibited area)												
to													
FFE6 58FC _H													

Timer G Registers (FFE6 5C00_H to FFE6 5CFF_H) (1/2)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 5C00 _H	(Prohibited area)						TSTRG						Common to G
FFE6 5C04 _H	(Prohibited area)						TIERG						
FFE6 5C08 _H	(Prohibited area)						RLDCRG						
FFE6 5C0C _H	(Prohibited area)												
FFE6 5C10 _H	TSCRG0		TSRG0			(Prohibited area)			TCRG0			G0	
FFE6 5C14 _H	TCNTG0												
FFE6 5C18 _H	OCRG0												
FFE6 5C1C _H	RLDG0												
FFE6 5C20 _H	TSCRG1		TSRG1			(Prohibited area)			TCRG1			G1	
FFE6 5C24 _H	TCNTG1												
FFE6 5C28 _H	OCRG1												
FFE6 5C2C _H	RLDG1												
FFE6 5C30 _H	TSCRG2		TSRG2			(Prohibited area)			TCRG2			G2	
FFE6 5C34 _H	TCNTG2												
FFE6 5C38 _H	OCRG2												
FFE6 5C3C _H	RLDG2												
FFE6 5C40 _H	TSCRG3		TSRG3			(Prohibited area)			TCRG3			G3	
FFE6 5C44 _H	TCNTG3												
FFE6 5C48 _H	OCRG3												
FFE6 5C4C _H	RLDG3												
FFE6 5C50 _H	TSCRG4		TSRG4			(Prohibited area)			TCRG4			G4	
FFE6 5C54 _H	TCNTG4												
FFE6 5C58 _H	OCRG4												
FFE6 5C5C _H	RLDG4												
FFE6 5C60 _H	TSCRG5		TSRG5			(Prohibited area)			TCRG5			G5	
FFE6 5C64 _H	TCNTG5												
FFE6 5C68 _H	OCRG5												
FFE6 5C6C _H	RLDG5												
FFE6 5C70 _H	TSCRG6		TSRG6			(Prohibited area)			TCRG6			G6	
FFE6 5C74 _H	TCNTG6												
FFE6 5C78 _H	OCRG6												
FFE6 5C7C _H	RLDG6												
FFE6 5C80 _H	TSCRG7		TSRG7			(Prohibited area)			TCRG7			G7	
FFE6 5C84 _H	TCNTG7												
FFE6 5C88 _H	OCRG7												
FFE6 5C8C _H	RLDG7												
FFE6 5C90 _H	TSCRG8		TSRG8			(Prohibited area)			TCRG8			G8	
FFE6 5C94 _H	TCNTG8												
FFE6 5C98 _H	OCRG8												
FFE6 5C9C _H	RLDG8												
FFE6 5CA0 _H	TSCRG9		TSRG9			(Prohibited area)			TCRG9			G9	
FFE6 5CA4 _H	TCNTG9												
FFE6 5CA8 _H	OCRG9												
FFE6 5CAC _H	RLDG9												

Timer G Registers (FFE6 5C00_H to FFE6 5CFF_H) (2/2)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 5CB0 _H	(Prohibited area)												
to													
FFE6 5CFC _H													

Trigger Select Function (FFE6 8000_H to FFE6 80FF_H)

Address	b31	+3	b24	b23	+2	b16	b15	+1	b8	b7	+0	b0	Page
FFE6 8000 _H	ATUINTSELA												int select
FFE6 8004 _H	(Prohibited area)												
FFE6 8008 _H	(Prohibited area)												
FFE6 800C _H	ATUINTSELD0												
FFE6 8010 _H	ATUINTSELD1												
FFE6 8014 _H	ATUINTSELD2												
FFE6 8018 _H	ATUINTSELD3												
FFE6 801C _H	ATUINTSELD4												
FFE6 8020 _H	(Prohibited area)												
to													
FFE6 8028 _H													
FFE6 802C _H	ATUDMASELB												dma select
FFE6 8030 _H	ATUDMASELCD0												
FFE6 8034 _H	ATUDMASELCD1												
FFE6 8038 _H	ATUDMASELCD2												
FFE6 803C _H	ATUDMASELCD3												
FFE6 8040 _H	ATUDMASELCD4												
FFE6 8044 _H	ATUDMASELCD5												
FFE6 8048 _H	ATUDMASELCD6												
FFE6 804C _H	ATUDMASELCD7												
FFE6 8050 _H	ATUDMASELE												
FFE6 8054 _H	(Prohibited area)												
to													
FFE6 807C _H													
FFE6 8080 _H	ATUDFEENTQ0												dfe select
FFE6 8084 _H	ATUDFESEL0												
FFE6 8088 _H	ATUDFEENTQ1												
FFE6 808C _H	ATUDFESEL1												
FFE6 8090 _H	ATUDFEENTQ2												
FFE6 8094 _H	ATUDFESEL2												
FFE6 8098 _H	ATUDFESELD1T												
FFE6 809C _H	ATUDSSELDSTS												dsadc select
FFE6 80A0 _H	ATUCASELCATS												cadc select
FFE6 80A4 _H	ATUP5SSEL												psi5s select
FFE6 80A8 _H	(Prohibited area)												
to													
FFE6 80FF _H													

Access size of the ATU-V registers are shown below.

Common Controller Registers

Module Name	Register Name	Symbol	Access Size
ATU5CTRL	ATU Master Enable Register	ATUENR	8
	Clock Bus Control Register	CBCNT	8
	Noise Cancellation Mode Register	NCMR	8

Prescaler Registers

Module Name	Register Name	Symbol	Access Size
ATU5CTRL	Prescaler Registers x	PSCRx	16

DMA/AD Requests Automatic Switching Registers

Module Name	Register Name	Symbol	Access Size
ATU5CTRL	Trigger Status Register DMA0	TRGSRDMA0	8
	Trigger Select Register DMA00	TRGSELDMA00	8
	Trigger Select Register DMA01	TRGSELDMA01	8
	Trigger Select Register AD	TRGSELAD	8
	Trigger Status Register DMA1	TRGSRDMA1	8
	Trigger Select Register DMA10	TRGSELDMA10	8
	Trigger Select Register DMA11	TRGSELDMA11	8

Timer A Registers

Module Name	Register Name	Symbol	Access Size
ATU5A	Timer Control Register 1A	TCR1A	8
	Timer Control Register 2A	TCR2A	8, 16
	Timer Control Register 2AS1	TCR2AS1	8
	Timer Control Register 2AS2	TCR2AS2	8
	Timer Control Register 3A	TCR3A	8, 16
	Timer Control Register 4A	TCR4A	8, 16
	Timer Control Register 5A	TCR5A	8
	Noise Cancellation Mode Channel Register 1A	NCMCR1A	8
	Noise Cancellation Mode Channel Register 2A	NCMCR2A	8
	Timer I/O Control Register 1	TIOR1A	8, 16
	Timer I/O Control Register 2	TIOR2A	8, 16, 32
	Timer Interrupt Enable Register A	TIERA	8, 16
	Timer Status Register A	TSRA	8, 16
	Timer Status Clear Register A	TSCRA	8, 16
	Input Capture Registers Ax	ICRAx	32
	Input Capture Registers A2x	ICRA2x	32
	Free-Running Counter A	TCNTA	32
	Timer Input Signal Level Register	TILRA	8
	Noise Canceler Counters Ax	NCNTAx	16
	Noise Cancel Registers Ax	NCRAx	16

Timer B Registers (1/2)

Module Name	Register Name	Symbol	Access Size
ATU5B	Timer Control Register B	TCRB	8
	Timer Control Register BS1	TCRBS1	8
	Timer Control Register BS2	TCRBS2	8
	Timer I/O Control Register B	TIORB	8
	Timer Status Register B	TSRB	8, 16
	Timer Status Clear Register B	TSCRB	8, 16
	Timer Interrupt Control Register B	TICRB	8
	Timer Interrupt Enable Register B	TIERB	8, 16
	Channel Select Register B	CHSELBR	8
	Multiplied-and-Corrected Clock Generating Counter B4 Control Register	TCNT4CRB	8
	Edge Interval Measuring Counter B0	TCNTB0	32
	Edge Interval Measuring Counter B0S1	TCNTB0S1	32
	Edge Interval Measuring Counter B0S2	TCNTB0S2	32
	Input Capture Register B0	ICRB0	32
	Input Capture Register B0	DICRB0	32
	Mirror Input Capture Register B0	MIICRB0	32
	Input Capture Register B0S1	ICRB0S1	32
	Input Capture Register B0S2	ICRB0S2	32
	Record Registers B1 to B6	RECRB1 to RECRB6	32
	Record Registers B1	DRECRB1	32
	Record Backup Register B0 to B6	RBURB0 to RBURB6	32
	Input Capture Registers B30 to B37	ICRB30 to ICRB37	8
	Output Compare Register B0	OCRB0	32
	Output Compare Register B0S1	OCRB0S1	32
	Output Compare Register B0S2	OCRB0S2	32
	Event Counter B1	TCNTB1	8
	Event Counter B1S1	TCNTB1S1	8
	Event Counter B1S2	TCNTB1S2	8
	Output Compare Register B1	OCRB1	8
	Output Compare Register B1S1	OCRB1S1	8
	Output Compare Register B1S2	OCRB1S2	8
	Output Compare Register B10	OCRB10	8
	Output Compare Register B10S1	OCRB10S1	8
	Output Compare Register B10S2	OCRB10S2	8
	Output Compare Register B11	OCRB11	8
	Output Compare Register B11S1	OCRB11S1	8
	Output Compare Register B11S2	OCRB11S2	8
	Output Compare Register B12	OCRB12	8
	Output Compare Register B12S1	OCRB12S1	8
	Output Compare Register B12S2	OCRB12S2	8

Timer B Registers (2/2)

Module Name	Register Name	Symbol	Access Size
ATU5B	Input Capture Register B1	ICRB1	32
	Input Capture Register B1S1	ICRB1S1	32
	Input Capture Register B1S2	ICRB1S2	32
	Input Capture Register B2	ICRB2	32
	Input Capture Register B2S1	ICRB2S1	32
	Input Capture Register B2S2	ICRB2S2	32
	Load Register B	LDB	32
	Reload Register B	RLDB	32
	Reloadable Counter B2	TCNTB2	32
	Pulse Interval Multiplier Register 1	PIMR1	16
	Pulse Interval Multiplier Register 2	PIMR2	16
	Multiplied Clock Counter B6	TCNTB6	32
	Input Capture Register B6	ICRB6	32
	Multiplication Setting Register B6	RARB6	8
	Frequency-Multiplied Clock Counter B6M	TCNTB6M	32
	Output Compare Register B6	OCRB6	32
	Output Compare Register B7	OCRB7	32
	Output Compare Registers B20 to 43	OCRB20 to OCRB43	8
	Timer Sequencer Control Register	TSEQCRB	8
	Timer Sequencer Register	TSEQRB	8
	Timer Sequencer Enable Registers 0 to 2	TSEQENB0 to TSEQENB2	8
	Timer Event Cycle Correction Formula Definition Registers B0 to 23	TEPCFB0 to TEPCFB23	16
	Correction Event Counter B3	TCNTB3	32
	Output Compare Register B8	OCRB8	32
	Timer Event Cycle Correction Enable Registers 0 to 2	TEPCFENB0 to TEPCFENB2	8
	Timer Event Input Cycle Correction Result Retention Register	TEPCVALRB0	32
	Timer Event Input Cycle Correction Result Log Register	TEPCRECRB1	32
	Multiplied-and-Corrected Clock Counter B4	TCNTB4	32
	Multiplied-and-Corrected Clock Generating Counter B5	TCNTB5	32
	Correcting Counter Clear Flag Register B	TCCLFRB	8
	Correcting Counter Clearing Register B	TCCLRB	32

Timer C Registers

Module Name	Register Name	Symbol	Access Size
ATU5C	Timer Start Register C	TSTRC	8, 16
	Noise Canceler Control Register Cx	NCCRCx	8
	Timer Control Registers Cx	TCRCx	8, 16
	Timer Status Registers Cx	TSRCx	8, 16
	Timer Status Clear Register Cx	TSCRCx	8, 16
	Timer I/O Control Registers Cx	TIORCx	8, 16
	Timer Counters Cx	TCNTCx	32
	Timer General Registers Cxy	GRCxy	32
	Noise Canceler Counters Cxy	NCNTCxy	16
	Noise Canceler Registers Cxy	NCRCxy	16
	Output Compare Registers Cxy	OCRCxy	32
	Output Compare Mirror Registers Cxy	OCMRCxy	32
	Timer General Mirror Registers Cxy	GMRCxy	32
	Timer Interrupt Enable Registers Cx	TIERCx	8, 16
	Counter Upper-Limit Setting Compare Registers Cx	CUCRCx	32
	Range Comparison Value Setting Register 1Cx	RCR1Cx	8
	Range Comparison Value Setting Register 2Cx	RCR2Cx	8
	Noise Cancellation Mode Channel Register 1C	NCMCR1C	8, 16
	Noise Cancellation Mode Channel Register 2C	NCMCR2C	8, 16

Timer D Registers (1/2)

Module Name	Register Name	Symbol	Access Size
ATU5D	Timer Start Register D	TSTRD	8, 16
	Timer Control Registers Dx	TCRDx	8, 16
	Timer Compare Control Register Dx	TCCLRDx	8, 16
	Timer I/O Control Registers 1Dx	TIOR1Dx	8, 16
	Timer I/O Control Register 2Dx	TIOR2Dx	8, 16
	Timer Compare Match Control Register Dx	TCMPEDx	8
	Timer I/O Control Register Dx	TIORDx	8, 16
	Output Select Register Dx	OSELRDx	8
	Output Value Register Dx	ODRDx	8
	Down Counter Start Register Dx	DSTRDx	8
	Down Counter Status Registers 1Dx	DSR1Dx	8
	Down Count Status Register 2Dx	DSR2Dx	8
	Down Count Status Clear Register Dx	DSCRDx	8
	Down Counter Control Registers Dx	DCRDx	8, 16
	Timer Status Registers Dx	TSRDx	8, 16
	Timer Status Clear Register Dx	TSCRDx	8, 16
	Timer Output Control Registers Dx	TOCRDx	8
	Timer Output Channel Control Registers Dx	TOCCRDx	8
	Timer Offset Base Registers Dx	OSBRDx	32
	Timer Input Capture Trigger Select Register Dx	TICTSELDx	8
	Timer Counter 1Dx	TCNT1Dx	32
	Timer Counter 2Dx	TCNT2Dx	32
	Counter Upper-Limit Setting Compare Register 1Dx	CUCR1Dx	32
	Counter Upper-Limit Setting Compare Register 2Dx	CUCR2Dx	32
	Output Compare Registers 1Dxy	OCR1Dxy	32
	Range Comparison Value Setting Register 1Dx	RCR1Dx	8
	Output Compare Registers 2Dxy	OCR2Dxy	32
	Range Comparison Value Setting Register 2Dx	RCR2Dx	8
	Input Capture Register 1Dxy	ICR1Dxy	32
	Input capture register 2Dxy	ICR2Dxy	32
	Timer Down Counters Dxy	DCNTDxy	32
	Timer Interrupt Enable Register 1Dx	TIER1Dx	8
	Timer Interrupt Enable Register 2Dx	TIER2Dx	8, 16, 32
	Timer Interrupt Enable Register 3Dx	TIER3Dx	8, 16
	MIN Guard Control Register Dx	MIGCRDx	8
	MIN Guard Pulse Select Register Dx	MIGSELDx	8
	Pulse Output Off Minimum Width Setting Register Dxy	OFMINDxy	32
	Pulse Output On Minimum Width Setting Register Dxy	ONMINDxy	32
	Pulse Output On-to-On Minimum Width Setting Register Dxy	OTOMINDxy	32
	Pulse Output Off Minimum Width Measurement Counter Dxy	OFMICNTDxy	32

Timer D Registers (2/2)

Module Name	Register Name	Symbol	Access Size
ATU5D	Pulse Output On Minimum Width Measurement Counter Dxy	ONMICNTDxy	32
	Pulse Output On-to-On Minimum Width Measurement Counter Dxy	OTOMICNTDxy	32
	Input capture register ONCAP1Dxy	ONCAP1Dxy	32
	Input capture register ONCAP2Dxy	ONCAP2Dxy	32
	Input capture register OFCAP1Dxy	OFCAP1Dxy	32
	Input capture register OFCAP2Dxy	OFCAP2Dxy	32
	Timer Status Register 2Dx	TSR2Dx	8, 16
	Timer Status Clear Register 2Dx	TSCR2Dx	8, 16
	Common Input capture Select Register D	CCAPSEL D	8
	Common Input Capture Register D0-7	CICRD0-7	32

Timer E Registers

Module Name	Register Name	Symbol	Access Size
ATU5E	Timer Start Register E	TSTRE	8, 16
	Subblock Reload Enable Register E	SBRENE	8, 16
	Subblock Starting Registers Ex	SSTREx	8
	Prescaler Registers Ex	PSCREx	8
	Prescaler Channel Registers Exy	PSCCRExy	8
	Timer Control Register Ex	TCREx	8
	Reload Control Registers Ex	RLDCREx	8
	Output Shutoff Control Register Ex	POECREx	16
	Output Shutoff Level Setting Register Ex	SOLVLEx	8
	Timer Status Registers Ex	TSREx	8, 16
	Timer Status Clear Registers Ex	TSCREx	8, 16
	Timer Interrupt Enable Registers Ex	TIEREx	8, 16
	Timer Output Control Registers Ex	TOCREx	8
	Forced Compare Match Trigger Registers Ex	FCTRGEx	8
	Timer Counters Exy	TCNTExy	32
	Cycle-Setting Registers Exy	CYLRExy	32
	Duty Cycle Setting Registers Exy	DTRExy	32
	Cycle Reload Registers Exy	CRLDExy	32
	Duty Cycle Reload Registers Exy	DRLDExy	32

Timer F Registers

Module Name	Register Name	Symbol	Access Size
ATU5F	Timer Start Register F	TSTRF	8, 16, 32
	Noise Cancellation Mode Channel Register 1F	NCMCR1F	8, 16, 32
	Noise Cancellation Mode Channel Register 2F	NCMCR2F	8, 16, 32
	Noise Canceller Control Register F	NCCRF	8, 16, 32
	Private Function Control Register F	PVFCRF	16
	Timer Control Registers 1Fx	TCR1Fx	8
	Timer Control Registers 2Fx	TCR2Fx	8
	Timer Interrupt Enable Registers Fx	TIERFx	8
	Backup Control Register Fx	BKCRFx	8
	Timer Status Registers Fx	TSRFx	8
	Timer Status Clear Register Fx	TSCRFx	8
	Timer Measurement Counters AFx	ECNTAFx	32
	Event Counters Fx	ECNTBFx	16
	Time Measurement Counters CFx	ECNTCFx	32
	General Registers AFx	GRAFx	32
	Backup General Registers AFx	BGRAFx	32
	General Registers BFx	GRBFx	16
	General Registers CFx	GRCFx	32
	Backup General Registers CFx	BGRCFx	32
	General Registers DFx	GRDFx	32
	Backup General Registers DFx	BGRDFx	32
	Capture Output Registers Fx	CDRFx	32
	Noise Canceller Counters FAX	NCNTFAX	16
	Noise Canceller Counters FBx	NCNTFBx	16
	Noise Cancel Registers FAX	NCRFAX	16
	Noise Cancel Registers FBx	NCRFBx	16

Timer G Registers

Module Name	Register Name	Symbol	Access Size
ATU5G	Timer Start Register G	TSTRG	8, 16
	Timer Interrupt Enable Register G	TIERG	8, 16
	Reload Control Register G	RLDCRG	8, 16
	Timer Control Register Gx	TCRGx	8
	Timer Status Registers Gx	TSRGx	8
	Timer Status Clear Register Gx	TSCRGx	8
	Timer Counters Gx	TCNTGx	32
	Compare Match Registers Gx	OCRGx	32
	Reload Registers Gx	RLDGx	32

Trigger Select Function

Module Name	Register Name	Symbol	Access Size
ATU5TRG	ATU TimerA Interrupt selection control register	ATUINTSELA	8, 16, 32
	ATU TimerD Interrupt selection control register 0	ATUINTSELD0	8, 16, 32
	ATU TimerD Interrupt selection control register 1	ATUINTSELD1	8, 16, 32
	ATU TimerD Interrupt selection control register 2	ATUINTSELD2	8, 16, 32
	ATU TimerD Interrupt selection control register 3	ATUINTSELD3	8, 16, 32
	ATU TimerD Interrupt selection control register 4	ATUINTSELD4	8, 16, 32
	ATU TimerB DMA selection control register	ATUDMASELB	8, 16, 32
	ATU TimerC/TimerD DMA selection control register 0	ATUDMASELCD0	8, 16, 32
	ATU TimerC/TimerD DMA selection control register 1	ATUDMASELCD1	8, 16, 32
	ATU TimerC/TimerD DMA selection control register 2	ATUDMASELCD2	8, 16, 32
	ATU TimerC/TimerD DMA selection control register 3	ATUDMASELCD3	8, 16, 32
	ATU TimerC/TimerD DMA selection control register 4	ATUDMASELCD4	8, 16, 32
	ATU TimerC/TimerD DMA selection control register 5	ATUDMASELCD5	8, 16, 32
	ATU TimerC/TimerD DMA selection control register 6	ATUDMASELCD6	8, 16, 32
	ATU TimerC/TimerD DMA selection control register 7	ATUDMASELCD7	8, 16, 32
	ATU TimerE DMA selection control register	ATUDMASELE	8, 16, 32
	ATU to DFE Capture trigger i input enable	ATUDFEENTQi	8, 16, 32
	ATU to DFE Capture trigger i input select	ATUDFESELi	8, 16, 32
	ATU to DFE Timer FIFO Capture trigger select	ATUDFESELD1T	8, 16, 32
	ATU to DSADC read gate trigger select	ATUDSSELDSTS	8, 16, 32
ATU to CADC read gate trigger select	ATUCASELCATS	8, 16, 32	
ATU to PSI5S Time stamp trigger select	ATUP5SSEL	8, 16, 32	

30.3 Common Controller

30.3.1 Overview of Operation

The common controller controls the ATU-V module as a whole. For example, it enables and disables the prescalers and timer counters for timers A to G and controls the clock bus.

Clock Bus

The clock bus consists of six signal lines used to distribute the clock signals for counting (counter enabling signals) to the timer channels. The timer counters on each of the channels run in synchronization with the internal peripheral clock (PCLK), and the signals on the clock bus functions as the counter enabling signal for these timer counters.

The following table shows the signals that can be input on the clock bus.

Bit Number of Clock Bus	Input Signals
5	Output signal from timer B (angle clock) or external input clock B (TCLKB)
4	External input clock A (TCLKA)
3	Output signal from prescaler 3
2	Output signal from prescaler 2
1	Output signal from prescaler 1
0	Output signal from prescaler 0

30.3.2 Registers Related to Common Controller

30.3.2.1 ATUENR — ATU Master Enable Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	TGE	TFE	TEE	TDE	TCE	TBE	TAE	PSCE
Value after reset	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.14 ATUENR Register Contents

Bit Position	Bit Name	Function
7	TGE	Timer G enable bit 0: Timer G counter operation disabled 1: Timer G counter operation enabled
6	TFE	Timer F enable bit 0: Timer F counter operation disabled 1: Timer F counter operation enabled
5	TEE	Timer E enable bit 0: Timer E counter operation disabled 1: Timer E counter operation enabled
4	TDE	Timer D enable bit 0: Timer D counter operation disabled 1: Timer D counter operation enabled
3	TCE	Timer C enable bit 0: Timer C counter operation disabled 1: Timer C counter operation enabled
2	TBE	Timer B enable bit 0: Timer B counter operation disabled 1: Timer B counter operation enabled
1	TAE	Timer A enable bit 0: Timer A counter operation disabled 1: Timer A counter operation enabled
0	PSCE	Prescaler enable bit 0: Prescaler counter operation disabled 1: Prescaler counter operation enabled

ATUENR is an 8-bit readable/writable register. This register is used to enable and disable the prescalers and the individual blocks in ATU-V. Setting an enable bit to 1 enables the corresponding block. Clearing the bit to 0 disables the corresponding block. Even when the enable bit is cleared to 0, the registers of these blocks remain accessible.

Timers can be synchronized by simultaneously setting multiple bits to 1.

Timer A Enable Bit (TAE)

Enables and disables counter operation of timer A.

When this bit is set to “1”, the timer A counter starts working. When this bit is cleared to “0”, the timer A counter stops working. When the counter is disabled, its value is retained. When this bit is set to “1” again, the counter resumes counting from the retained value.

Timer B Enable Bit (TBE)

Enables and disables counter operation of timer B.

When this bit is set to “1”, the timer B counter starts working. When this bit is cleared to “0”, the timer B counter stops working. When the counter is disabled, its value is retained. When this bit is set to “1” again, the counter resumes counting from the retained value.

Timer C Enable Bit (TCE)

Enables and disables counter operation of timer C.

When this bit is set to “1”, the timer C counter starts working. When this bit is cleared to “0”, the timer C counter stops working. When the counter is disabled, its value is retained. When this bit is set to “1” again, the counter resumes counting from the retained value.

Note, however, that the subblock counter does not start working when the TCE bit is set of the corresponding bit in the timer C start register is not set to “1”.

Timer D Enable Bit (TDE)

Enables and disables counter operation of timer D.

When this bit is set to “1”, the timer D counter starts working. When this bit is cleared to “0”, the timer D counter stops working. When the counter is disabled, its value is retained. When this bit is set to “1” again, the counter resumes counting from the retained value.

Note, however, that the subblock counter does not start working when the TDE bit is set of the corresponding bit in the timer D start register is not set to “1”.

Timer E Enable Bit (TEE)

Enables and disables counter operation of timer E.

When this bit is set to “1”, the timer E counter starts working. When this bit is cleared to “0”, the timer E counter stops working. When the counter is disabled, its value is retained. When this bit is set to “1” again, the counter resumes counting from the retained value.

Note, however, that the subblock counter does not start working when the TEE bit is set of the corresponding bit in the timer E start register is not set to “1”.

Timer F Enable Bit (TFE)

Enables and disables counter operation of timer F.

When this bit is set to “1”, the timer F counter starts working. When this bit is cleared to “0”, the timer F counter stops working. When the counter is disabled, its value is retained. When this bit is set to “1” again, the counter resumes counting from the retained value.

Note, however, that the subblock counter does not start working when the TFE bit is set of the corresponding bit in the timer F start register is not set to “1”.

Timer G Enable Bit (TGE)

Enables and disables counter operation of timer G.

When this bit is set to “1”, the timer G counter starts working. When this bit is cleared to “0”, the timer G counter stops working. When the counter is disabled, its value is retained. When this bit is set to “1” again, the counter resumes counting from the retained value.

Note, however, that the subblock counter does not start working when the TGE bit is set of the corresponding bit in the timer G start register is not set to “1”.

Prescaler Enable Bit (PSCE)

Enables and disables counter operation of the prescaler.

When this bit is set to “1”, the prescaler counter starts working. When this bit is cleared to “0”, the prescaler counter stops working. When the counter is disabled, its value is retained. When this bit is set to “1” again, the counter resumes counting from the retained value.

CAUTION

Timer operation setting should be done before starting.

Also, be sure to observe the cautions when using registers to be operated after startup.

30.3.2.2 CBCNT — Clock Bus Control Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	CB4EG[1:0]		—	CB5SEL	CB5EG[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 30.15 CBCNT Register Contents

Bit Position	Bit Name	Function
7, 6	—	These bits are not used. Fix these bits to 0.
5, 4	CB4EG[1:0]	Clock bus 4 edge select bits These bits specify the edge sense for external input clock (TCLKA) to be output on signal line 4 of the clock bus. 00: Neither edge of the external clock is sensed. 01: Rising edges of the external clock are sensed. 10: Falling edges of the external clock are sensed. 11: Both rising and falling edges of the external clock are sensed.
3	—	These bits are not used. Fix these bits to 0.
2	CB5SEL	Clock bus 5 source select bit This bit specifies the clock to be output on signal line 5 of the clock bus. 0: External clock B (TCLKB) 1: Angle clock output by timer B
1, 0	CB5EG[1:0]	Clock bus 5 edge select bits These bits specify the edge sense for external input clock to be output on signal line 5 of the clock bus when CB5S is cleared to "0". 00: Neither edge of the external clock is sensed. 01: Rising edges of the external clock are sensed. 10: Falling edges of the external clock are sensed. 11: Both rising and falling edges of the external clock are sensed.

CBCNT is an 8-bit readable/writable register that selects the source of the clock signal to be supplied on the signal lines of the clock bus and the effective edge of the external clock signal.

Clock Bus 4 Edge Select Bits (CB4EG[1:0])

These bits select the edge sense for externally input clock A (TCLKA). The clock signal is output on signal line 4 of the clock bus. Counters for which signal line 4 of the clock bus has been selected as the source for counting count on the edge selected by these bits.

Clock Bus 5 Source Select Bit (CB5SEL)

This bit selects the source of the clock to be output on signal line 5 of the clock bus.

Clock Bus 5 Edge Select Bits (CB5EG[1:0])

These bits select the edge sense for external input clock B (TCLKB). The clock signal is output on signal line 5 of the clock bus. Counters for which signal line 5 of the clock bus has been selected as the source for counting count on the edge selected by these bits.

The setting of these bits is only valid when the TCLKB signal is selected as the source for line 5 of the clock bus. When the angle clock is selected as the source for line 5 of the clock bus, the setting of these bits is ignored.

30.3.2.3 NCMR — Noise Cancellation Mode Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NCCSEL	—	—	NCMSEL	—	NCMF	NCMC	NCMA
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R/W	R	R/W	R/W	R/W

Table 30.16 NCMR Register Contents

Bit Position	Bit Name	Function
7	NCCSEL	Noise canceler counter clock select bit This bit selects the clock for counting by the noise cancelers. 0: Internal peripheral clock (PCLK) divided by 128 is used as the counter clock 1: Internal peripheral clock (PCLK) is used as the counter clock
6, 5	—	These bits are not used. Fix these bits to 0.
4	NCMSEL	Noise cancellation mode select bit This bit selects minimum time-at-level cancellation mode or level accumulation cancellation mode. 0: Minimum time-at-level cancellation mode 1: Level accumulation cancellation mode
3	—	This bit is not used. Fix this bit to 0.
2	NCMF	Timer F noise cancellation mode bit This bit specifies the operating mode of the timer F noise canceler. 0: Premature-transition cancellation mode 1: Minimum time-at-level cancellation mode (when NCMSEL = 0) 1: Level accumulation cancellation mode (when NCMSEL = 1)
1	NCMC	Timer C noise cancellation mode bit This bit specifies the operating mode of the timer C noise canceler. 0: Premature-transition cancellation mode 1: Minimum time-at-level cancellation mode (when NCMSEL = 0) 1: Level accumulation cancellation mode (when NCMSEL = 1)
0	NCMA	Timer A noise cancellation mode bit This bit specifies the operating mode of the timer A noise canceler. 0: Premature-transition cancellation mode 1: Minimum time-at-level cancellation mode (when NCMSEL = 0) 1: Level accumulation cancellation mode (when NCMSEL = 1)

NCMR is an 8-bit readable/writable register that selects the operating mode and clock to drive the counter for of the noise canceler in each of timers A, C, and F.

In premature-transition cancellation mode, after an input signal level change is detected, any input signal level change in the specified period is ignored. This mode regards a signal level change in the specified period as noise after the first level change is detected.

In minimum time-at-level cancellation mode, after an input signal level change is detected, the first and subsequent level changes are ignored unless the input signal level remains the same over the specified period. This mode regards a signal level change that occurs within a shorter period as noise.

In level accumulation cancellation mode, input signal levels are accumulated and the input level is regarded to have reached 0 or 1 when the accumulation result becomes 0 or the specified value.

The period is set by noise canceler registers in each of the applicable blocks (i.e. in timers A, C, and F) and is counted by a noise canceler counter.

To disable noise cancellation mode from premature-transition/minimum time-at-level cancellation mode, do so after the value of the counter (timer A: NCNTA0 to 6; timer C: NCNTC00 to NCNTC03, NCNTC10 to NCNTC13, NCNTC20 to NCNTC23, NCNTC30 to NCNTC33, NCNTC40 to NCNTC43, NCNTC50 to 53, NCNTC60 to 63, NCNTC70 to 73; timer F: NCNTFA0 to 15, NCNTFB0 to 2) held in each timer becomes 0000_H.

Outline of noise canceling operation (example of TIA00 input signal of timer A) is shown in **Figure 30.1** (in premature-transition cancellation mode), **Figure 30.2** (in minimum time-at-level cancellation mode), and **Figure 30.3** (in level accumulation cancellation mode).

Each timer (A, C, and F) detects the edge of a signal that has passed through the noise canceler. **Figure 30.1**, **Figure 30.2**, and **Figure 30.3** show an example of detecting the rising edge of a signal that has passed through the noise canceler.

Noise Canceler Counter Clock Select Bit (NCCSEL)

Selects the clock for counting by the noise cancelers. Either of the internal peripheral clock (PCLK) divided by 128 or the internal peripheral clock can be selected. The default setting is the clock divided by 128. The same counter clock must be used for all timers other than timer A. In the case of timer A, the clock signal on signal line 5 of the clock bus is also available. For setting the counter clock, see the description of the TIOR2A register in timer A.

Noise Cancellation Mode Select Bit (NCMSEL)

Noise cancellation mode when each noise cancellation mode bit (NCMA, NCMC, or NCMF) is set to “1” can be set to minimum time-at-level cancellation mode or level accumulation cancellation mode.

Timer F Noise Cancellation Mode Bit (NCMF)

Selects the noise cancellation mode for timer F. This bit cannot be used to specify the operating mode for each channel of timer F individually.

To specify the operation mode for each channel of timer F individually, set this bit to “0” and then set noise cancellation mode channel registers 1F (NCMCR1F) and 2F (NCMCR2F).

Timer C Noise Cancellation Mode Bit (NCMC)

Selects the noise cancellation mode for timer C. This bit cannot be used to specify the operating mode for each channel of timer C individually.

To specify the operation mode for each channel of timer C individually, set this bit to “0” and then set noise cancellation mode channel registers 1C (NCMCR1C) and 2C (NCMCR2C).

Timer A Noise Cancellation Mode Bit (NCMA)

Selects the noise cancellation mode for timer A. This bit cannot be used to specify the operating mode for each channel of timer A individually.

To specify the operation mode for each channel of timer A individually, set this bit to “0” and then set noise cancellation mode channel registers 1A (NCMCR1A) and 2C (NCMCR2A).

The tables below show the truth table for settings that determine the noise cancellation mode.

<<Timer A>>

Settings that Determine Noise Cancellation Modes for Timer A

Channel Enable	Filter Mode		Channel Register		Mode of Cancellation	Unit of Filtering
	TIOR2A.NCEAx	NCMR.NCMA	NCMR.NCMSEL	NCMCR1A.NCM1Ax		
0	—	—	—	—	Filter invalid	—
1	0	(-)	0	(-)	Premature-transition	Individual channel/ All channels
1	0	(-)	1	0	Minimum time-at-level	Individual channel
1	0	(-)	1	1	Level accumulation	Individual channel
1	1	(0)	—	—	Minimum time-at-level	All channels
1	1	1	—	—	Level accumulation	All channels

Note 1: x = SB_A-1, 0**Note 2:** The noise canceller enable bit (NCEAx) can be set for each channel.**Methods of Setting Noise Cancellation for Timer A when the Unit of Filtering is Individual Channel (precondition: channel enable = 1)**

- Settings for individual channels:

After setting the noise cancellation mode bit (NCMA) in the noise cancellation mode register (NCMR) to “0”, set any noise cancellation mode channel setting bit (NCM1Ax) in noise cancellation mode channel register 1A (NCMCR1A) of timer A to “1”.

- Settings for all channels:

Set the noise cancellation mode bit (NCMA) in the noise cancellation mode register (NCMR) to “1”, or set all noise cancellation mode channel setting bits (NCM1Ax) in noise cancellation mode channel register 1A (NCMCR1A) to “0”.

<<Timer C>>

Settings that Determine Noise Cancellation Modes for Timer C

Channel Enable	Filter Mode		Channel Register		Mode of Cancellation	Unit of Filtering
	NCCRCx.NCECxy	NCMR.NCMC	NCMR.NCMSEL	NCMCR1C.NCM1Cx		
0	—	—	—	—	Filter invalid	—
1	0	(-)	0	(-)	Premature-transition	Individual subblock/ All subblocks
1	0	(-)	1	0	Minimum time-at-level	Individual subblock
1	0	(-)	1	1	Level accumulation	Individual subblock
1	1	(0)	—	—	Minimum time-at-level	All subblocks
1	1	1	—	—	Level accumulation	All subblocks

Note 1: x = SB_C-1, 0 (subblock), y = 3 to 0 (channel)**Note 2:** The noise canceller enable bit (NCECxy) can be set for each channel.

Methods of Setting Noise Cancellation for Timer C when the Unit of Filtering is Individual Subblock (precondition: channel enable = 1)

Only timer C is set in units of subblock. Each subblock of timer C consists of four channels.

- Settings for individual channels:

After setting the noise cancellation mode bit (NCMC) in the noise cancellation mode register (NCMR) to “0”, set any noise cancellation mode channel setting bit (NCM1Cx) in noise cancellation mode channel register 1C (NCMCR1C) of timer C to “1”.

- Settings for all channels:

Set the noise cancellation mode bit (NCMC) in the noise cancellation mode register (NCMR) to “1”, or set all noise cancellation mode channel setting bits (NCM1Cx) in noise cancellation mode channel register 1C (NCMCR1C) to “0”.

<<Timer F>>

Settings that Determine Noise Cancellation Modes for Timer F

Channel Enable TCCRF. NCEF _x	Filter Mode		Channel Register		Mode of Cancellation	Unit of Filtering
	NCMR. NCMF	NCMR. NCMSEL	NCMCR1F. NCM1F _x	NCMCR2F. NCM2F _x		
0	—	—	—	—	Filter invalid	—
1	0	(-)	0	(-)	Premature-transition	Individual channel/ All channels
1	0	(-)	1	0	Minimum time-at-level	Individual channel
1	0	(-)	1	1	Level accumulation	Individual channel
1	1	(0)	—	—	Minimum time-at-level	All channels
1	1	1	—	—	Level accumulation	All channels

Note 1: $x = SB_F - 1, 0$

Note 2: The noise canceller enable bit (NCEF_x) can be set for each channel.

Methods of Setting Noise Cancellation for Timer F when the Unit of Filtering is Individual Channel (precondition: channel enable = 1)

- Settings for individual channels:

After setting the noise cancellation mode bit (NCMF) in the noise cancellation mode register (NCMR) to “0”, set any noise cancellation mode channel setting bit (NCM1F_x) in noise cancellation mode channel register 1F (NCMCR1F) of timer F to “1”.

- Settings for all channels:

Set the noise cancellation mode bit (NCMF) in the noise cancellation mode register (NCMR) to “1”, or set all noise cancellation mode channel setting bits (NCM1F_x) in noise cancellation mode channel register 1F (NCMCR1F) to “0”.

The figures below show an overview of the timing of operations for noise cancellation in the respective modes. Detailed descriptions and timing charts of operations in the respective modes follow that. Since these figures and descriptions are provided for the case of timer A, be sure to replace the registers, bits, and input pin for timers C and F by referring to the following table.

List of Registers, Bits, and Input Pins of Timers Related to Noise Canceller

	Timer A	Timer C	Timer F
Common units			
Noise cancel mode register (NCMR register bit name →)	NCMA	NCMC	NCMF
	NCMSEL	NCMSEL	NCMSEL
Each timer			
Noise cancellation mode channel register (register name →)	NCMCR1A	NCMCR1C	NCMCR1F
	NCMCR2A	NCMCR2C	NCMCR2F
Noise canceller enable bit (register name, bit name →)	TIOR2A. NCEAx	NCCRCx. NCECxy	NCCRF. NCEFx
Noise cancellation counter (register name →)	NCNTAx	NCNTCxy	NCNTFxA NCNTFBx
Noise cancellation register (register name →)	NCRAx	NCRCxy	NCRFAx NCRFBx
Input pin	TIAx	TIOCxy	TIFxA TIFxB

Note: x indicates the number of subblocks of each timer; y indicates the number of channels of each timer.

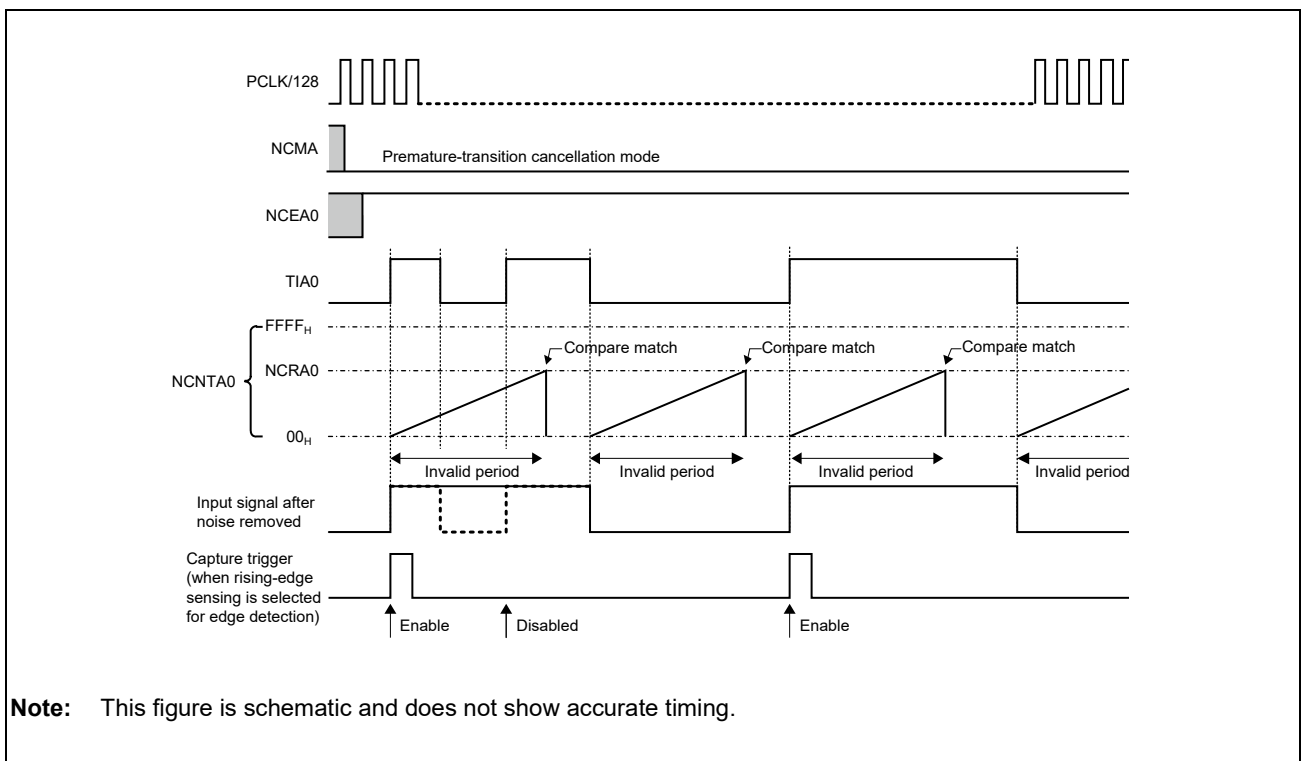


Figure 30.1 Outline of Noise Canceling Operation in Premature-Transition Cancellation Mode

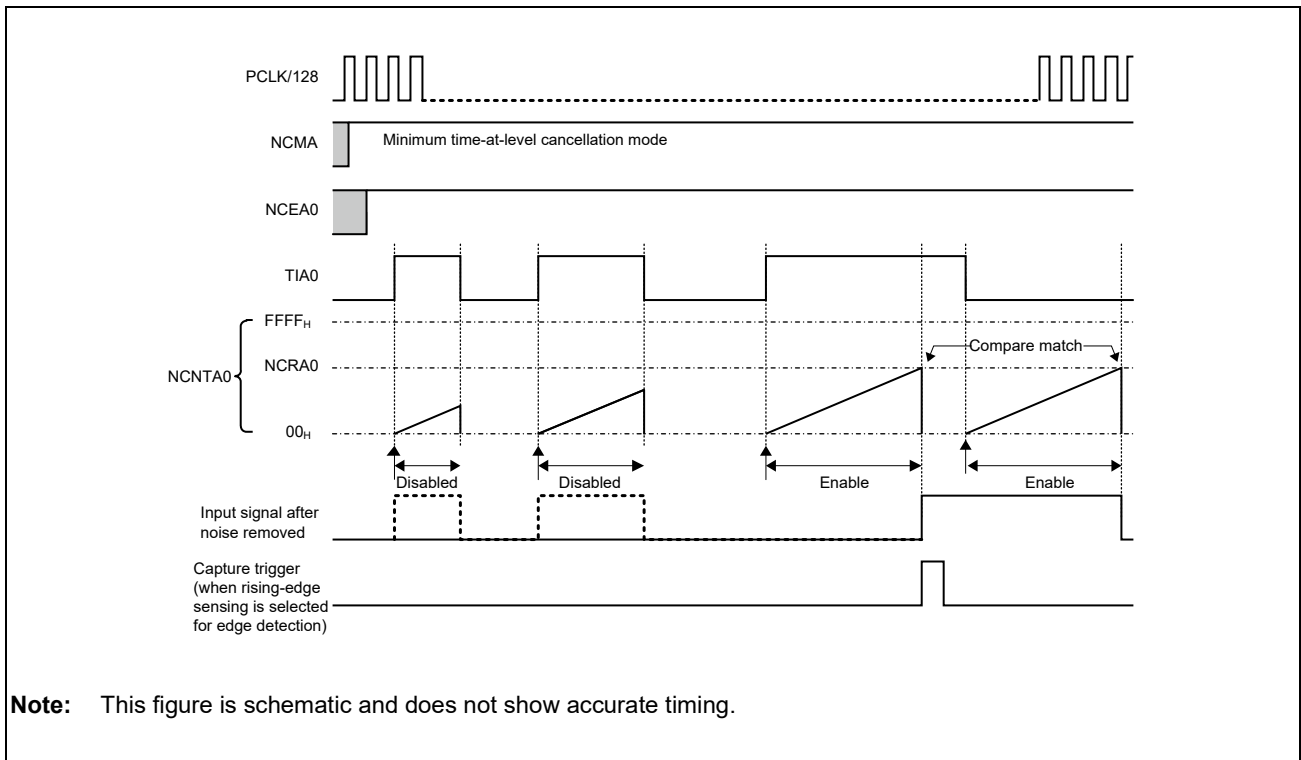


Figure 30.2 Outline of Noise Canceling Operation in Minimum Time-at-Level Cancellation Mode

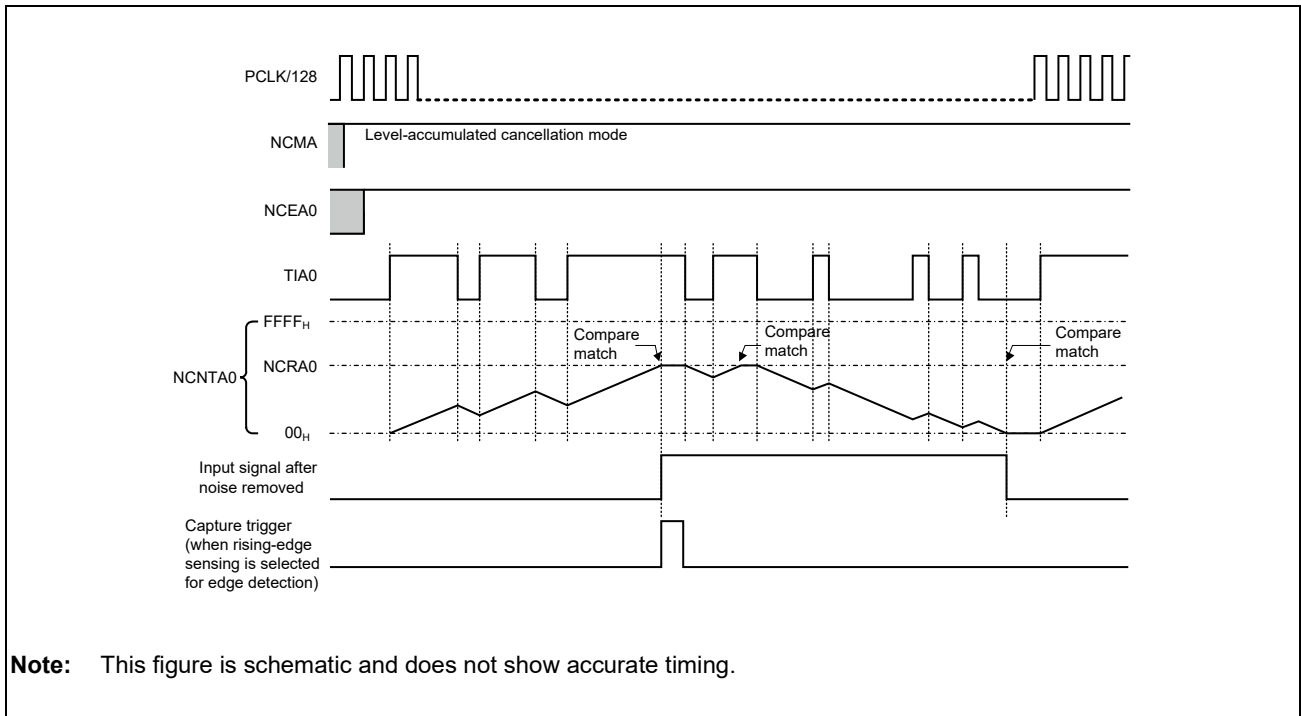


Figure 30.3 Outline of Noise Canceling Operation in Level Accumulation Cancellation Mode

30.3.2.4 Description of Operations for Noise Cancellation

The noise canceller has three operating modes: minimum time-at-level cancellation mode, premature-transition cancellation mode, and level-accumulative cancellation mode. Either of these modes can be selected by setting the noise cancellation mode register (NCMR) of the common controller unit, the noise cancellation mode channel register 1A (NCMCR1A) of timer A, and noise cancellation mode channel register 2A (NCMCR2A) of timer A.

Figure 30.4 shows an example of operations for noise cancellation in premature-transition cancellation mode and **Figure 30.6** shows an example of operations for noise cancellation in minimum time-at-level cancellation mode. These figures show an example in which TIA00 input is used and the edge deletion level is the rising.

In premature-transition cancellation mode, the noise cancellation counter (NCNTAx) starts counting in response to a change in the level of the input signal as a trigger. At the same time as counting starts, changes in the level of the input signal are output as the signal after noise cancellation.

Counting continues until the count value matches the value set in the noise cancellation register (NCRAx). All changes in the level of the input signal are ignored during the counting and are not output as the signal after noise cancellation.

When the count value matches the value set in the noise cancellation register, the level of the input signal at that time is output as the signal after noise cancellation. Note that the signal after noise cancellation changes upon compare match if the input level at the start of counting (after transition) differs from that upon compare match. **Figure 30.5** shows difference in noise cancellation processing between two input waveforms in premature-transition cancellation mode.

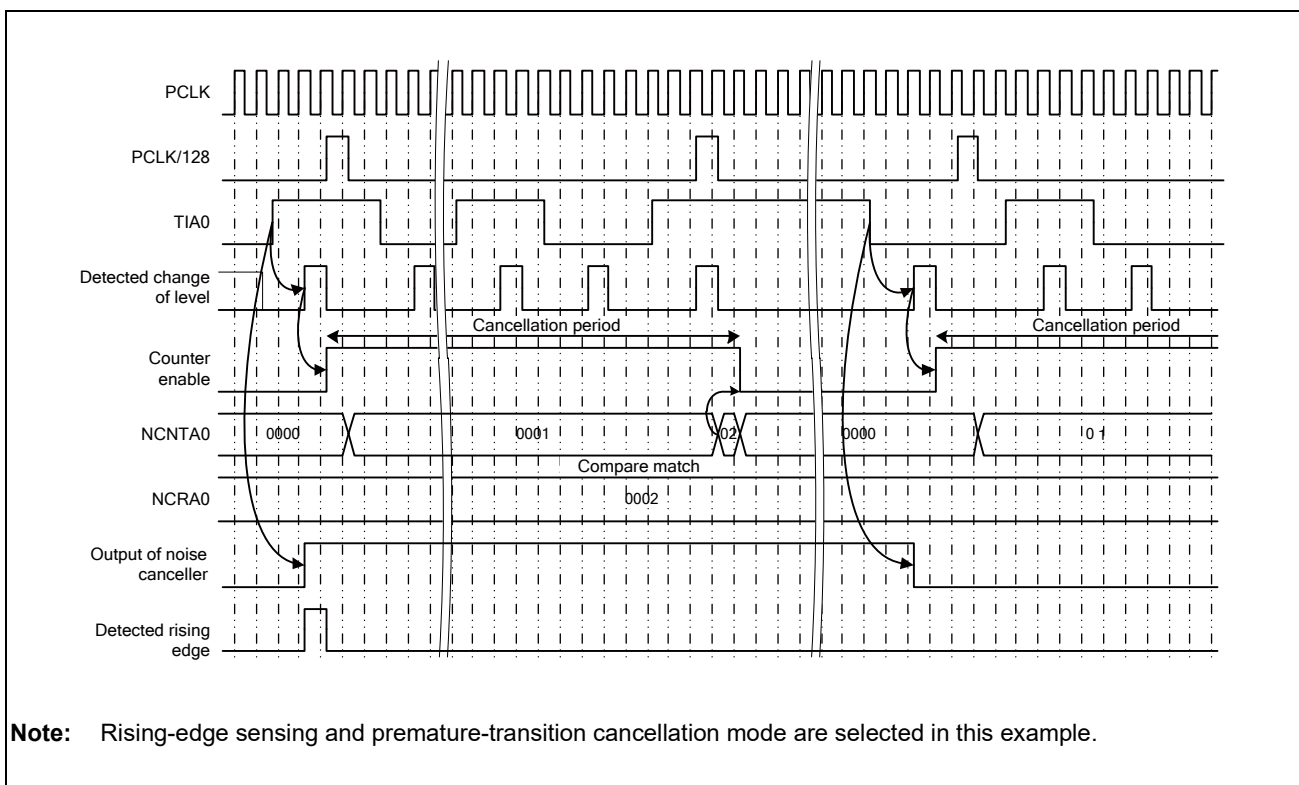


Figure 30.4 Example of Noise Cancellation in Premature-Transition Cancellation Mode

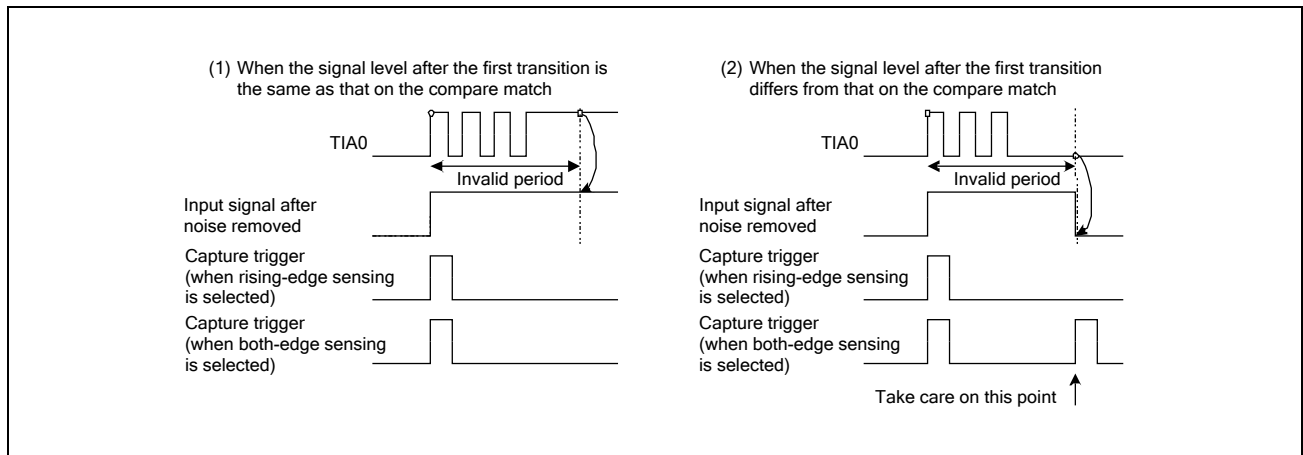


Figure 30.5 Example of Noise Cancellation for Two Types of Waveforms
(in Premature-Transition Cancellation Mode)

In minimum time-at-level cancellation mode, the noise cancellation counter (NCNTAx) starts counting in response to a change in the level of the input signal as a trigger. Counting continues until the count value matches the value set in the noise cancellation register (NCRAx) or until the level of the input signal changes.

When the count value matches the value set in the noise cancellation register, any change in the level of the input signal after the start of counting is output as the signal after noise cancellation. If the level of the input signal changes before the count value and the value of the noise cancellation register match, the changes in the level from that when counting started are deemed to be noise, and are not output as the signal after noise cancellation.

Figure 30.7 shows an example of operations for noise cancellation in level accumulation cancellation mode. This figure shows an example in which TIA00 input is used and the edge deletion level is the rising.

In level accumulation cancellation mode, noise cancellation counter (NCNTAx) counts up or down according to the level of the input signal. If the counter is counting up, the counting operation continues until the count value matches the value set in the noise cancellation register (NCNTAx). If the counter is counting down, the counting operation continues until the count value matches 0000_H.

The noise canceller output is updated upon compare match. If the count value matches the NCRAx value during counting-up operation, the noise canceller output value is updated to “1”. If the counter value matches 0000_H during counting-down operation, the noise canceller output value is updated to “0”.

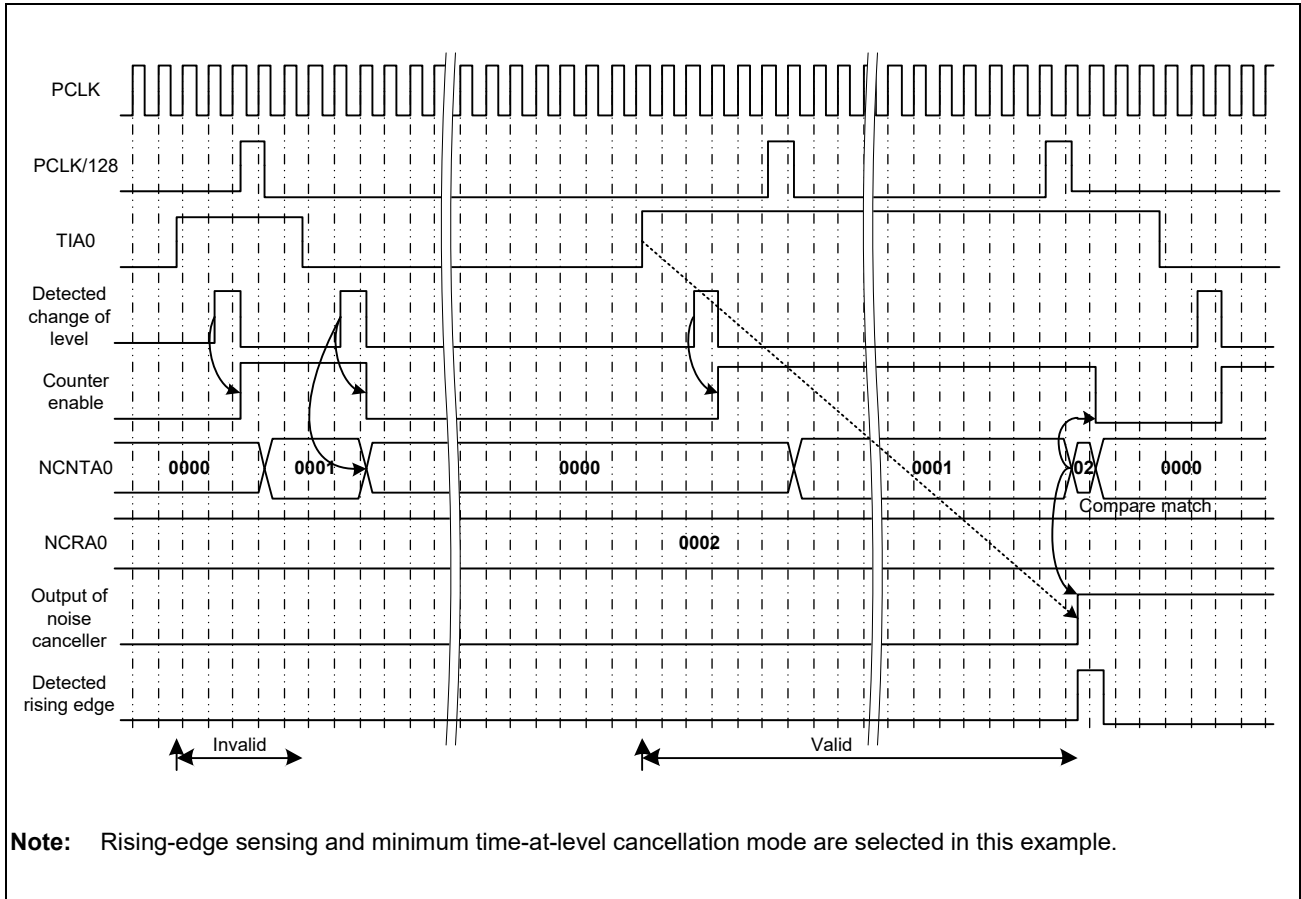


Figure 30.6 Example of Noise Cancellation in Minimum Time-at-Level Cancellation Mode

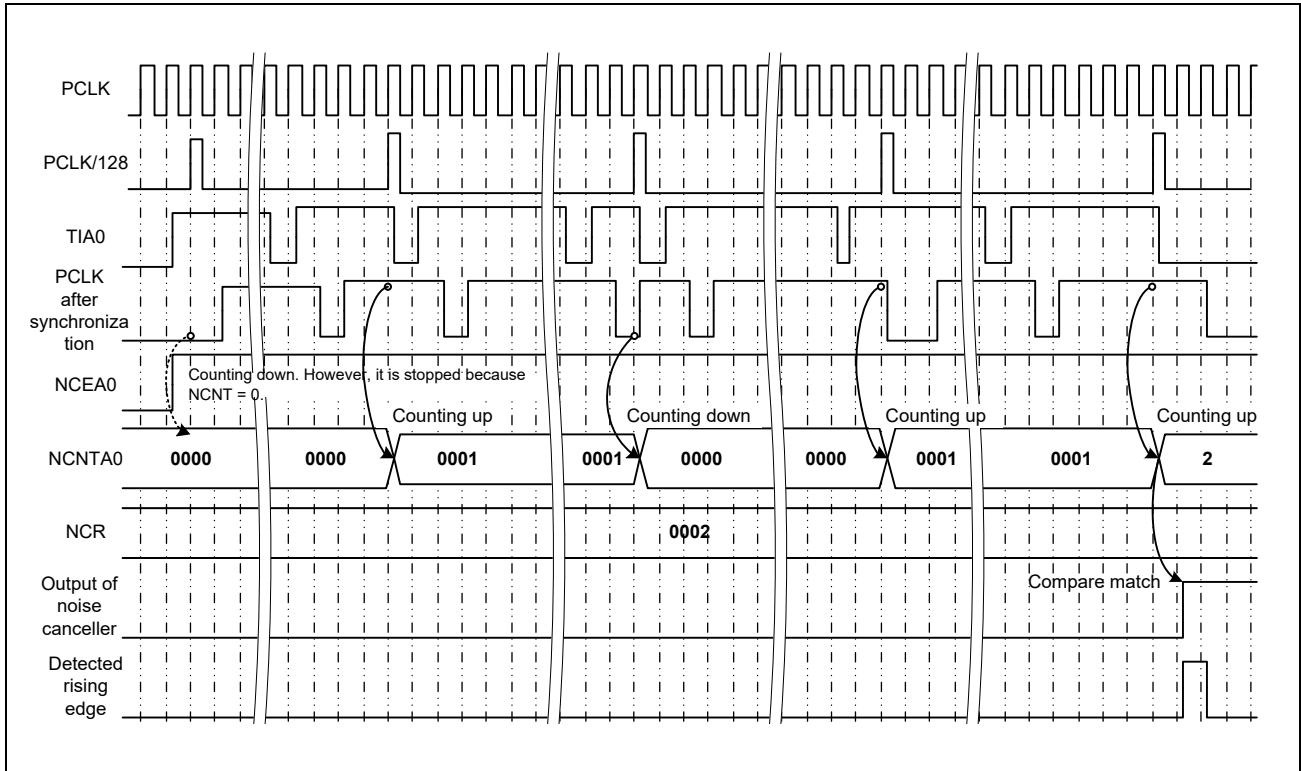


Figure 30.7 Example of Noise Cancellation in Level Accumulation Cancellation Mode

30.4 Prescaler

30.4.1 Overview of Operation

ATU-V includes general prescalers of four channels and a prescaler for the noise-canceler clock.

Each general prescaler are implemented as 10-bit down-counters, in which the prescaled clock signals are generated by frequency-dividing the internal peripheral clock (PCLK) by N ($1 \leq N \leq 1024$). The generated clock signals are supplied to the individual timers via the clock bus. The prescalers for each channel operate independently.

The prescaler for the noise-canceler clock is implemented as a 7-bit down-counter. It generates a clock signal by frequency-dividing the peripheral clock (PCLK) by 128. The clock signal thus generated is supplied to the noise cancelers of timers A, C, and F.

Whether to generate a noise canceler clock by frequency-dividing the peripheral clock frequency by 1 or 128 can be selected by the NCCSEL bit in NCMR of the common controller. No other division ratios are available.

The down-counters of the prescalers are initialized to 0000_H after reset.

30.4.2 Registers Related to Prescaler

30.4.2.1 PSCRx — Prescaler Registers x

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PSCx[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.17 PSCRx Register Contents

Bit Position	Bit Name	Function
15 to 10	—	This bit is not used. Fix this bit to 0.
9 to 0	PSCx[9:0]	Division ratio bits These bits specify the division ratio for the corresponding prescaler.

Note: x = 0 to 3

Prescaler registers are 16-bit readable/writable registers, each of which holds a division ratio for one of the four prescalers.

When the value in prescaler register x is changed, the prescaler updates the value on its underflow. The setting range is from 000_H to 3FF_H.

The division ratio is obtained from the following expression.

$$\text{Division ratio of prescaler} = \frac{1}{\text{PSCx}[9:0] + 1} \quad (\text{setting range: } 1/1 \text{ to } 1/1024)$$

A duty ratio of 50% is not guaranteed for the clock output by the prescaler. The level of the PCLK is high during one-cycle period only, and is low during the other periods.

30.4.3 Details of Operation

30.4.3.1 Starting Prescalers

The prescalers start operating when the PSCE bit in the ATU master enable register is set to 1 and generates a clock with a frequency given by the division ratio in the PSCx bits. While a prescaler is operating, the high level is output for one cycle of the PCLK clock each time the corresponding prescaler counter underflows.

When the setting in the PSCx bits is changed during operation, the division ratio of the output clock is updated on the first subsequent counter underflow.

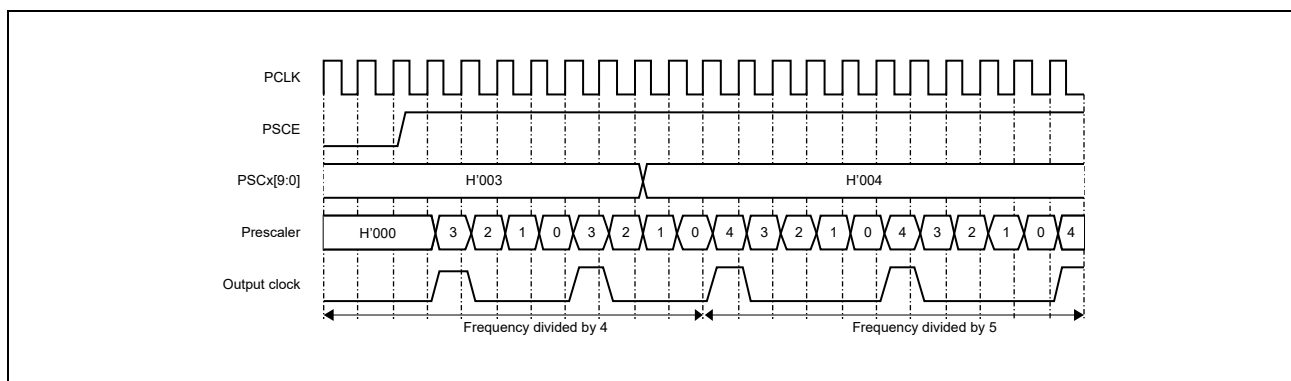


Figure 30.8 Starting Prescaler

30.4.3.2 Stopping and Restarting Operation

The prescaler stops operating when the PSCE bit in the ATU master enable register is cleared to “0”. The clock signal retains the low level while the prescaler is stopped.

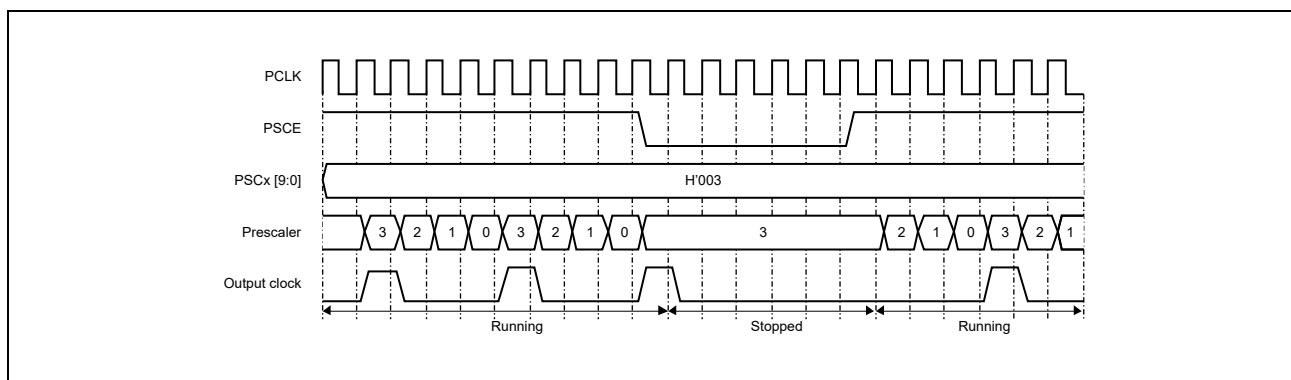


Figure 30.9 Stopping Prescaler

30.5 Timer A

30.5.1 Overview of Operation

Timer A integrates a 32-bit free-running counter A (TCNTA) and SB_A 32-bit input capture registers (ICRA0 to ICRASB_A-1). TCNTA is a free-running up counter. An interrupt request can be output when the counter overflows.

SB_A input capture registers A0 to ASB_A-1 (ICRA0 to ICRASB_A-1) capture the value of free-running counter A (TCNTA) on an assertion of the corresponding external input signal (TIA00 to TIA0SB_A-1). The rising edge, falling edge, or both edges can be selected as the trigger for capture. The detection edge is selected by timer I/O control register 1 (TIOR1A). An interrupt request can be issued at the same timing as capturing.

Noise on the external input signals can be removed by the noise canceler. The pin level of these external signal input pins (TIA00 to TIA0SB_A-1) after noise cancellation can be read. Input pin (TIA00 to TIA0SB_A-1) signals after noise cancellation can be output as input signals to timer F. Some input pin signals can be output to timer B, C or D as event signals after noise has been removed and their edges have been extracted. One of TIA00 to TIA0SB_A-1 can be output to timer B as an event signal (event output 1_0, 1_1, or 1_2). SB_A event signals TIA00 to TIA0SB_A-1 are output (event outputs 1B to 1I) to timer B to be used as the capture trigger of the event counter B1 (TCNTB1). One of SB_A event signals TIA00 to TIA0SB_A-1 is output (event output 2A and 2B) to timer D0 to DSB_D-1 respectively to be used as the capture trigger of timer offset base registers OSBRD0 to OSBRDSB_D-1. The signals of these event outputs 1_0, 1_1, 1_2, and event outputs 2A and 2B can be used as the input capture trigger of timer C.

Figure 30.10 shows a block diagram of timer A.

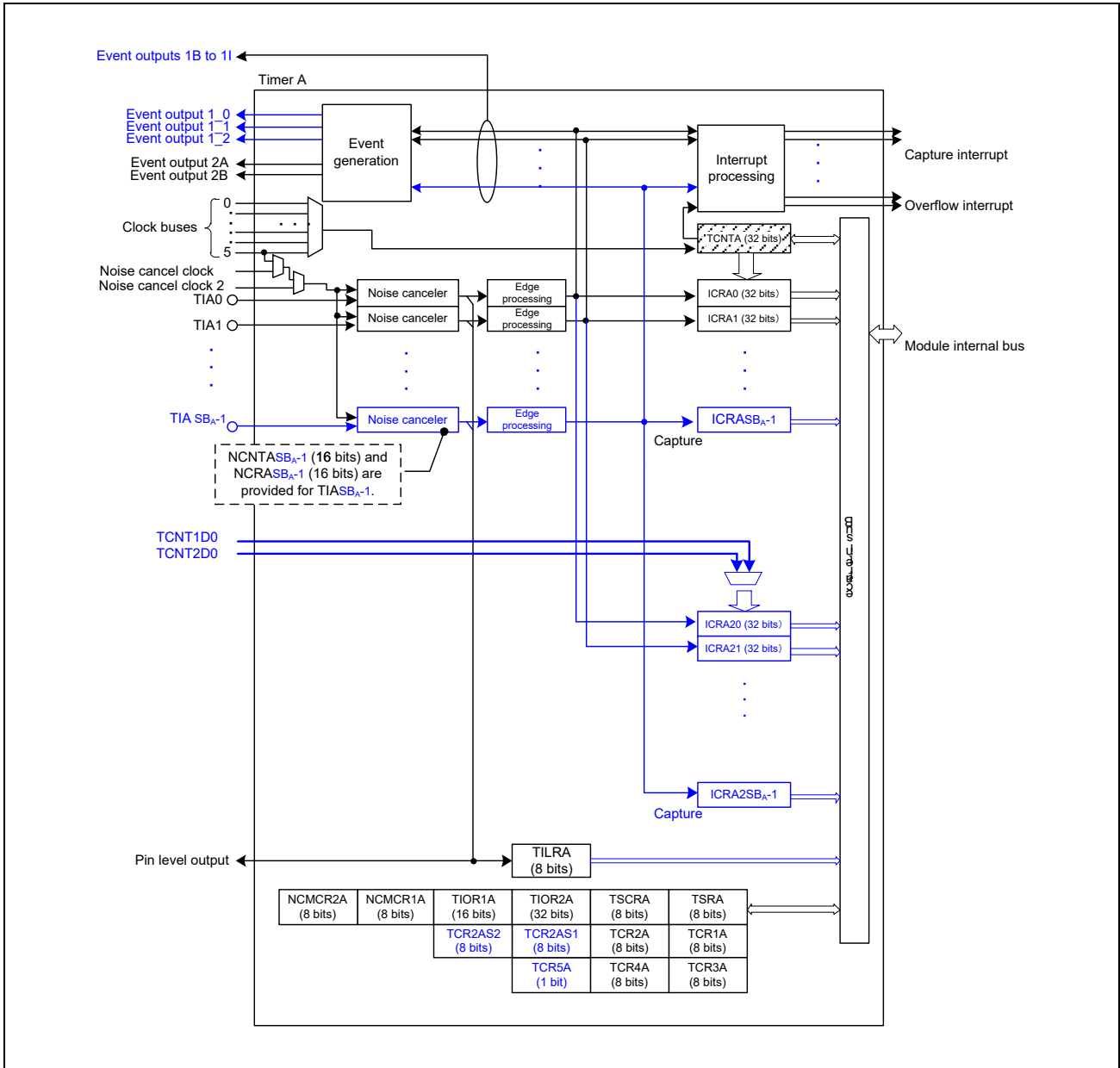


Figure 30.10 Block Diagram of Timer A Configuration

30.5.2 Registers Related to Timer A

30.5.2.1 TCR1A — Timer Control Register 1A

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	EVOSEL2A	EVOSEL2B	EVOSEL1[2:0]			CKSELA[2:0]		
Value after reset	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.18 TCR1A Register Contents

Bit Position	Bit Name	Function
7	EVOSEL2A	Event output 2A select 0: TIA01 is selected as event output 2A. 1: TIA02 is selected as event output 2A.
6	EVOSEL2B	Event output 2B select 0: TIA01 is selected as event output 2B. 1: TIA02 is selected as event output 2B.
5 to 3	EVOSEL1	Event output 1_0 select 000: No signal is selected as event output 1. 001: TIA00 is selected as event output 1_0 010: TIA01 is selected as event output 1_0 011: TIA01 and TIA00 input edges are output together. 100: TIA02 is selected as event output 1_0 101: TIA02 and TIA00 input edges are output together. 110: TIA02 and TIA01 input edges are output together. 111: TIA02, TIA01, and TIA00 input edges are output together.
2 to 0	CKSELA	Clock select A 000: Select clock bus 0 (prescaler 0) 001: Select clock bus 1 (prescaler 1) 010: Select clock bus 2 (prescaler 2) 011: Select clock bus 3 (prescaler 3) 100: Select clock bus 4 (TCLKA) 101: Select clock bus 5 (TCLKB or multiplied-and corrected clock) 110: Setting prohibited 111: Setting prohibited

TCR1A is an 8-bit readable/writable register that sets the event output generated from external input signals (TIA00 to TIA02) and the counter clock.

TCR1A is initialized to 00_H after reset.

Event Output 2A Select (EVOSEL2A)

Selects either of the externally input signals TIA01 and TIA02 (after the noise cancellation and edge detection) to be output as the event output 2A signal. Timers C0 to CSB_C-1 and timers D0 to DSB_D-1 can use the event output 2A signal as a capture trigger.

Event Output 2B Select (EVOSEL2B)

Selects either of the externally input signals TIA01 and TIA02 (after the noise cancellation and edge detection) to be output as the event output 2B signal. Timers C0 to CSB_C-1 and timers D0 to DSB_D-1 can use the event output 2B signal as a capture trigger.

Event Output 1_0 Select (EVOSEL1)

Selects either of the externally input signals TIA02 to TIA00 (after the noise cancellation and edge detection) to be output to timers B and C as the event output 1_0 signal. Furthermore, input edge combinations of TIA00 and TIA01, TIA00 and TIA02, TIA01 and TIA02, and TIA00, TIA01, and TIA02 can be output together to timer B as the event output 1_0.

Clock Select A (CKSELA)

These bits select the signal on one of clock-bus lines 0 to 5 as the clock signal for counting. The signals on lines 0 to 3 have been frequency-divided by prescalers 0 to 3, respectively. Clock-bus line 4 supplies external clock input A (TCLKA). Clock-bus line 5 supplies external input clock B (TCLKB) or the multiplied-and-corrected clock output by timer B according to the setting of the clock bus control register.

Be sure to stop timer A before selecting the counter clock.

CAUTION

The edge of an external input clock is detected before it is output on a clock bus. When using external input clock A or B, select the edge to be detected by setting the edge select bit in the clock bus control register (CBCNT).

30.5.2.2 TCR2A — Timer Control Register 2A

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EVOSELE1E	—	—	—	—	—	—	—	EVOSELE1[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.19 TCR2A Register Contents

Bit Position	Bit Name	Function
15	EVOSELE1E	Event output 1_0 extension select enable 0: Event output 1_0 extension selection is disabled. 1: Event output 1_0 extension selection is enabled.
14-SB _A	—	These bits are not used. Fixe these bits to “0”.
bSB _A -1 to 0	EVOSELE1	Event output 1_0 extension select register These bits select the combination of inputs (TIA0x) to be output as an event. 0: TIA0x input edge is not output. 1: TIA0x input edge is output. Example: For b0 b0 = 0: TIA00 input edge is not output. b0 = 1: TIA00 input edge is output.

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1.

Event Output 1_0 Extension Select Enable (EVOSELE1E)

Event output 1_0 extension select enable (EVOSELE1E) is an 8-bit/16-bit readable/writable register to enable or disable the event output 1_0 extension selection.

EVOSELE1E is initialized to 0_B after reset.

Event Output 1_0 Extension Select Register (EVOSELE1)

Event output 1_0 extension select register (EVOSELE1) is an 8-bit/16-bit readable/writable register to set event output 1_0 of external input edge TIA0x when EVOSELE1E is enabled.

EVOSELE1 is initialized to 00_H after reset.

Any external input signal TIA0x (after noise cancellation and edge detection) or combination of the input edges can be output to timer B or timer C as event 1_0 according to the setting of this register.

When this function is enabled, event output selection specified by TCR1A is disabled.

30.5.2.3 TCR2AS1 — Timer Control Register 2AS1

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	EVOSELE1S1[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.20 TCR2AS1 Register Contents

Bit Position	Bit Name	Function
SB _A -1 to 0	EVOSELE1S1	Event output 1_1 extension select register These bits select the combination of inputs (TIA0x) to be output as an event. 0: TIA0x input edge is not output. 1: TIA0x input edge is output. Example: For b0 b0 = 0: TIA00 input edge is not output. b0 = 1: TIA00 input edge is output.

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1.

Event Output 1_1 Extension Select Register (EVOSELE1S1)

Event output 1_1 extension select register (EVOSELE1S1) is an 8-bit readable/writable register to set event output 1_1 of external input edge TIA0x.

EVOSELE1S1 is initialized to 00_H after reset.

Any external input signal TIA0x (after noise cancellation and edge detection) or combination of the input edges can be output to timer B as event 1_1 according to the setting of this register.

30.5.2.4 TCR2AS2 — Timer Control Register 2AS2

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	EVOSELE1S2[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.21 TCR2AS2 Register Contents

Bit Position	Bit Name	Function
SB _A -1 to 0	EVOSELE1S2	Event output 1_2 extension select register These bits select the combination of inputs (TIA0x) to be output as an event. 0: TIA0x input edge is not output. 1: TIA0x input edge is output. Example: For b0 b0 = 0: TIA00 input edge is not output. b0 = 1: TIA00 input edge is output.

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1.

Event Output 1_2 Extension Select Register (EVOSELE1S2)

Event output 1_2 extension select register (EVOSELE1S2) is an 8-bit readable/writable register to set event output 1_2 of external input edge TIA0x.

EVOSELE1S2 is initialized to 00_H after reset.

Any external input signal TIA0x (after noise cancellation and edge detection) or combination of the input edges can be output to timer B as event 1_2 according to the setting of this register.

30.5.2.5 TCR3A — Timer Control Register 3A

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EVOSELE2AE	—	—	—	—	—	—	—	EVOSELE2A							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.22 TCR3A Register Contents

Bit Position	Bit Name	Function
15	EVOSELE2AE	Event output 2A extension select enable 0: Event output 2A extension selection is disabled. 1: Event output 2A extension selection is enabled.
14-SB _A	—	These bits are not used. Fixe these bits to “0”.
SB _A -1 to 0	EVOSELE2A	Event output 2A extension select register These bits select the combination of inputs (TIA0x) to be output as an event. 0: TIA0x input edge is not output. 1: TIA0x input edge is output. Example: For b0 b0 = 0: TIA00 input edge is not output. b0 = 1: TIA00 input edge is output.

Note 1: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1.

Event Output 2A Extension Select Enable (EVOSELE2AE)

Event output 2A extension select enable (EVOSELE2AE) is an 8-bit/16-bit readable/writable register to enable or disable the event output 2A extension selection.

EVOSELE2AE is initialized to 0_B after reset.

Event Output 2A Extension Select Register (EVOSELE2A)

Event output 2A extension select register (EVOSELE2A) is an 8-bit/16-bit readable/writable register to set event output 2A of external input edge TIA0x when EVOSELE2AE is enabled.

EVOSELE2A is initialized to 00_H after reset.

Any external input signal TIA0x (after noise cancellation and edge detection) or combination of the input edges can be output as event 2A according to the setting of this register. Timers C0 to CSB_C-1 and timers D0 to DSB_D-1 can use this event 2A signal as the capture trigger.

When this function is enabled, event output selection specified by TCR1A is disabled.

30.5.2.6 TCR4A — Timer Control Register 4A

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EVOSELE2BE	—	—	—	—	—	—	—	EVOSELE2B[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.23 TCR4A Register Contents

Bit Position	Bit Name	Function
15	EVOSELE2BE	Event output 2B extension select enable 0: Event output 2B extension selection is disabled. 1: Event output 2B extension selection is enabled.
14-SB _A	—	These bits are not used. Fixe these bits to “0”.
SB _A -1 to 0	EVOSELE2B	Event output 2B extension select register These bits select the combination of inputs (TIA0x) to be output as an event. 0: TIA0x input edge is not output. 1: TIA0x input edge is output. Example: For b0 b0 = 0: TIA00 input edge is not output. b0 = 1: TIA00 input edge is output.

Note 1: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1.

Event Output 2B Extension Select Enable (EVOSELE2BE)

Event output 2B extension select enable (EVOSELE2BE) is an 8-bit/16-bit readable/writable register to enable or disable the event output 2B extension selection.

EVOSELE2BE is initialized to 0_B after reset.

Event Output 2B Extension Select Register (EVOSELE2B)

Event output 2B extension select register (EVOSELE2B) is an 8-bit/16-bit readable/writable register to set event output 2B of external input edge TIA0x when EVOSELE2BE is enabled.

EVOSELE2B is initialized to 00_H after reset.

Any external input signal TIA0x (after noise cancellation and edge detection) or combination of the input edges can be output as event 2B according to the setting of this register. Timers C0 to CSB_C-1 and timers D0 to DSB_D-1 can use this event 2B signal as the capture trigger.

When this function is enabled, event output selection specified by TCR1A is disabled.

30.5.2.7 TCR5A — Timer Control Register 5A

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ICRA2SEL
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 30.24 TCR5A Register Contents

Bit Position	Bit Name	Function
7 to 1	—	These bits are not used. Fixe these bits to "0".
0	ICRA2SEL	ICRA2x register capture data select 0: The value in timer D counter TCNT1D0 is captured. 1: The value in timer D counter TCNT2D0 is captured.

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1.

Input Capture Register 2 Capture Select (ICRA2SEL)

ICRA2SEL selects the value that is captured to input capture register 2Ax when the valid edge of the external signal (TIA0x) selected by TIOR1A is detected, from the value of timer D counter TCNT1D0 or TCNT1D0.

30.5.2.8 NCMCR1A — Noise Cancellation Mode Channel Register 1A

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NCM1A[SB _A -1:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.25 NCMCR1A Register Contents

Bit Position	Bit Name	Function
SB _A -1 to 0	NCM1Ax	Noise cancellation mode bit of channel x Specify the operating mode of noise canceler in channel x. 0: Premature-transition cancellation mode 1: Minimum time-at-level cancellation mode (When NCMA = 0 and NCM2Ax = 0) 1: Level accumulation cancellation mode (When NCMA = 0 and NCM2Ax = 1)

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1.

NCMCR1A is an 8-bit readable/writable register that selects the operation mode of noise canceler for channel unit.

In premature-transition cancellation mode, after an input signal level change is detected, any input signal level change in the specified period is ignored. This mode regards a signal level change in the specified period as noise after the first level change is detected.

In minimum time-at-level cancellation mode, after an input signal level change is detected, the first and subsequent level changes are ignored unless the input signal level remains the same over the specified period. This mode regards a signal level change that occurs within a shorter period as noise.

In level accumulation cancellation mode, input signal levels are accumulated and the input level is regarded to have reached 0 or 1 when the accumulation result becomes 0 or the specified value.

Each period is specified by the noise canceler register of each channel and the noise canceler counter measures time.

To disable noise cancellation mode from premature-transition cancellation mode or minimum time-at-level cancellation mode, do so after the value of the counter (NCNTAx) held in each timer becomes 0000_H.

For details of operations for noise cancellation, see the description of the noise cancellation mode register in **Section 30.3.2, Registers Related to Common Controller**.

Noise Cancellation Mode Bit of Channel x (NCM1Ax)

Specify the operating mode of noise canceler in channel x.

CAUTION

This register is only effective when the NCMA bit of the noise cancellation mode register (NCMR) in the common control unit is "0". Furthermore, when the corresponding channel bit (NCM1Ax) in this register is set to "1", the mode can be set to premature-transition cancellation mode, minimum time-at-level cancellation mode, or level accumulation cancellation mode according to the state of the corresponding channel bit in the noise cancellation mode channel register 2A (NCMCR2A).

30.5.2.9 NCMCR2A — Noise Cancellation Mode Channel Register 2A

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NCM2A[SB _A -1:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.26 NCMCR2A Register Contents

Bit Position	Bit Name	Function
SB _A -1 to 0	NCM2Ax	Noise cancellation mode bit of channel x Specify the operating mode of noise canceler in channel x. 0: Minimum time-at-level cancellation mode (When NCMA = 0 and NCM1Ax = 1) 1: Level accumulation cancellation mode (When NCMA = 0 and NCM1Ax = 1)

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1

NCMCR2A is an 8-bit readable/writable register that selects the operation mode of noise canceler for channel unit.

In minimum time-at-level cancellation mode, after an input signal level change is detected, the first and subsequent level changes are ignored unless the input signal level remains the same over the specified period. This mode regards a signal level change that occurs within a shorter period as noise.

In level accumulation cancellation mode, input signal levels are accumulated and the input level is regarded to have reached 0 or 1 when the accumulation result becomes 0 or the specified value.

Each period is specified by the noise canceler register of each channel and the noise canceler counter measures time.

For details on the operations for noise cancellation, see the description of the noise cancellation mode register in **Section 30.3.2, Registers Related to Common Controller**.

Noise Cancellation Mode Bit 2 of Channel x (NCM2Ax)

Specify the operating mode of noise canceler in channel x.

CAUTION

This register is only effective when the NCMA bit of the noise cancellation mode register (NCMR) in the common control unit is "0" and the noise cancellation mode bit of the corresponding channel (NCM1Ax) in the noise cancellation mode channel register 1A of timer A (NCMCR1A) is "1".

30.5.2.10 TIOR1A — Timer I/O Control Register 1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IOA7[1:0]		IOA6[1:0]		IOA5[1:0]		IOA4[1:0]		IOA3[1:0]		IOA2[1:0]		IOA1[1:0]		IOA0[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.27 TIOR1A Register Contents

Bit Position	Bit Name	Function
SB _A -1 to 0	IOAx	I/O control Ax 00: TIA0x input capturing is prohibited 01: TCNTA is captured in ICRAx at the rising edge of TIA0x. 10: TCNTA is captured in ICRAx at the falling edge of TIA0x. 11: TCNTA is captured in ICRAx at both the rising and falling edges of TIA0x.

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1

Timer I/O control register 1 (TIOR1A) is used to select the edge of event outputs 2A, 2B, 1_0, and 1B to 1H.

This is a 16-bit readable/writable register that sets the edge of external input (TIA0x) to be detected.

TIOR1A is initialized to 0000_H after reset.

I/O Control Ax (IOAx)

These bits select the edge of external inputs (TIA00 to TIA0SB_A-1) that is to be detected as a trigger for input capture. When these bits are set to “00”, input capturing is not performed. When the bit is set to “01”, “10” or “11”, the contents of free-running counter A (TCNTA) are transferred to input capture register A0 to ASB_A-1 (ICRA0 to ICRASB_A-1) on detection of the selected edge from one of the external inputs.

Edge detection is synchronized with the PCLK clock. Make sure that the cycle of the external input signal is at least twice the cycle of the PCLK clock. Otherwise, edge detection is not performed correctly.

Edge detection is performed for signals that have passed through the noise canceler. When the noise canceler is disabled, edge detection is performed for the external inputs (TIA00 to TIA0SB_A-1). When the noise canceler is enabled, edge detection is performed for the signals after noise cancellation.

The edge detection signals of TIA00 to TIA0SB_A-1 are directly output to event outputs 1B to 1H. TIA00 corresponds to event output 1B, TIA01 to event output 1C, TIA02 to event output 1D, TIA03 to event output 1E, TIA04 to event output 1F, TIA05 to event output 1G, TIA06 to event output 1H, and TIA07 to event output 1I, respectively.

Furthermore, the signal detected by the valid edge is output to other timer modules by setting the event output 1_0, event output 2A, and event output 2B select bits (EVOSEL1, EVOSEL2A, and EVOSEL2B) in TCR1A, the event output 1_0 extension select bits (EVOSELE1) in TCR2A, the event output 2A extension select bits (EVOSELE2A) in TCR3A, and the event output 2B extension select bits (EVOSELE2B) in TCR4A. (Signals to be output after edge detection are generated in positive logic.)

30.5.2.11 TIOR2A — Timer I/O Control Register 2

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NCKGA [SB _A -1:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NCKA[SB _A -1:0]								NCEA[SB _A -1:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.28 TIOR2A Register Contents

Bit Position	Bit Name	Function
31 to 24	—	These bits are not used. Fixe these bits to "0".
23 to 16	NCKGA _x	Noise canceler clock select G A _x 0: The clock selected by NCKA _x is selected as the count source clock of NCNTA _x . 1: Noise canceler count clock (AGCK1) is selected as the count source clock of NCNTA _x
15 to 8	NCKA _x	Noise canceler clock select A _x 0: Noise canceler count clock (PCLK or PCLK/128) is selected as the count source clock of NCNTA _x . 1: Clock-bus line 5 is selected as the count source clock of NCNTA _x .
7 to 0	NCEA _x	Noise canceler enable A _x 0: Noise cancellation function for TIA0 _x is disabled. 1: Noise cancellation function for TIA0 _x is enabled.

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1.

Timer I/O control register 2 (TIOR2A) is an 8-bit/16-bit/32-bit readable/writable register that selects the noise canceler clock and enables and disables the noise cancelers for external input signals (TIA0_x).

TIOR2A is initialized to 0000 0000_H after reset.

Noise Canceler Clock Select G A_x (NCKGA0 to x)

These bits select the count source clock of noise canceler counters A_x (NCNTA_x).

The noise canceler count clock selected by the noise canceler clock select bits (NCKA_x) or AGCK1 can be selected as the count source clock.

Noise Canceler Clock Select A_x (NCKA_x)

These bits select the count source clock of noise canceler counter A_x (NCNTA_x).

The noise canceler count clock or the signal on clock-bus line 5 can be selected as the count source clock. Either the PCLK clock frequency divided by 128 or the PCKL clock can be selected as the noise canceler count clock by setting the NCCSEL bit of the common controller.

Noise Canceler Enable Ax (NCEAx)

These bits enable and disable the noise cancelers for external input signal TIA0x.

When a level change of the external input signal TIA0x is detected while this bit is set to “1”, it is processed in premature-transition cancellation mode, minimum time-at-level cancellation mode, or level accumulation cancellation mode depending on the setting in the noise cancellation mode register (NCMR) of the common controller, noise cancellation mode channel register 1A (NCMCR1A) of timer A, and noise cancellation mode channel register 2A (NCMCR2A) of timer A.

In premature-transition cancellation mode, when a level change of the external input signal is detected, the change is output as the signal whose noise is removed and the corresponding noise canceler counter Ax (NCNTAx) starts counting up. Subsequent level changes are masked until the value in the counter reaches the value in the noise canceler register Ax (NCRAx). This compare match occurs, the level of the external input signal at that time is output as the signal after noise cancellation.

When these bits are cleared to “0” while NCNTAx is being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes of TIA0x are also masked over this period.

In minimum time-at-level cancellation mode, when a level change of the external input signal is detected, the corresponding noise canceler counter Ax (NCNTAx) starts counting up. If subsequent level changes are not detected until the value in the counter reaches the value in the noise canceler register Ax (NCRAx), the previously accepted level change is output as the signal after noise cancellation upon compare match of the counter and noise canceler register. When the subsequent level change is detected before the values in the counter and noise canceler register match, all the changes are treated as noise. Therefore, the noise canceler assumes that the levels of external input signals have not changed and does not change the signal after cancellation.

When these bits are cleared to “0” while NCNTAx is being incremented, counting continues until the values in the counter and the noise canceler register match or a level change on the external input signal is detected.

In level accumulation cancellation mode, the corresponding noise canceler counter Ax (NCNTAx) increments or decrements according to the input signal level. NCNTAx increments (up-counting) when the input signal level is high, and decrements (down-counting) when the input signal level is low. Up-counting continues until NCNTAx reaches the value set in the noise canceler register Ax (NCRAx). Down-counting continues until NCNTAx reaches 00_H.

When a compare match with NCRAx occurs during up-counting, the noise canceler output is updated to “1”. When a compare match with 00_H occurs during down-counting, the noise canceler output is updated to “0”.

In minimum time-at-level cancellation mode and premature-transition cancellation mode, level changes are always detected by PCLK regardless of the selected noise-canceler clock. In level accumulation cancellation mode, input level sampling is performed by the clock selected by the noise canceler clock select bit Ax (NCKAx).

For details of operations for noise cancellation, see the description of the noise cancellation mode register in **Section 30.3.2, Registers Related to Common Controller**.

30.5.2.12 TIERA — Timer Interrupt Enable Register A

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OVEA	—	—	—	—	—	—	—	ICIEA[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.29 TIERA Register Contents

Bit Position	Bit Name	Function
15	OVEA	Overflow interrupt enable 0: Interrupt request due to TCNTA overflow is disabled. 1: Interrupt request due to TCNTA overflow is enabled.
14 to 8	—	These bits are not used. Fixe these bits to "0".
7 to 0	ICIEAx	Input capture interrupt enable 0: Interrupt request due to input capture x is disabled. 1: Interrupt request due to input capture x is enabled.

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1

Timer interrupt enable register A (TIERA) is 8-bit/16-bit readable/writable registers used to set interrupt requests due to comparison in overflow and input capture.

30.5.2.13 TSRA — Timer Status Register A

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OVFA	—	—	—	—	—	—	—	ICFA[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.30 TSRA Register Contents

Bit Position	Bit Name	Function
15	OVFA	Overflow flag A 0: Indicates that TCNTA has not overflowed 1: Indicates that TCNTA has overflowed
14 to 8	—	These bits are not used. Fixe these bits to “0”.
7 to 0	ICFAx	Input capture flag Ax 0: No input capture has occurred. 1: Input capture has occurred

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1

Timer status register A (TSRA) is an 8-bit/16-bit read-only register that indicates occurrence of overflow of the free-running counter A (TCNTA) and input capture of input capture registers Ax (ICRAx).

These status flags indicate occurrence of an interrupt request, and can be cleared to “0” by setting the corresponding bit in timer status clear register A (TSCRA). If an interrupt source takes effect when one of these flags is set to “1”, an interrupt request is issued again. An interrupt request is issued even if flag clearing by the corresponding timer status clear register conflicts with flag setting by an interrupt source.

TSRA is initialized to 0000_H after reset.

Overflow Flag A (OVFA)

This bit indicates whether free-running counter A (TCNTA) has overflowed. This flag indicates that an overflow occurred in TCNTA when this bit is read as “1”. This flag cannot be set to 1 or 0 by software.

- Setting (to 1) condition
When TCNTA overflows (from FFFF FFFF_H to 0000 0000_H)
- Clearing (to 0) condition
When “1” is written to OVFA in timer status clear register A (TSCRA)

Input Capture Flag Ax (ICFAx)

These bits indicate whether the value in free-running counter A (TCNTA) has been captured by input capture register Ax (ICRAx). When one of these bits is read as “1”, the value in TCNTA has been stored in the corresponding ICRAx.

These bits cannot be set to 1 or 0 by software.

- Setting (to 1) condition
When the value of TCNTA is transferred to ICRAx on assertion of the input capture signal TIA0x
- Clearing (to 0) condition
When “1” is written to ICFAx in timer status clear register A (TSCRA)

30.5.2.14 TSCRA — Timer Status Clear Register A

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OVFCA	—	—	—	—	—	—	—	ICFCA[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 30.31 TSCRA Register Contents

Bit Position	Bit Name	Function
15	OVFCA	Overflow flag clear A enable 0: Disabled (default) 1: OVFA in the timer status register A (TSRA) is cleared to 0.
14 to 8	—	These bits are not used. Fixe these bits to “0”.
7 to 0	ICFCA _x	Input capture flag clear A _x enable 0: Disabled (default) 1: ICFCA _x in the timer status register A (TSRA) is cleared to 0

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1

TSCRA is an 8-bit/16-bit readable/writable register that sets clearing of flags at occurrence of an overflow or input capture.

When TSCRA is read, “0” is always returned.

TSCRA is initialized to 0000_H after reset.

Overflow Flag Clear A Enable (OVFCA)

Writing “1” to this bit while overflow flag A (OVFA) in the timer status register A (TSRA) is set to “1” clears OVFA to “0”. When this bit is read, “0” is always returned.

Input Capture Flag Clear A_x Enable (ICFCA_x)

Writing “1” to this bit while input capture flag A_x (ICFA_x) in the timer status register A (TSRA) is set to “1” clears ICFCA_x to “0”. When this bit is read, “0” is always returned.

30.5.2.15 ICRAx — Input Capture Registers Ax

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICAx[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICAx[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.32 ICRAx Register Contents

Bit Position	Bit Name	Function
31 to 0	ICAx	Input capture Ax These bits store 32-bit input capture value.

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1

Input capture registers Ax (ICRAx) are 32-bit read-only registers dedicated to input capture. No value can be written to these registers.

These input capture-dedicated registers detect an external input capture signal (TIA0x) and store the free-running counter A (TCNTA) value. At this time, the corresponding bit in the timer status register A (TSRA) is set to “1”.

The detection edge for the input capture signal is set by the I/O control bits in the timer I/O control register 1 (TIOR1A).

ICRAx can be read only in 32-bit units. Do not read these registers in 16-bit or 8-bit units.

ICRAx is initialized to 0000 0000_H after reset.

30.5.2.16 ICRA2x — Input Capture Registers A2x

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICA2x[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICA2x[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.33 ICRA2x Register Contents

Bit Position	Bit Name	Function
31 to 0	ICA2x	Input capture A2x These bits store 32-bit input capture value.

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1

Input capture registers A2x (ICRA2x) are 32-bit read-only registers dedicated to input capture. No value can be written to these registers.

These input capture-dedicated registers detect an external input capture signal (TIA00 to TIA0SB_A-1) and store the values of timer D counters (TCNT1D0, TCNT2D0). At this time, the corresponding bit in the timer status register A (TSRA) is set to “1”. Whether to store the value of TCNT1D0 or TCNT2D0 is determined by setting the ICRA2SEL bit in the timer control register 5A (TCR5A).

The detection edge for the input capture signal is set by the I/O control bits in the timer I/O control register 1 (TIOR1A).

ICRA2x can be read only in 32-bit units. Do not read these registers in 16-bit or 8-bit units.

ICRA2x is initialized to 0000 0000_H after reset.

30.5.2.17 TCNTA — Free-Running Counter A

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.34 TCNTA Register Contents

Bit Position	Bit Name	Function
31 to 0	CNTA	Timer count A These bits store the counter value in 32-bit units.

TCNTA is a 32-bit readable/writable register that counts on the signal output by the prescaler via the clock bus, external input clock signal, or multiplied-and-corrected signal output by timer B.

TCNTA starts counting up when the TAE bit in the ATU master enable register (ATUENR) is set to “1”. The clock input to the counter is selected by setting the clock select A bit (CKSELA) in timer control register 1A (TCR1A).

When TCNTA overflows (FFFF FFFF_H to 0000 0000_H), an overflow interrupt is issued and the overflow flag (OVFA) in timer status register A (TSRA) is set to “1”.

TCNTA can be read and written only in 32-bit units. Do not read or write TCNTA in 16-bit or 8-bit units.

TCNTA is initialized to 0000 0000_H after reset.

30.5.2.18 TILRA — Timer Input Signal Level Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	TIAL07	TIAL06	TIAL05	TIAL04	TIAL03	TIAL02	TIAL01	TIAL00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 30.35 TILRA Register Contents

Bit Position	Bit Name	Function
SB _A -1 to 0	TIAL0x	Input signal level read Register only for reading the TIA0x pin level

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1

The timer input signal level register (TILRA) is an 8-bit read-only register that can read external input (TIA0x) pin levels after noise cancellation.

TILRA can be read only in 8-bit units.

TILRA is initialized to 00_H after reset.

30.5.2.19 NCNTAx — Noise Canceler Counters Ax

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NCNTAx[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.36 NCNTAx Register Contents

Bit Position	Bit Name	Function
15 to 0	NCNTAx	Noise canceler count Ax These bits store a 16-bit count value.

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1

Noise canceler counters Ax (NCNTAx) are 16-bit readable/writable registers.

In premature-transition cancellation mode or minimum time-at-level cancellation mode, NCNTAx starts counting up by an assertion of external input signals (TIA0x) as a trigger when the noise canceler function is enabled by the noise canceler enable bit (NCEAx) in timer I/O control register 2 (TIOR2A). In level accumulation cancellation mode, NCNTAx performs up-counting or down-counting according to the external input level. The count clock for the noise cancelers or the signal on clock-bus line 5 can be selected as the count source by setting the noise canceler clock select bits (NCKAx) in TIOR2A.

NCNTAx is initialized to 00_H after reset.

The noise cancellation mode can be selected from premature-transition cancellation mode, minimum time-at-level cancellation mode, and the level accumulation cancellation mode, by setting the noise cancellation mode select bit (NCMSEL) and the timer A noise cancellation mode bit (NCMA) in the noise cancellation mode register (NCMR) of the common controller, the noise cancellation mode bits (NCM1Ax) in noise cancellation mode channel register 1A (NCMCR1A) of timer A, and the noise cancellation mode bits (NCM2Ax) in noise cancellation mode channel register 2A (NCMCR2A) of timer A.

- Premature-transition cancellation mode

If a level change on pin TIA0x is detected when the NCEAx bit is “1” and NCNTAx is stopped, NCNTAx starts counting up. The counter is cleared to 0000_H and stopped on the first edge of the PCLK after the counter value matches the value in noise canceler register Ax (NCRAx).

NCNTAx is incremented regardless of the TAE bit in the ATU master enable register (ATUENR).

The level change at the start of counting is output as the signal after noise cancellation and its edge is to be detected. However, since subsequent level changes are masked until the counter value matches the value in NCRAx, the signal after noise cancellation does not change. When the values in the counter and NCRAx match, the input signal level at that time is output as the signal after noise cancellation.

When NCEAx bits are cleared to “0” while the counter is being incremented, counting continues until the values in the counter and NCRAx match. The subsequent level changes are also masked over this period.

- Minimum time-at-level cancellation mode

If a level change on pin TIA0x is detected when the NCEAx bit is “1” and NCNTAx is stopped, NCNTAx starts counting up. If subsequent level change is detected or the values in the counter and the noise canceler register Ax (NCRAx) match, the counter is cleared to 0000_H and stopped on the next PCLK cycle.

NCNTAx is incremented regardless of the TAE bit in the ATU master enable register (ATUENR).

The signal after noise cancellation changes only upon compare match between the counter and NCRAx in synchronization with the level change at the start of counting. When the counter is stopped before the compare match with NCRAx, level changes at the start and stop of counting are masked and the signal after noise cancellation does not change.

When the NCEAx bit is cleared during the counter operation, counting continues until compare match or a level change on the pin is detected.

- Level accumulation cancellation mode

When the NCEAx bit is set to “1”, NCNTAx performs up-counting or down-counting depending on the input signal level. NCNTAx performs up-counting when the input level is high, and if the counter value matches the value of the noise cancel register Ax (NCRAx), the up-counting stops at the next PCLK cycle. NCNTAx performs down-counting when the input level is low, and if the counter value reaches 0000_H, the down-counting stops at the next PCLK cycle.

Counting of NCNTAx is performed regardless of the setting of the TAE bit of the ATU master enable register (ATUENR).

If an up-count results in a compare match of NCRAx, the noise canceler output is updated to “1”. If a down-count results in a compare match with 0000_H, the noise canceler output is updated to “0”.

If the NCEAx bit is cleared during counting operation, the noise cancel counter stops counting, and the value is changed from the noise canceler output to the current input signal level. Note that clearing the NCEAx bit in level accumulation cancellation mode might cause an edge detection due to this value change.

For details of operations for noise cancellation, see the description of the noise cancellation mode register in **Section 30.3.2, Registers Related to Common Controller**.

30.5.2.20 NCRAx — Noise Cancel Registers Ax

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NCTAx[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.37 NCRAx Register Contents

Bit Position	Bit Name	Function
15 to 0	NCTAx	Noise cancellation time Ax These bits store the TIA0x noise cancellation time (16-bit compare value)

Note: x = 0 to SB_A-1: Corresponding to channels 0 to SB_A-1

Noise cancel registers Ax (NCRAx) are 16-bit readable/writable registers that set the upper limit value of the noise cancel counters Ax (NCNTAx). When the PCLK cycle divided by 128 is selected as the noise cancel clock, setting FFFF_H to this register can cancel noise for maximum of 0.21 second period (when PCLK = 40 MHz).

The noise cancellation mode can be selected from premature-transition cancellation mode, minimum time-at-level cancellation mode, and the level accumulation cancellation mode, by setting the noise cancellation mode select bit (NCMSEL) and the timer A noise cancellation mode bit (NCMA) in the noise cancellation mode register (NCMR) of the common controller, the noise cancellation mode bits (NCMI1Ax) in noise cancellation mode channel register 1A (NCMCR1A) of timer A, and the noise cancellation mode bits (NCM2Ax) in noise cancellation mode channel register 2A (NCMCR2A) of timer A.

- Premature-transition cancellation mode

While NCNTAx counting is in progress, level changes of succeeding input signals are masked. Values of NCNTAx and NCRAx are always compared. If a compare match occurs, the NCNTAx counter value is cleared, the counting is stopped, and the input signal mask is released at the next PCLK cycle.

- Minimum time-at-level cancellation mode

While NCNTAx counting is in progress, the noise canceler processing is waited for. Values of NCNTAx and NCRAx are always compared. If a compare match occurs, the NCNTAx counter value is cleared, the counting is stopped, and the noise canceler starts to output the input signal after noise cancellation at the next PCLK cycle.

- Level accumulation cancellation mode

When NCNTAx is performing up-counting, values of NCNTAx and NCRAx are compared. When a compare match occurs, the up-counting of NCNTAx stops at the next PCLK cycle. When NCNTAx is performing down-counting, NCNTAx is compared with 0000_H.

NCRAx is initialized to 0000_H after reset.

For details of operations for noise cancellation, see the description of the noise cancellation mode register in **Section 30.3.2, Registers Related to Common Controller**.

30.5.3 Details of Operation

30.5.3.1 Operation of Noise Canceler

For details of operations for noise cancellation, see the description of the noise cancellation mode register in **Section 30.3.2, Registers Related to Common Controller**.

30.5.3.2 Operation of Free-Running Counter

Free-running counter A (TCNTA) starts counting up by setting the TAE bit in ATU master enable register (ATUENR) to “1”. When TCNTA overflows (FFFF FFFF_H to 0000 0000_H), the OVFA bit in timer status register A (TSRA) is set to “1”. An interrupt request is issued to the CPU at this time. After overflow, TCNTA continues counting up from 0000 0000_H.

If the TAE bit in ATU master enable register (AUTENR) is cleared to “0” when TCNTA is operating, TCNTA stops counting while retaining the value at that time. By setting the TAE bit to “1” again, TCNTA resumes counting from the value when it is stopped.

TCNTA can be written during operation and writing takes priority over counting. After that, TCNTA resumes counting from the written value. The write access is completed in two cycles of the PCLK clock, regardless of the count source.

CAUTION

The prescalers run independently of the setting of the TAE bit and are not initialized when the TCNTA starts counting. Therefore, the time from when the TAE bit is set to when TCNTA starts counting may be less than the cycle of the selected count source clock (resolution), being undetermined due to hardware factors.

Figure 30.11 shows an example of free-running counter A (TCNTA) operation.

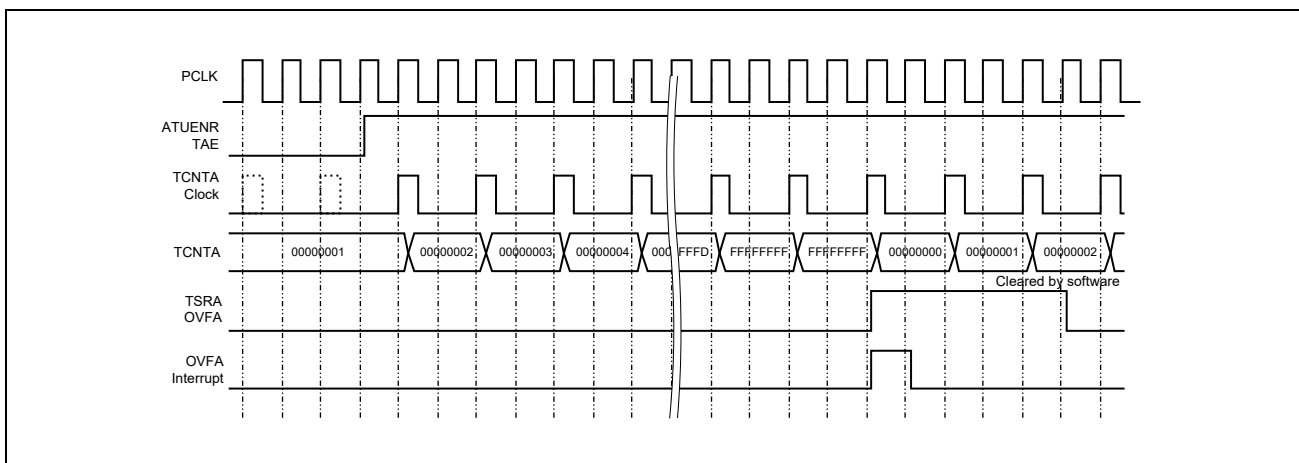


Figure 30.11 Example of Free-Running Counter A (TCNTA) Operation: Overflow Timing

30.5.3.3 Input Capture

Input capture is performed by input capture register Ax (ICRAx) when input capture is enabled in bits IOAx in timer I/O control register 1 (TIOR1A). ICRAx capture the value in free-running counter A (TCNTA) by detecting the edge of external input signals (TIA0x).

Noise on the external input signals can be removed by the noise cancelers. TCNTA starts counting up by setting the TAE bit in ATU master enable register (ATUENR). When the valid edge of the signal corresponding to ICRA is detected, the corresponding bit in timer status register A (TSRA) is set to “1” and the value in TCNTA is transferred to ICRA. The detection edge can be selected from rising edge, falling edge, or both rising and falling edges. It is also possible to output an interrupt request to the CPU upon input capture.

When input capture and writing to free-running counter A (TCNTA) occur simultaneously, the counter value before the write is captured.

Figure 30.12 shows an example of input capture when the edges to be sensed are rising edges for TIA00, falling edges for TIA01, and both edges for TIA02

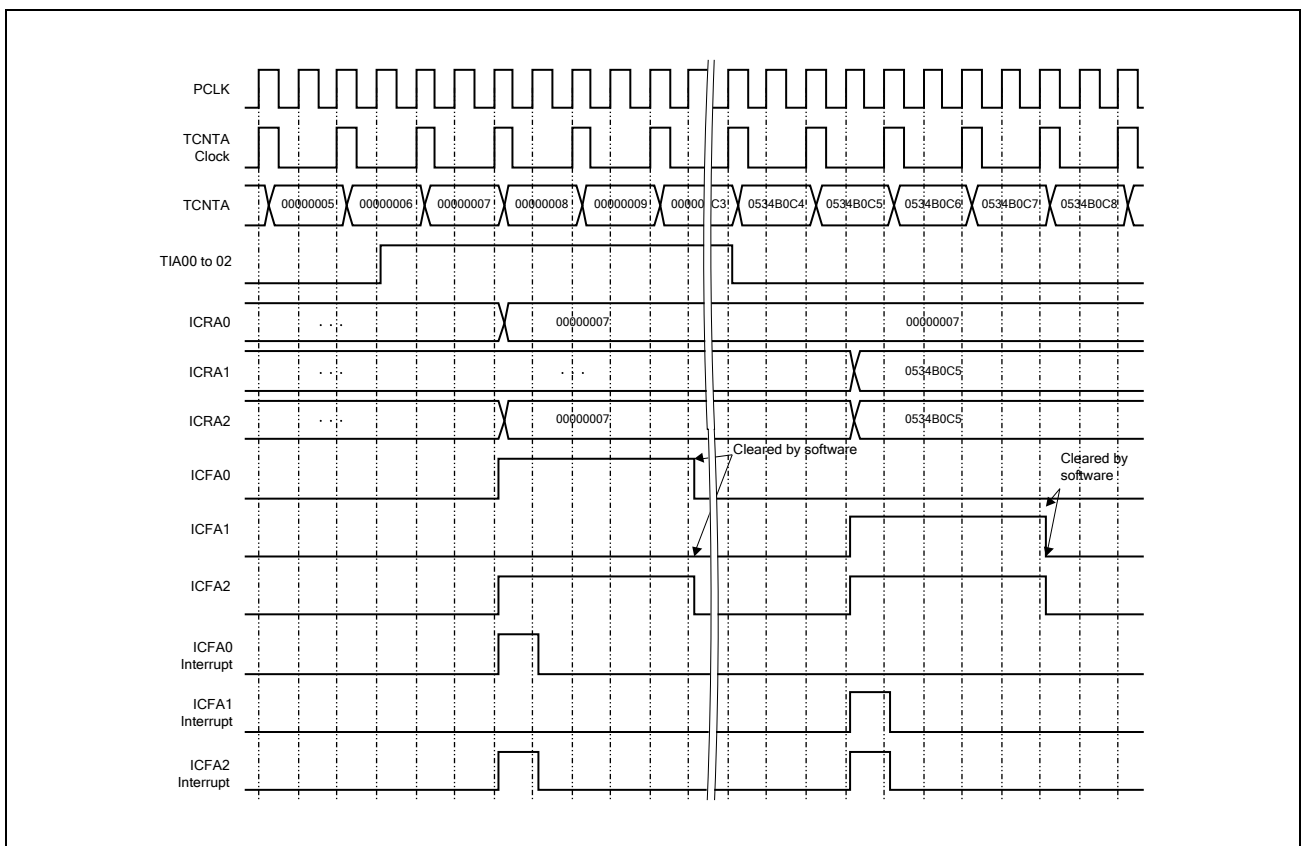


Figure 30.12 Example of Input Capture of Timer A

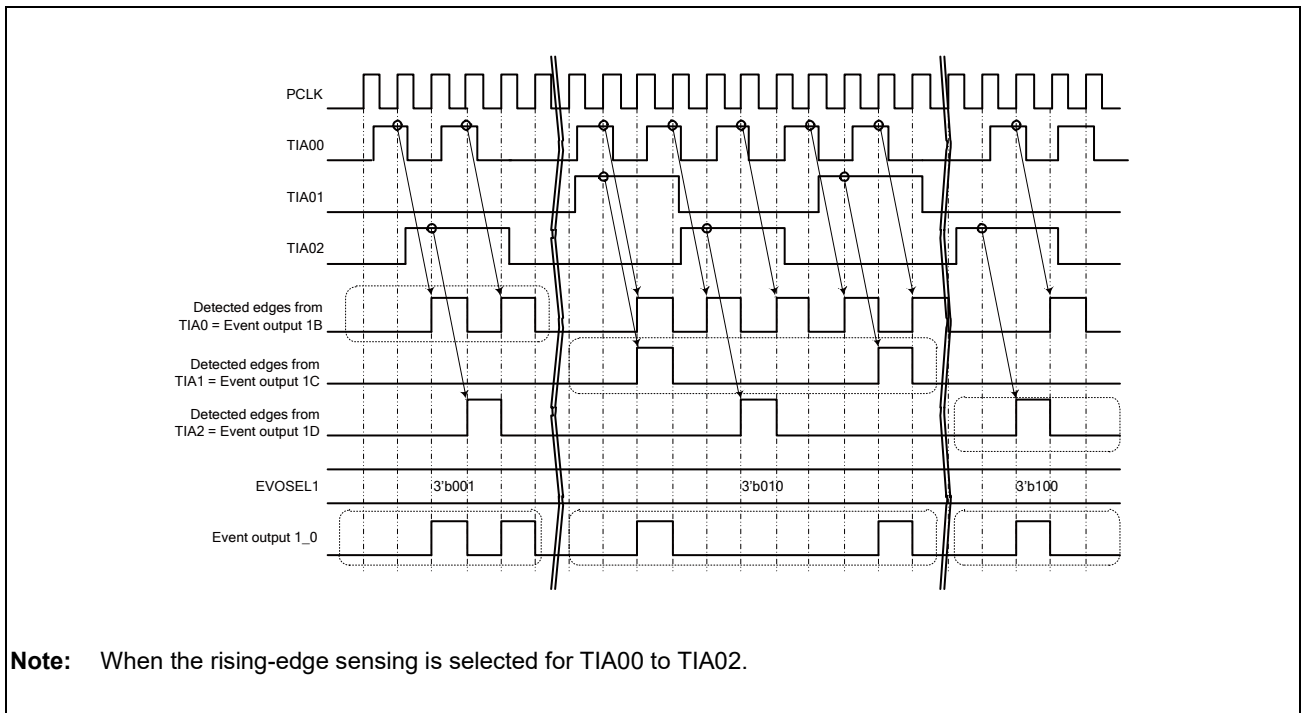


Figure 30.13 Example of TIA00 to TIA02 Event Output

By setting the EVOSEL1 bit in timer control register 1A (TCR1A), any of external input signals TIA02 to TIA00 after edge detection and noise cancellation can be output to timers B and C as an event (event output 1_0).

Furthermore, either TIA01 or TIA02 or a combination of them (after edge detection and noise cancellation) can be output to timers C and D as an event (event outputs 2A and 2B) by setting the EVOSEL2A and EVOSEL2B bits. In addition, any one of external input pins TIA0x or a combination of them (after edge detection and noise cancellation) can be output to timers B and C as an event (event output 1_0) by setting the EVOSELE1E and EVOSELE1 bits in timer control register 2A (TCR2A). Any one of external input pins TIAs or a combination of them (after edge detection and noise cancellation) can be output to timers C and D as an event (event outputs 2A and 2B) by setting the EVOSELE2AE and EVOSELE2A bits in timer control register 3A (TCR3A) and the EVOSELE2BE and EVOSELE2B bits in timer control register 4A (TCR4A).

Figure 30.13 shows an example of event output operation in case EVOSEL1 is set to 001_B (TIA00 input edge output), 010_B (TIA01 input edge output), and 100_B (TIA02 input edge output).

30.5.3.4 Pin Level Capture Operation

Levels of the external input pins (TIA0x) after noise cancellation can be read from the timer input signal level register (TILRA). **Figure 30.14** shows an example of operation of the TILRA register when signals are input to TIA0x.

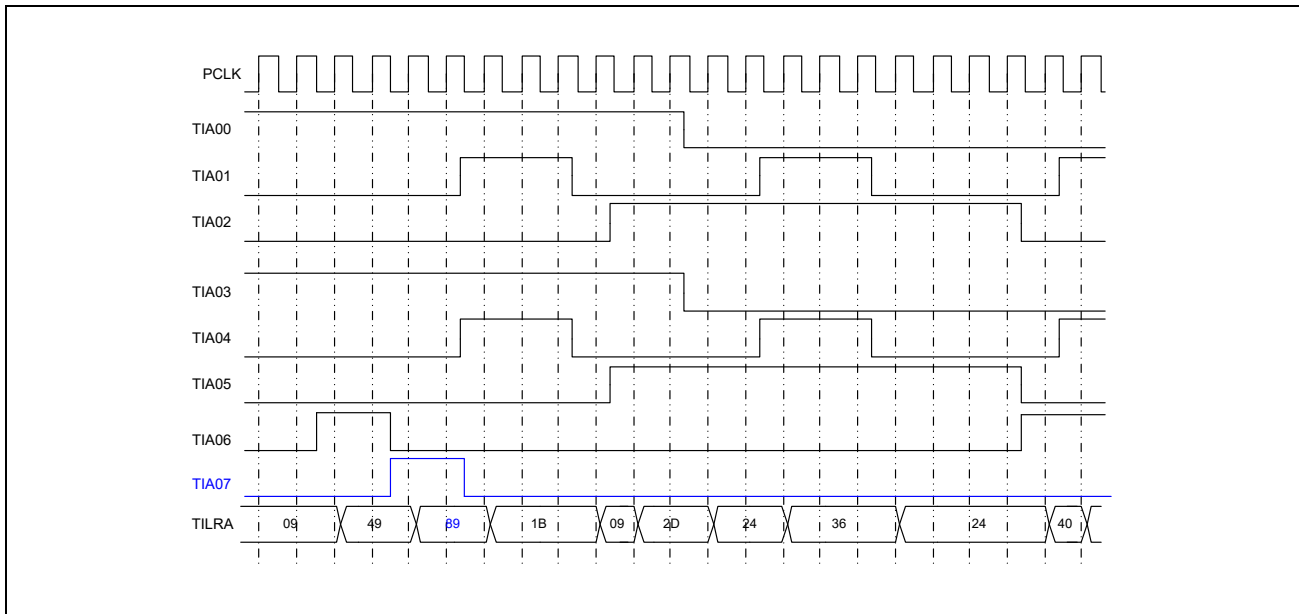


Figure 30.14 Example of TILRA Capture Operation

30.6 Timer B

30.6.1 Overview of Operation

Timer B generates a multiplied-and-corrected clock signal based on an external-event input and supplies the generated signal to other timers. Timer B consists of an edge-interval measuring block, frequency-multiplying clock generator, and corrector for the frequency-multiplied clock signal.

Edge-Interval Measuring Block

The edge-interval measuring block measures the intervals between edges of external-event signals 1_0 input via timer A (TCNTB0, ICRB0). Interrupt requests can be issued for the CPU in response to matches between edge interval measuring counter (TCNTB0) and output compare register B0 (OCRB0). TCNTB0 can also be captured in ICRB2 via ICRB1 on matches between event counter B1 (TCNTB1) that uses an external event input 1_0 as the count source and output compare register B1 (OCRB1). This provides a way to measure the intervals between multiple event inputs. Although TCNTB0 is cleared every event input 1_0, ICRB1 keeps a running total of the TCNTB0 value. ICRB2 latches the running totals on compare matches of the event counter. Furthermore, it is also possible to set the value of the event counter B1 (TCNTB1) that uses an external event input 1_0 as a count source to 01_H by the CLRBI bit (TCNTB1 clear bit) in the timer control register B (TCRB) at the next external event input timing after an event compare match with OCRB10. Furthermore, up to seven capture values (ICRB0) can be retained in record registers B1 to B6 (RECRB1 to RECRB6) at the external input event timing. Values retained in ICRB0 and RECRB1 to RECRB6 can be backed up in record backup registers B0 to B6 (RBURB0 to RBURB6) on event compare match between the event counter B1 (TCNTB1) and OCRB12.

For event counter (TCNB1), 24 compare registers (OCRB20 to OCRB 43) are provided for sequencer control. The sequencer transits to 24 states upon compare match between event counter (TCNTB1 and compare register (OCRB20 to OCRB 43). When the sequencer has transited, it generates a correction value for generating a frequency-multiplied clock by calculating the correction formula that uses the captured value (ICRB0) and Record Registers B1 to 6 (RECRB1 to RECRB 6).

In the edge-interval measuring block, the event counter B1 (TCNTB1) value is captured for the event counter B1 (TCNTB1) at occurrence of any of seven external event input signals 1B to 1I that are input via timer A (ICRB30 to ICRB37). The event counter B1 (TCNTB1) is not cleared at occurrence of any of external event input signals 1B to 1I.

The edge-interval measuring block consists of three channels. Edge-interval blocks 0 to 3 operate based on the external event signals 1_0, 1_1, and 1_2 input via timer A, respectively.

Table 30.38 below shows the events that correspond to each channel of the edge-interval measuring block and the registers provided for each channel.

What external-event and register value of the channel is used for other blocks can be selected by channel select register B (CHSELBR).

Table 30.38 Edge-Interval Measuring Blocks

	Edge-interval block 0	Edge-interval block 1	Edge-interval block 2
Corresponding event	External event 1_0	External event 1_1	External event 1_2
Register	TCNTB0	TCNTB0S1	TCNTB0S2
	ICRB0	ICRB0S1	ICRB0S2
	OCRB0	OCRB0S1	OCRB0S2
	TCNTB1	TCNTB1S1	TCNTB1S2
	OCRB1	OCRB1S1	OCRB1S2
	OCRB10	OCRB10S1	OCRB10S2
	OCRB11	OCRB11S1	OCRB11S2
	OCRB12	OCRB12S1	OCRB12S2
	ICRB1	ICRB1S1	ICRB1S2
	ICRB2	ICRB2S1	ICRB2S2

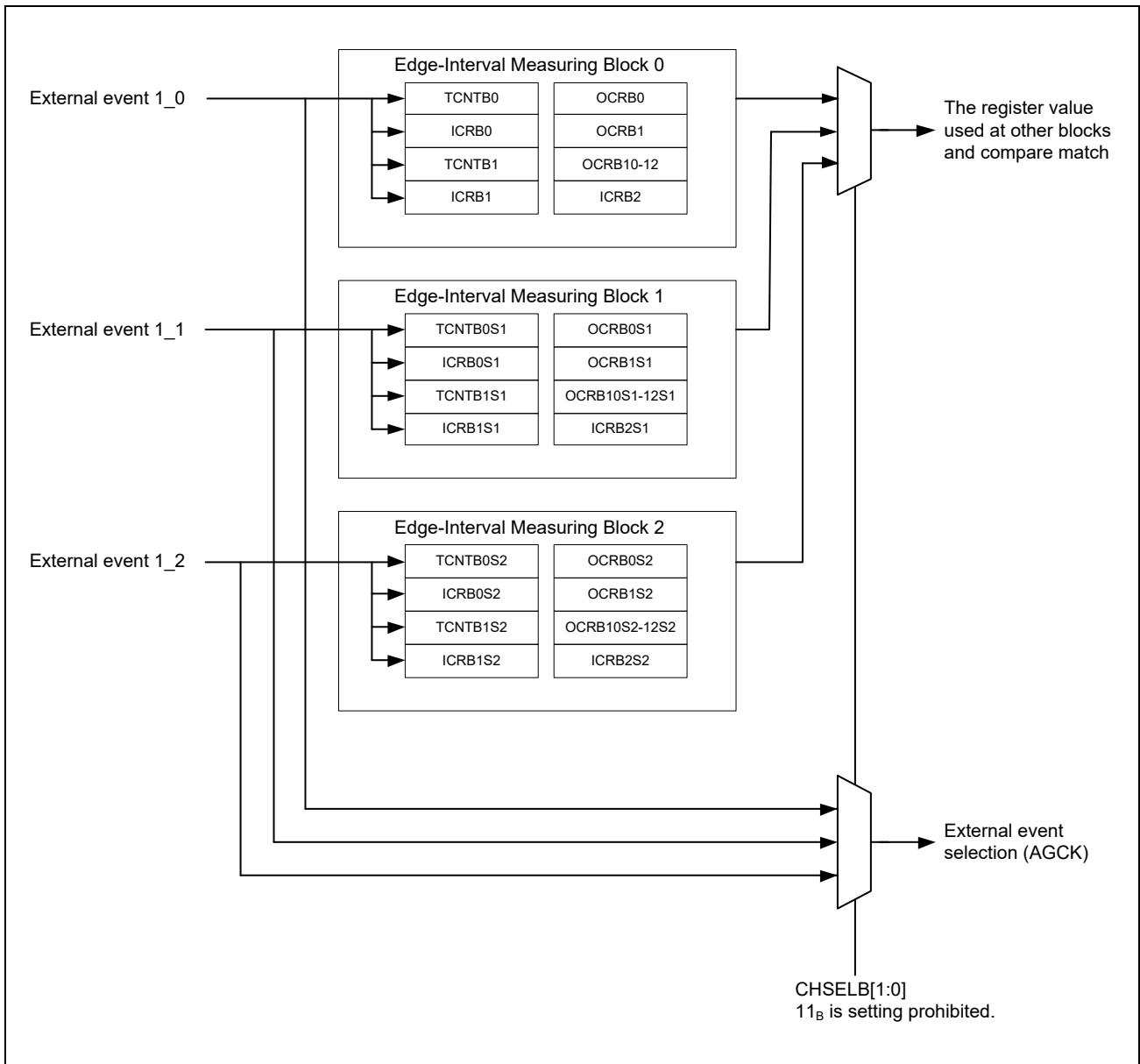


Figure 30.15 Timer B Edge-Interval Measuring Blocks 0 to 2

NOTE

The output event selection (AGCK) is selected from output event 1_0, 1_1 and 1_2 in CHSELB[1:0]. When CHSELB [1:0] is 00B, output event 1_0 is selected. When CHSELB [1:0] is 01B, output event 1_1 is selected. When CHSELB [1:0] is 10B, output event 1_2 is selected.

Frequency-Multiplied Clock Generator

The frequency-multiplied clock generator generates a clock signal obtained by frequency-multiplying an external-event selection (AGCK) input signal by a value from 1 to 4095.

In the edge-interval measuring block, down-counting (TCNTB2) with reload (RLDB) is performed for the capture value (ICRB0) in the input edge-interval measuring counter B0 (TCNTB0) or the corrected value calculated through sequencer control. Down-counting is performed by the set division ratio value specified in the PIMR select register. In addition to PIMR1, a division ratio value of PIMR2 can be selected by an event compare match between the event counter B1 (TCNTB1) in the edge-interval measuring block and OCRB11. When the down counter underflows, the multiplied clock (AGCK1) signal is asserted. (The multiplied clock is supplied as the count clock source for the noise canceler of timer A.)

TCNTB6 is a measuring counter using the multiplied clock as a source clock, and an interrupt request to the CPU can be output due to the compare match with OCRB6. ICRB6 is a capture register, and this TCNTB6 value is captured when an external event selection (AGCK) input occurs.

TCNTB6M is a measuring counter using the multiplied clock as a source clock. The up-count value can be set within a range from 0 to less than 4 in units of 1/64. When an external event selection (AGCK) input occurs, the TCNTB6M value is compared with the ICRB6 value. When the TCNTB6M value is smaller than the ICRB6 value, an interrupt request to the CPU can be output.

Frequency-Multiplied Clock Signal Corrector

The multiplied clock signal (AGCK1) needs to be corrected when two consecutive edge intervals differ significantly, since the earlier interval is referred to in calculating the multiplier for the current interval. The frequency-multiplied clock signal corrector generates a multiplied-and-corrected clock signal (AGCKM) by using three correcting counters (TCNTB3 to TCNTB5) and correcting counter clearing register B (TCCLR B). Output of the clock signal thus produced on clock-bus line 5 can be selected by using the clock-bus control register (CBCNT). Other timers can then use this clock as a source for counting. For the automatic up-counting function that continues counting until the multiplied-and-corrected clock generating counter B5 (TCNTB5) value matches the correcting counter clear register B (TCCLR B) value when the multiplied-and-corrected clock (AGCKM) is generated, whether to enable or disable this function is selectable. Furthermore, the correcting counter (TCNTB3) can be automatically cleared by the CLR B3 bit (TCNTB3 clear bit) in the timer control register B (TCRB) at the timing of an event compare match between the event counter B1 (TCNTB1) in the edge-interval measuring block and OCRB10 or between the correcting event counter B3 (TCNTB3) and OCRB8.

Figure 30.16 is a block diagram of timer B.

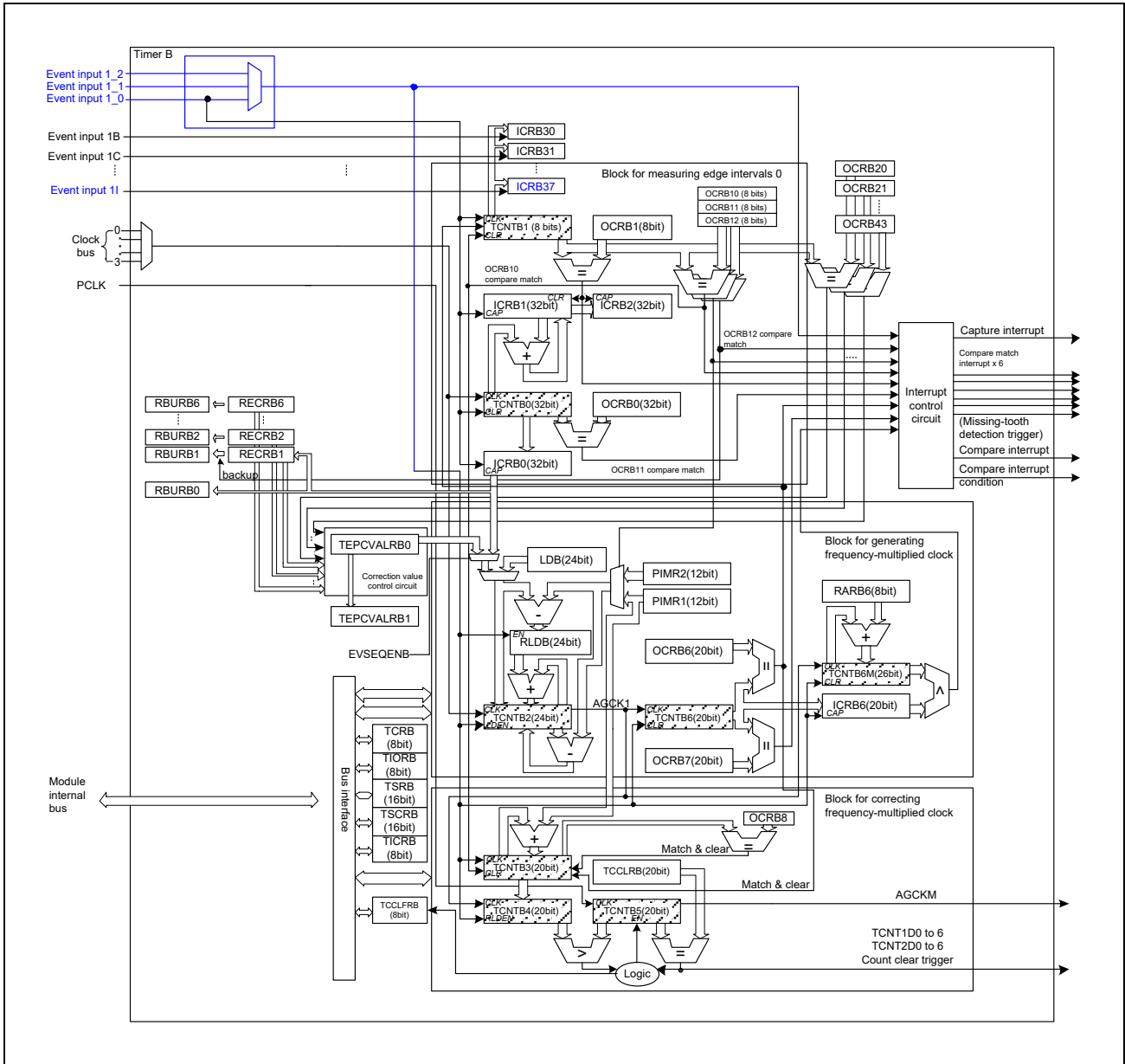


Figure 30.16 Block Diagram of Timer B

CAUTION

The edge-interval measuring block consists of three channels. The channel to be used is selected by channel select register B (CHSELBR). See **Table 30.38** for the registers provided for each channel.

30.6.2 Timer B Control Registers

30.6.2.1 TCRB — Timer Control Register B

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	PIMRSEL	CLRB3	CLRB1	CLRB3SEL[1:0]		CLRB1SEL	CKSELB[1:0]	
Value after reset	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.39 TCRB Register Contents

Bit Position	Bit Name	Function
7	PIMRSEL	PIMR select 0: PIMR1 is always selected. 1: PIMR2 is selected instead of PIMR1 while TCNTB1 and OCRB11 match.
6	CLRB3	TCNTB3 clear 0: TCNTB3 is not cleared. 1: TCNTB3 is cleared.
5	CLRB1	TCNTB1 clear select 0: TCNTB1 is not cleared. 1: TCNTB1 is cleared.
4, 3	CLRB3SEL	TCNTB3 clear select 00: Compare match between TCNTB1 and OCRB10 is used to clear TCNTB3 01: Compare match between TCNTB6 and OCRB6 is used to clear TCNTB3 10: Compare match between TCNTB3 and OCRB8 is used to clear TCNTB3 11: Setting prohibited
2	CLRB1SEL	TCNTB1 clear select 0: Compare match between TCNTB1 and OCRB10 is used to clear TCNTB1 1: Compare match between TCNTB6 and OCRB6 is used to clear TCNTB1
1, 0	CKSELB	Clock select B 00: Clock-bus line 0 is selected for count source 01: Clock-bus line 1 is selected for count source 10: Clock-bus line 2 is selected for count source 11: Clock-bus line 3 is selected for count source

TCRB is an 8-bit readable/writable register that allows you to select a count source for the input edge interval measuring counter B0 (TCNTB0) and the reload counter B2 (TCNTB2), select whether to clear the TCNTB1 and TCNTB3 values at a compare match between TCNTB1 and OCRB10 or a compare match between TCNTB6 and OCRB6, and to select a pulse interval multiplier.

TCRB can be read and written in 8-bit units.

TCRB is initialized to 00_H after reset.

Clock Select B (CKSELB)

These bits select the clock driving TCNTB0 and TCNTB2 from clock-bus lines 0 to 3, which are clock signals divided by prescalers 0 to 3. The counters are incremented on the rising edge of the selected clock.

To select the clock, stop timer B operation.

Pulse Interval Multiplier Select (PIMRSEL)

By setting the PIMRSEL bit to 1 while TCNTB1 matches OCRB11, PIMR2 is selected for down-count values to be loaded to reload register B (RLDB) or for down-count values of TCNT2B. In other cases, PIMR1 is selected. However, PIMR1 is always used for adding the TCNTB3 value.

While the PIMRSEL bit is 0, PIMR1 is always used.

TCNTB1 Clear (CLRB1)

Setting the TCNTB1 clear bit (CLRB1) to 1 clears the TCNTB1 value at the timing of the first event input after a compare match between TCNTB1 and OCRB10 or between TCNTB6 and OCRB6 for the CLRB1SEL setting condition.

TCNTB3 Clear (CLRB3)

Setting the TCNTB3 clear bit (CLRB3) to 1 clears the TCNTB3 value in synchronization with the first PCLK after a compare match between TCNTB1 and OCRB10, between TCNTB6 and OCRB6, or between TCNTB3 and OCRB8 for the CLRB3SEL setting condition.

TCNTB1 Clear Select (CLRB1SEL)

Setting the TCNTB1 clear select bit (CLRB1SEL) sets the TCNTB1 clearing condition. When CLRB1SEL is 0, TCNTB1 can be cleared on compare match between TCNTB1 and OCRB10. When CLRB1SEL is 1, TCNTB1 can be cleared on compare match between TCNTB6 and OCRB6.

TCNTB3 Clear Select (CLRB3SEL)

Setting the TCNTB1 clear select bit (CLRB3SEL) sets TCNTB3 clearing condition. When CLRB3SEL is 00, TCNTB3 can be cleared on compare match between TCNTB1 and OCRB10. When CLRB3SEL is 01, TCNTB3 can be cleared on compare match between TCNTB6 and OCRB6. When CLRB3SEL is 10, TCNTB3 can be cleared on compare match between TCNTB3 and OCRB8.

30.6.2.2 TCRBS1 — Timer Control Register BS1

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	CLRB1S1	—	—	CLRB1SELS1	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R	R/W	R	R

Table 30.40 TCRBS1 Register Contents

Bit Position	Bit Name	Function
7, 6	—	These bits are not used. Fix these bits to “0”.
5	CLRB1S1	TCNTB1S1 clear 0: TCNTB1S1 is not cleared. 1: TCNTB1S1 is cleared.
4, 3	—	These bits are not used. Fix these bits to “0”.
2	CLRB1SELS1	TCNTB1S1 clear select 0: Compare match between TCNTB1S1 and OCB10S1 is used to clear TCNTB1S1. 1: Compare match between TCNTB6 and OCB6 is used to clear TCNTB1S1.
1, 0	—	These bits are not used. Fix these bits to “0”.

TCRBS1 is an 8-bit readable/writable register that allows you to select whether or not to clear TCNTB1S1 upon compare match between TCNTB1S1 and OCB10S1 or compare match between TCNTB6 and OCB6.

TCRBS1 can be read and written in 8-bit units.

TCRBS1 is initialized to 00_H after reset.

TCNTB1S1 Clear (CLRB1S1)

Setting the TCNTB1S1 clear bit (CLRB1S1) to 1 clears the TCNTB1S1 value at the timing of the first event input after a compare match between TCNTB1S1 and OCB10S1 or between TCNTB6 and OCB6 for the CLRB1SELS1 setting condition.

TCNTB1S1 Clear Select (CLRB1SELS1)

Setting the TCNTB1S1 clear select bit (CLRB1SELS1) sets the TCNTB1S1 clearing condition. When CLRB1SELS1 is 0, TCNTB1S1 can be cleared on compare match between TCNTB1S1 and OCB10S1. When CLRB1SELS1 is 1, TCNTB1S1 can be cleared on compare match between TCNTB6 and OCB6.

30.6.2.3 TCRBS2 — Timer Control Register BS2

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	CLRB1S2	—	—	CLRB1SELS2	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R	R/W	R	R

Table 30.41 TCRBS2 Register Contents

Bit Position	Bit Name	Function
7, 6	—	These bits are not used. Fix these bits to “0”.
5	CLRB1S2	TCNTB1S2 clear 0: TCNTB1S2 is not cleared. 1: TCNTB1S2 is cleared.
4, 3	—	These bits are not used. Fix these bits to “0”.
2	CLRB1SELS2	TCNTB1S2 clear select 0: Compare match between TCNTB1S2 and OCRB10S2 is used to clear TCNTB1S2. 1: Compare match between TCNTB6 and OCRB6 is used to clear TCNTB1S2
1, 0	—	These bits are not used. Fix these bits to “0”.

TCRBS2 is an 8-bit readable/writable register that allows you to select whether or not to clear TCNTB1S2 upon compare match between TCNTB1S2 and OCRB10S2 or compare match between TCNTB6 and OCRB6.

TCRBS2 can be read and written in 8-bit units.

TCRBS2 is initialized to 00_H after reset.

TCNTB1S2 Clear (CLRB1S2)

Setting the TCNTB1S2 clear bit (CLRB1S2) to 1 clears the TCNTB1S2 value at the timing of the first event input after a compare match between TCNTB1S2 and OCRB10S2 or between TCNTB6 and OCRB6 for the CLRB1SELS2 setting condition.

TCNTB1S2 Clear Select (CLRB1SELS2)

Setting the TCNTB1S2 clear select bit (CLRB1SELS2) sets the TCNTB1S2 clearing condition. When CLRB1SELS2 is 0, TCNTB1S2 can be cleared on compare match between TCNTB1S2 and OCRB10S2. When CLRB1SELS2 is 1, TCNTB1S2 can be cleared on compare match between TCNTB6 and OCRB6.

30.6.2.4 TIORB — Timer I/O Control Register B

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LDSEL	CTCNTB5	EVCNTB	LDEN	CCS	—	—	IOB6
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Table 30.42 TIORB Register Contents

Bit Position	Bit Name	Function
7	LDSEL	Loading Data Select 0: ICRB0 value is used to calculate the data to be loaded to TCNTB2 and RLDB. 1: LDB value is used to calculate the data to be loaded to TCNTB2 and RLDB
6	CTCNTB5	Count Control B5 0: TCNTB5 count operation is enabled. 1: TCNTB5 count operation is disabled.
5	EVCNTB	Event Control B 0: External event input 1_0, 1_1, 1_2 is disabled. 1: External event input 1_0, 1_1, 1_2 is enabled.
4	LDEN	Load Enable 0: TCNTB2 and RLDB are updated on ICEB0 input capture. 1: TCNTB2 and RLDB are not updated on ICEB0 input capture.
3	CCS	Counter Correction Select 0: TCNTB4 is in operation when TCNTB3 = TCNTB4 1: [When CU4SEL bit is 0] TCNTB4 is stopped when TCNTB3 = TCNTB4 [When CU4SEL bit is 1] TCNTB4 is stopped when TCNTB3 = TCNTB4 - 1
2, 1	—	These bits are not used. Fix these bits to "0".
0	IOB6	I/O Control B6 0: Compare match between TCNTB6 and OCRB6 is disabled. 1: Compare match between TCNTB6 and OCRB6 is enabled.

TIORB is an 8-bit readable/writable register that selects the source of the frequency-multiplied clock and enables and disables the externally input signals, loading data, and correcting frequency multiplied clock. TIORB also controls multiplied-and-corrected clock generating counter B5 (TCNTB5) and output compare register B6 (OCRB6).

TIORB can be read and written in 8-bit units.

TIORB is initialized to 00_H after reset.

Loading Data Select (LDSEL)

Selects the register to be loaded to reloadable counter B2 (TCNTB2) and to be used for calculating data to be loaded to reload register B (RLDB) from ICRB0 or LDB.

Count Control B5 (CTCNTB5)

Selects whether the multiplied-and-corrected clock generating counter B5 (TCNTB5) is enabled or stopped.

Setting this bit to 1 stops TCNTB5 and the multiplied-and-corrected clock to be output to other timers. Even if the counter is stopped, it is not cleared. Clearing this bit to 0 resumes TCNTB5 and multiplied- and-corrected clock.

Event Control B (EVCNTB)

Disables and enables the externally input 1_0, 1_1, 1_2 events. Clearing this bit to 0 disables the input. Setting this bit to 1 enables the input with which input capture or generating the multiplied-and-corrected clock signal.

Load Enable (LDEN)

Selects whether to update the input capture register B0 (ICRB0) value, the reload counter B2 (TCNTB2) value, and the reload register B (RLDB) value at the input capture of the input edge interval measuring counter B0 (TCNTB0).

Counter Correction Select (CCS)

Selects whether or not correcting multiplied clock counter B4 (TCNTB4) is stopped when $TCNTB3 = TCNTB4$ (CU4SEL = 0) or $TCNTB3 = TCNTB4 - 1$ (CU4SEL = 1).

I/O Control B6 (IOB6)

Enables and disables compare match between TCNTB6 and OCRB6. When this bit is cleared to 0, compare match between TCNTB6 and OCRB6 is disabled. When it is set to 1, compare match between TCNTB6 and OCRB6 is enabled. At this time, an interrupt request is output to the CPU at a compare match.

CAUTION

When setting "1" to the CU4SEL bit of TCNT4CRB, be sure to set "1" to the CCS bit.

30.6.2.5 TSRB — Timer Status Register B

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CMFB6 E	CMFB6 M	—	CMFB1 2	CMFB1 1	CMFB1 0	CMFB6	CMFB1	ICFB0	CMFB0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.43 TSRB Register Contents

Bit Position	Bit Name	Function
15-10	—	These bits are not used. Fix these bits to “0”.
9	CMFB6E	Compare Match Flag B6E 0: Function is not selected or no compare match has occurred. 1: A compare match has occurred under the condition set for IREGB6E in TICRB.
8	CMFB6M	Compare Match Flag B6M 0: The TCNTB6M value is the ICRB6 value or larger. 1: When the TCNTB6M value is smaller than ICRB6, the next event input occurs
7	—	This bit is not used. Fixe this bit to “0”.
6	CMFB12	Compare Match Flag B12 0: No compare match has occurred. 1: Compare match has occurred.
5	CMFB11	Compare Match Flag B11 0: No compare match has occurred. 1: Compare match has occurred.
4	CMFB10	Compare Match Flag B10 0: No compare match has occurred. 1: Compare match has occurred.
3	CMFB6	Compare Match Flag B6 0: No compare match has occurred. 1: Compare match has occurred.
2	CMFB1	Compare Match Flag B1 0: No compare match has occurred. 1: Compare match has occurred.
1	ICFB0	Input Capture Flag B0 0: No input capture has occurred 1: Input capture has occurred
0	CMFB0	Compare Match Flag B0 0: No compare match has occurred. 1: Compare match has occurred.

The timer status register B (TSRB) is an 8-bit or 16-bit read-only register that indicates occurrence of input capture and compare match.

These status flags indicate occurrence of an interrupt request, and can be cleared to 0 by setting the corresponding bit in timer status clear register B (TSCRB). When an interrupt source occurs while one of these flags is set to 1, an interrupt request is generated again.

Even if flag clear by the corresponding timer status clear register conflicts with flag setting due to occurrence of an interrupt source, an interrupt request is generated.

NOTE

If an interrupt source is generated while the CMFB6 flag is set, the interrupt request is not generated.

TSRB can be read in 8-bit or 16-bit units only.

TSRB is initialized to 0000_H after reset.

Compare Match Flag B6E (CMFB6E)

This status flag indicates that both (or one of) CMFB6 and CMFB6M flags have been set by the setting for IREGB6E in the timer interrupt control register B (TICRB). When 1 is read from this flag, it shows that both (or one of) CMFB6 and CMFB6M flags have been set. This flag cannot be set to 1 or 0 by the software.

- Setting (to 1) condition

When a CMFB6E interrupt request has occurred by the setting of IREGB6E in the timer interrupt control register B (TICRB). For details, see the description of the setting for IREGB6E in TICRB.

- Clearing (to 0) condition

Writing 1 to CMFCB6E in timer status clear register B (TSCRB)

Compare Match Flag B6M (CMFB6M)

This status flag indicates that the multiplied clock counter B6M (TCNTB6M) value was smaller than the input capture register B6 (ICRB6) value at the external input event timing. When 1 is read from this flag, the TCNTB6M value is not larger than the ICRB6 value and the next event input has occurred. This flag cannot be set to 1 or 0 by the software.

- Setting (to 1) condition

When the TCNTB6M count value does not exceed the ICRB6 value and the next event input has occurred

- Clearing (to 0) condition

Writing 1 to CMFCB6M in timer status clear register B (TSCRB)

Compare Match Flag B12 (CMFB12)

This status flag indicates occurrence of a compare match of output compare register B12 (OCRB12) or output compare register B12S1 (OCRB12S1) or output compare register B12S2 (OCRB12S2). When 1 is read from this flag, it shows that a compare match has occurred in CMFB12. This flag cannot be set to 1 or 0 by the software.

It's established by channel select bit (CHSELB) whether it's set when which comparing match occurred.

- Setting (to 1) condition

CHSELB	Setting (to 1) conditions
00	When the TCNTB1 count value matches the OCRB12 value and the next PCLK has occurred
01	When the TCNTB1S1 count value matches the OCRB12S1 value and the next PCLK has occurred
10	When the TCNTB1S2 count value matches the OCRB12S2 value and the next PCLK has occurred
11 (Prohibited)	—

- Clearing (to 0) condition

Writing 1 to CMFCB12 in timer status clear register B (TSCRB)

Compare Match Flag B11 (CMFB11)

This status flag indicates occurrence of a compare match of output compare register B11 (OCRB11) or output compare register B11S1 (OCRB11S1) or output compare register B11S2 (OCRB11S2). When 1 is read from this flag, it shows that a compare match has occurred in CMFB11. This flag cannot be set to 1 or 0 by the software.

It's established by channel select bit (CHSELB) whether it's set when which comparing match occurred.

- Setting (to 1) condition

CHSELB	Setting (to 1) conditions
00	When the TCNTB1 count value matches the OCRB11 value and the next PCLK has occurred
01	When the TCNTB1S1 count value matches the OCRB11S1 value and the next PCLK has occurred
10	When the TCNTB1S2 count value matches the OCRB11S2 value and the next PCLK has occurred
11 (Prohibited)	—

- Clearing (to 0) condition

Writing 1 to CMFCB11 in timer status clear register B (TSCRb)

Compare Match Flag B10 (CMFB10)

This status flag indicates occurrence of a compare match of output compare register B10 (OCRB10) or output compare register B10S1 (OCRB10S1) or output compare register B10S2 (OCRB10S2). When 1 is read from this flag, it shows that a compare match has occurred in CMFB10. This flag cannot be set to 1 or 0 by the software.

It's established by channel select bit (CHSELB) whether it's set when which comparing match occurred.

- Setting (to 1) condition

CHSELB	Setting (to 1) conditions
00	When the TCNTB1 count value matches the OCRB10 value and the next PCLK has occurred
01	When the TCNTB1S1 count value matches the OCRB10S1 value and the next PCLK has occurred
10	When the TCNTB1S2 count value matches the OCRB10S2 value and the next PCLK has occurred
11 (Prohibited)	—

- Clearing (to 0) condition

Writing 1 to CMFCB10 in timer status clear register B (TSCRb)

Compare Match Flag B6 (CMFB6)

This status flag indicates occurrence of a compare match of output compare register B6 (OCRB6). When 1 is read from this flag, it shows that a compare match has occurred in CMFB6. This flag cannot be set to 1 or 0 by the software.

This flag is automatically cleared to 0 at an OCRB7 compare match only when IREGB6 in TICRB is set to 10.

- Setting (to 1) condition

When the TCNTB6 count value matches the OCRB6 value and the next AGCK1 has occurred with compare match enabled by the IOB6 bit in timer I/O control register B (TIORb)

- Clearing (to 0) condition

- Writing 1 to CMFCB6 in timer status clear register B (TSCRb)
- Automatically cleared to 0 at an OCRB7 compare match when IREGB6 = 10.

Compare Match Flag B1 (CMFB1)

This status flag indicates occurrence of a compare match of output compare register B1 (OCRB1), output compare register B1S1 (OCRB1S1), output compare register B1S2 (OCRB1S2). When 1 is read from this flag, this indicates that a compare match has occurred in CMFB1. This flag cannot be set to 1 or 0 by the software.

The compare match for which this status flag is set is specified by the channel select bit (CHSELB).

- Setting (to 1) condition

CHSELB	Setting (to 1) conditions
00	When the TCNTB1 count value matches the OCRB1 value and the next PCLK has occurred
01	When the TCNTB1S1 count value matches the OCRB1S1 value and the next PCLK has occurred
10	When the TCNTB1S2 count value matches the OCRB1S2 value and the next PCLK has occurred
11 (Prohibited)	—

- Clearing (to 0) condition

Writing 1 to CMFCB1 in timer status clear register B (TSCRB)

Input Capture Flag B0 (ICFB0)

This status flag indicates occurrence of an input capture of input capture register B0 (ICRB0), input capture register B0S1 (ICRB0S1), or input capture register B0S2 (ICRB0S2). When 1 is read from this flag, it shows that an input capture has occurred in ICRB0, ICRB0S1, or ICRB0S2. This flag cannot be set to 1 or 0 by the software.

The input capture to set this flag is selected by channel select bit (CHSELB).

- Setting (to 1) condition

CHSELB	Setting (to 1) conditions
00	When the TCNTB0 value has been transferred to ICRB0 at an input capture trigger due to an external event 1_0
01	When the TCNTB0 value has been transferred to ICRB0 at an input capture trigger due to an external event 1_1
10	When the TCNTB0 value has been transferred to ICRB0 at an input capture trigger due to an external event 1_0
11 (Prohibited)	—

- Clearing (to 0) condition

Writing 1 to ICFCB0 in timer status clear register B (TSCRB)

Compare Match Flag B0 (CMFB0)

This status flag indicates occurrence of a compare match of output compare register B0 (OCRB0) or output compare register B0S1 (OCRB0S1) or output compare register B0S2 (OCRB0S2). When 1 is read from this flag, it shows that a compare match has occurred in CMFB0. This flag cannot be set to 1 or 0 by the software.

It's established by channel select bit (CHSELB) whether it's set when which comparing match occurred.

- Setting (to 1) condition

CHSELB	Setting (to 1) conditions
00	When the TCNTB0 count value matches the OCRB0 value and the next clock set by the prescaler has occurred
01	When the TCNTB0 count value matches the OCRB0S1 value and the next clock set by the prescaler has occurred
10	When the TCNTB0 count value matches the OCRB0S2 value and the next clock set by the prescaler has occurred
11 (Prohibited)	—

- Clearing (to 0) condition

Writing 1 to CMFCB0 in the timer status clear register B (TSCRB)

30.6.2.6 TSCR_B — Timer Status Clear Register B

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CMFCB 6E	CMFCB 6M	—	CMFCB 12	CMFCB 11	CMFCB 10	CMFCB 6	CMFCB 1	ICFCB0	CMFCB 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W	R	W	W	W	W	W	W	W

Table 30.44 TSCR_B Register Contents

Bit Position	Bit Name	Function
15 to 10	—	These bits are not used. Fix these bits to “0”.
9	CMFCB6E	Compare Match Flag Clear B6E Enable 0: Disabled (Default) 1: CMFB6E in timer status register B (TSRB) is cleared to 0.
8	CMFCB6M	Compare Match Flag Clear B6M Enable 0: Disabled (Default) 1: CMFB6M in timer status register B (TSRB) is cleared to 0.
7	—	This bit is not used. Fix this bit to “0”.
6	CMFCB12	Compare Match Flag Clear B12 Enable 0: Disabled (Default) 1: CMFB12 in timer status register B (TSRB) is cleared to 0.
5	CMFCB11	Compare Match Flag Clear B11 Enable 0: Disabled (Default) 1: CMFB11 in timer status register B (TSRB) is cleared to 0.
4	CMFCB10	Compare Match Flag Clear B10 Enable 0: Disabled (Default) 1: CMFB10 in timer status register B (TSRB) is cleared to 0.
3	CMFCB6	Compare Match Flag Clear B6 Enable 0: Disabled (Default) 1: CMFB6 in timer status register B (TSRB) is cleared to 0.
2	CMFCB1	Compare Match Flag Clear B1 Enable 0: Disabled (Default) 1: CMFB1 in timer status register B (TSRB) is cleared to 0.
1	ICFCB0	Input Capture Flag Clear B0 Enable 0: Disabled (Default) 1: ICFB0 in timer status register B (TSRB) is cleared to 0.
0	CMFCB0	Compare Match Flag Clear B0 Enable 0: Disabled (Default) 1: CMFB0 in timer status register B (TSRB) is cleared to 0.

The timer status clear register B (TSCR_B) is an 8-bit or 16-bit readable/writable register to set clearing of flags at occurrence of an input capture or compare match.

TSCR_B can be read and written in 16-bit units. When read, “0” is always returned.

TSCR_B is initialized to 0000_H after reset.

Compare Match Flag Clear B6E Enable (CMFCB6E)

Writing 1 to this bit while the compare match flag B6E (CMFB6E) in timer status register B (TSRB) is set to 1 clears CMFB6E to 0. This bit is always read as 0.

Compare Match Flag Clear B6M Enable (CMFCB6M)

Writing 1 to this bit while the compare match flag B6M (CMFB6M) in timer status register B (TSRB) is set to 1 clears CMFB6M to 0. This bit is always read as 0.

Compare Match Flag Clear B12 Enable (CMFCB12)

Writing 1 to this bit while the compare match flag B12 (CMFB12) in timer status register B (TSRB) is set to 1 clears CMFB12 to 0. This bit is always read as 0.

Compare Match Flag Clear B11 Enable (CMFCB11)

Writing 1 to this bit while the compare match flag B11 (CMFB11) in timer status register B (TSRB) is set to 1 clears CMFB11 to 0. This bit is always read as 0.

Compare Match Flag Clear B10 Enable (CMFCB10)

Writing 1 to this bit while the compare match flag B10 (CMFB10) in timer status register B (TSRB) is set to 1 clears CMFB10 to 0. This bit is always read as 0.

Compare Match Flag Clear B6 Enable (CMFCB6)

Writing 1 to this bit while the compare match flag B6 (CMFB6) in timer status register B (TSRB) is set to 1 clears CMFB6 to 0. This bit is always read as 0.

Compare Match Flag Clear B1 Enable (CMFCB1)

Writing 1 to this bit while the compare match flag B1 (CMFB1) in timer status register B (TSRB) is set to 1 clears CMFB1 to 0. This bit is always read as 0.

Input Capture Flag Clear B0 Enable (ICFCB0)

Writing 1 to this bit while the input capture flag B0 (ICFB0) in timer status register B (TSRB) is set to 1 clears ICFB0 to 0. This bit is always read as 0.

Compare Match Flag Clear B0 Enable (CMFCB0)

Writing 1 to this bit while the compare match flag B0 (CMFB0) in timer status register B (TSRB) is set to 1 clears CMFB0 to 0. This bit is always read as 0.

30.6.2.7 TICRB — Timer Interrupt Control Register B

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	IREGB6E[1:0]		IREGB6[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 30.45 TICRB Register Contents

Bit Position	Bit Name	Function
7 to 4	—	These bits are not used. Fix these bits to “0”.
3, 2	IREGB6E	Interrupt Enable Edge B6E 00: No interrupt request is output. 01: An interrupt request is output when CMFB6 is enabled and then CMFB6M is enabled or when CMFB6M is enabled and then CMFB6 is enabled (AND condition). 10: An interrupt request is output when CMFB6 or CMFB6M is enabled (OR condition) 11: Setting prohibited
1, 0	IREGB6	Interrupt Enable Edge B6 00: An interrupt request is output when CMFB6 is enabled 01: An interrupt request is output at the next external event selection (AGCK) input timing after CMFB6 is enabled. 10: An interrupt request is output at the second external event selection (AGCK) input timing after CMFB6 is enabled. However, no interrupt request is output if compare match B7 occurs before the second external event selection (AGCK) is input. 11: Setting prohibited

The timer interrupt control register B (TICRB) is an 8-bit readable/writable register to control the compare match interrupt request output timing.

TICRB can be read and written in 8-bit units.

TICRB is initialized to 00_H after reset.

Interrupt Enable Edge B6E (IREGB6E)

These bits select conditions for CMFB6E interrupt request output of TSRB. Operation of these bits is disabled if these bits are set to 00.

While these bits are set to 01, an interrupt request is output at the external event selection (AGCK) input timing at the time when CMFB6M is enabled while a CMFB6 interrupt by the IREGB6 setting is present or at the timing of a CMFB6 interrupt request by the IREGB6 setting with CMFB6M enabled (AND condition). If CMFB6 is cleared even after a CMFB6 interrupt has occurred, no CMFB6E interrupt request is output even while CMFB6M is enabled. Otherwise, if CMFB6M is cleared even while CMFB6M is enabled, a CMFB6 interrupt request by the IREGB6 setting occurs but no CMFB6E interrupt request is output.

While these bits are set to 10, an interrupt request is output when a CMFB6 interrupt by the IREGB6 setting occurs or when CMFB6M is enabled (OR condition). Note that, if the CMFB6 interrupt request timing by the IREGB6 setting differs from the CMFB6M interrupt request timing, respective interrupt requests occur.

However, if the CMFB6 interrupt request timing matches the CMFB6M interrupt request timing, an interrupt request occurs once.

Do not modify these bits while the counter is working or the CMFB6 or CMFB6M bit in TSRB is set to 1. If these bits are modified, this register does not function correctly.

Interrupt Enable Edge B6 (IREGB6)

These bits select CMFB6 interrupt request output timing of TSRB. While these bits are set to 00, a CMFB6 interrupt request is output when CMFB6 is enabled. While these bits are set to 01, an interrupt request is output at the first external event selection (AGCK) input timing after CMFB6 is enabled.

While these bits are set to 10, an interrupt request is output at the second external event selection (AGCK) input timing after CMFB6 is enabled. However, if compare match B7 occurs during the period of waiting for the second external event selection (AGCK) input, CMFB6 is automatically cleared and no interrupt request is output even if the second external event selection (AGCK) is input.

Do not modify these bits while the counter is working or the CMFB6 bit in TSRB is set to 1. If these bits are modified, this register does not function correctly.

30.6.2.8 TIERB — Timer Interrupt Enable Register B

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CMFB6EIE	CMFB6MIE	—	CMFB12IE	CMFB11IE	CMFB10IE	CMFB6IE	CMFB1IE	ICFB0IE	CMFB0IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.46 TIERB Register Contents

Bit Position	Bit Name	Function
15 to 10	—	These bits are not used. Fix these bits to "0".
9	CMFB6EIE	Compare Match B6E Interrupt Enable 0: Interrupt request by CMFB6E is disabled. 1: Interrupt request by CMFB6E is enabled.
8	CMFB6MIE	Compare Match B6M Interrupt Enable 0: Interrupt request by CMFB6M is disabled. 1: Interrupt request by CMFB6M is enabled.
7	—	This bit is not used. Fix this bit to "0".
6	CMFB12IE	Compare Match B12 Interrupt Enable 0: Interrupt request by CMFB12 is disabled. 1: Interrupt request by CMFB12 is enabled.
5	CMFB11IE	Compare Match B11 Interrupt Enable 0: Interrupt request by CMFB11 is disabled. 1: Interrupt request by CMFB11 is enabled.
4	CMFB10IE	Compare Match B10 Interrupt Enable 0: Interrupt request by CMFB10 is disabled. 1: Interrupt request by CMFB10 is enabled.
3	CMFB6IE	Compare Match B6 Interrupt Enable 0: Interrupt request by CMFB6 is disabled. 1: Interrupt request by CMFB6 is enabled.
2	CMFB1IE	Compare Match B1 Interrupt Enable 0: Interrupt request by CMFB1 is disabled. 1: Interrupt request by CMFB1 is enabled.
1	ICFB0IE	Input Capture B0 Interrupt Enable 0: Interrupt request by ICFB0 is disabled. 1: Interrupt request by ICFB0 is enabled.
0	CMFB0IE	Compare Match B0 Interrupt Enable 0: Interrupt request by CMFB0 is disabled. 1: Interrupt request by CMFB0 is enabled.

The timer interrupt enable register B (TIERB) is an 8-bit or 16-bit readable/writable register that enables or disables interrupt requests corresponding to the status flags of timer status register B.

TIERB is initialized to 0000_H after reset.

30.6.2.9 CHSELBR — Channel Select Register B

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CHSELB[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.47 CHSELBR Register Contents

Bit Position	Bit Name	Function
7 to 2	—	These bits are not used. When these bits are read, “0” is returned. When writing, write “0”.
1, 0	CHSELB[1:0]	Timer B Channel Select 00: Edge-interval measuring block 0 is selected. 01: Edge-interval measuring block 1 is selected. 10: Edge-interval measuring block 2 is selected. 11: Setting prohibited (Edge-interval measuring block 0 is selected, as same as 00 is specified.)

The channel select register B (CHSELBR) is an 8-bit readable/writable register.

CHSELBR is initialized to 00_H after reset.

Channel Select (CHSELB)

Select the counter to be compared with OCRB20 to OCRB43 for compare match, the output of the edge-interval measuring block to be used in frequency-multiplied clock generator, the source for setting input capture flag B0 (ICFB0), compare match flag B0 (CMFB0), compare match flag B1 (CMFB1), compare match flag B10 (CMFB10), compare match flag B11 (CMFB11) and compare match flag B12 (CMFB12).

30.6.2.10 TCNT4CRB — Multiplied-and-Corrected Clock Generating Counter B4 Control Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CU4SEL
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 30.48 TCNT4CRB Register Contents

Bit Position	Bit Name	Function
7 to 1	—	These bits are not used. When these bits are read, “0” is returned. When writing, write “0”.
0	CU4SEL	TCNTB4 count up setting. 0: TCNTB4 is stopped when TCNTB3 = TCNTB4. The clear value of TCNTB5 is H'00001000. (ATU-IV movement) 1: TCNTB4 is stopped when (TCNTB3 - 1) = TCNTB4. The clear value of TCNTB5 is H'00000000.

The Multiplied-and-Corrected clock generating counter B4 control register (TCNT4CRB) is an 8-bit readable/writable register.

TCNT4CRB is initialized to 00_H after reset.

TCNTB4 count up setting (CU4SEL)

This bit can select the count up conditions of Multiplied-and-Corrected Clock Counter B4 (TCNTB4) and the clear value of Multiplied-and-Corrected Clock Generating Counter B5 (TCNTB5).

CAUTION

When setting “1” to the CU4SEL bit of TCNT4CRB, be sure to set “1” to the CCS bit of Timer I/O Control Register B (TIORB).

30.6.2.11 TCNTB0 — Edge Interval Measuring Counter B0

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTB0[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTB0[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.49 TCNTB0 Register Contents

Bit Position	Bit Name	Function
31 to 0	CNTB0	Edge Interval Count These bits store 32-bit counter value.

TCNTB0 is a 32-bit readable/writable register that functions as a counter driven by the clock selected in the clock select bit B (CKSELB) of timer control register B (TCRB). TCNTB0 is cleared to 0000 0001_H on input capture by external input event 1_0.

TCNTB0 is started when the timer B enable bit (TBE) in ATU master enable register (ATUENR) is set to 1. Clearing the TBE bit to 0 stops the counting but the counter value is not cleared.

TCNTB0 can be read and written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

TCNTB0 is initialized to 0000 0001_H after reset.

30.6.2.12 TCNTB0S1 — Edge Interval Measuring Counter B0S1

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTB0S1[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTB0S1[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.50 TCNTB0S1 Register Contents

Bit Position	Bit Name	Function
31 to 0	CNTB0S1	Edge Interval Count These bits store 32-bit counter value.

TCNTB0S1 is a 32-bit readable/writable register that functions as a counter driven by the clock selected in the clock select bit B (CKSELB) of timer control register B (TCRB). TCNTB0S1 is cleared to 0000 0001_H on input capture by external input event 1_1.

TCNTB0S1 is started when the timer B enable bit (TBE) in ATU master enable register (ATUENR) is set to 1. Clearing the TBE bit to 0 stops the counting but the counter value is not cleared.

TCNTB0S1 can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

TCNTB0S1 is initialized to 0000 0001_H after reset.

30.6.2.13 TCNTB0S2 — Edge Interval Measuring Counter B0S2

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTB0S2[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTB0S2[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.51 TCNTB0S2 Register Contents

Bit Position	Bit Name	Function
31 to 0	CNTB0S2	Edge Interval Count These bits store 32-bit counter value.

TCNTB0S2 is a 32-bit readable/writable register that functions as a counter driven by the clock selected in the clock select bit B (CKSELB) of timer control register B (TCRB). TCNTB0S2 is cleared to 0000 0001_H on input capture by external input event 1_2.

TCNTB0S2 is started when the timer B enable bit (TBE) in ATU master enable register (ATUENR) is set to 1. Clearing the TBE bit to 0 stops the counting but the counter value is not cleared.

TCNTB0S2 can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

TCNTB0S2 is initialized to 0000 0001_H after reset.

**30.6.2.14 ICRB0 — Input Capture Register B0
DICRB0 — Input Capture Register B0
MIICRB0 — Mirror Input Capture Register B0**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICB0[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICB0[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.52 ICRB0 / DICRB0 Register Contents

Bit Position	Bit Name	Function
31 to 0	ICB0	Input Capture B0 These bits store 32-bit captured value.

ICRB0 is a 32-bit read-only register that is loaded with the value in TCNTB0 when external input event 1_0 is detected. At this time, ICFB0 in timer status register B (TSRB) is set to 1.

When the timer B channel select bits (CHSELB[1:0]) are set to “00”, an interrupt request can be output to the CPU at this ICRB0 input capture timing. TCNTB0 is cleared to 0000 0001_H.

ICRB0 can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

ICRB0 is initialized to 0000 0000_H after reset.

The read value of ICRB0 and DICRB0 is select by the set value of CHSELB[1:0], and when CHSELB[1:0] are set to “00”, the value of ICRB0 (edge-interval measuring block) is read. When CHSELB[1:0] are set to “01”, the value of ICRB0S1 (edge-interval measuring block 1) is read, and when CHSELB[1:0] are set to “10”, the value of ICRB0S2 (edge-interval measuring block 2) is read. The read value of MIICRB0 is always that of input capture register B0 (ICRB0).

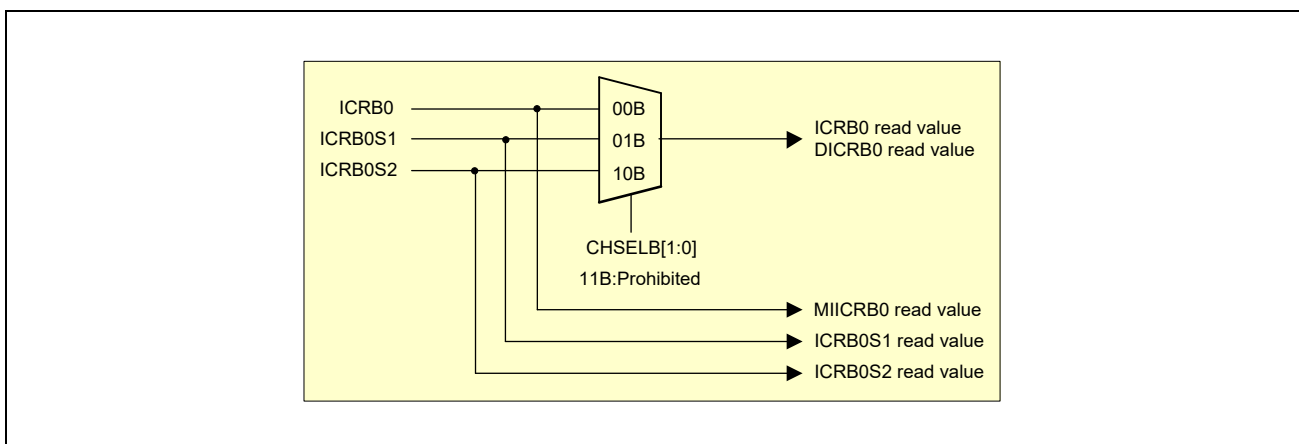


Figure 30.17 Input Capture Register B0

30.6.2.15 ICRB0S1 — Input Capture Register B0S1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICB0S1[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICB0S1[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.53 ICRB0S1 Register Contents

Bit Position	Bit Name	Function
31 to 0	ICB0S1	Input Capture B0S1 These bits store 32-bit captured value.

ICRB0S1 is a 32-bit read-only register that is loaded with the value in TCNTB0S1 when external input event 1_1 is detected.

When the timer B channel select bits (CHSELB[1:0]) are set to “01”, an interrupt request can be output to the CPU at this ICRB0S1 input capture timing. TCNTB0S1 is cleared to 0000 0001_H.

ICRB0S1 can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

ICRB0S1 is initialized to 0000 0000_H after reset.

A mirror register DICRB0 is provided to ICRB0S1. When the timer B channel select bits (CHSELB[1:0]) are set to “01”, the contents in ICRB0S1 and DICRB0 are the same.

30.6.2.16 ICRB0S2 — Input Capture Register B0S2

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICB0S2[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICB0S2[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.54 ICRB0S2 Register Contents

Bit Position	Bit Name	Function
31 to 0	ICB0S2	Input Capture B0S2 These bits store 32-bit captured value.

ICRB0S2 is a 32-bit read-only register that is loaded with the value in TCNTB0S2 when external input event 1_2 is detected.

When the timer B channel select bits (CHSELB[1:0]) are set to “10”, an interrupt request can be output to the CPU at this ICRB0S2 input capture timing. TCNTB0S2 is cleared to 0000 0001_H.

ICRB0S2 can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

ICRB0S2 is initialized to 0000 0000_H after reset.

A mirror register DICRB0 is provided to ICRB0S2. When the timer B channel select bits (CHSELB[1:0]) are set to “10”, the contents in ICRB0S2 and DICRB0 are the same.

30.6.2.17 RECRB1 to RECRB6 — Record Registers B1 to B6 DRECRB1 — Record Registers B1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RECRB1-6[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RECRB1-6[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.55 RECRB1 to RECRB6 / DRECRB1 Register Contents

Bit Position	Bit Name	Function
31 to 0	RECRB1 to 6	Record Register B1 to B6 These bits store 32-bit event capture value.

Record registers B1 to B6 (RECRB1 to RECRB6) are 32-bit registers that capture the ICRB0, ICRB0S1 or ICRB0S2 value at the timing of external event selection (AGCK) input (event 1 input). Which register RECRB1 captures is selected with the channel select bit (CHSELB).

CHSELB	Captured register RECRB1
00	ICRB0
01	ICRB0S1
10	ICRB0S2
11 (Prohibited)	—

Each time an external event selection (AGCK) is input, the ICRB0, ICRB0S1, ICRB0S2 value is captured by shifting the ICRB0 value to RECRB1, the RECRB1 value to RECRB2, and the RECRB2 value to RECRB3. Thus up to seven past TCNT0B, TCNT0BS1 or TCNT0BS2 values can be retained in ICRB0 and RECRB1 to RECRB6.

RECRB1 to RECRB6 can be read or written in 32-bit units only.

RECRB1 to RECRB6 are used as the calculation value for correction formula in sequencer control.

To initialize these registers, write 0000 0000_H to them.

RECRB1 to RECRB6 are initialized to 0000 0000_H after reset.

A mirror register DRECRB1 is provided to RECRB1. The contents in RECRB1 and DRECRB1 are the same.

30.6.2.18 RBURB0 to RBURB6 — Record Backup Register B0 to B6

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RBURB0-6[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBURB0-6[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.56 RBURB0 to RBURB6 Register Contents

Bit Position	Bit Name	Function
31 to 0	RBURB0-6	Record Backup Registers B0 to B6 These bits store 32-bit event capture backup value.

Record backup registers B0 to B6 (RBURB0 to RBURB6) are 32-bit registers that back up values of ICRB0 and RECRB1 to RECRB6 in this register at the timing of event compare match between “event counter B1 (TCNTB1) and OCRB12” or “event counter B1S1 (TCNTB1) and OCRB12S1” or “event counter B1S2 and OCRB12S2”.

Which register RBURB0 captures is selected with the channel select bit (CHSELB).

CHSELB	RBURB0 backup factor	Register of a backup target
00	Compare match of TCNTB1 and OCRB12	ICRB0
01	Compare match of TCNTB1S1 and OCRB12S1	ICRB0S1
10	Compare match of TCNTB1S2 and OCRB12S2	ICRB0S2
11 (Prohibited)	—	—

The ICRB0 value is retained in RBURB0, the RECRB1 value in RBURB1, the RECRB2 value in RBURB2, the RECRB3 value in RBURB3, the RECRB4 value in RBURB4, the RECRB5 value in RBURB5, and the RECRB6 value is retained in RBURB6 by compare match genesis timing of OCRB12, OCRB12S1 or OCRB12S2.

RBURB0 to RBURB6 can be read or written in 32-bit units only.

RBURB0 to RBURB6 are initialized to 0000 0000_H after reset.

30.6.2.19 ICRB30 to ICRB37 — Input Capture Registers B30 to B37

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	ICRB30-37[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 30.57 ICRB30 to ICRB37 Register Contents

Bit Position	Bit Name	Function
7 to 0	ICRB30 to ICRB37	Input Capture Registers B30 to B37 These bits store 8-bit event capture value.

Input capture registers B30 to B37 (ICRB30 to ICRB37) are 8-bit read-only registers. The event counter B1 (TCNTB1) value is captured at occurrence of any of seven external event inputs 1B to 1I that are input via timer A (ICRB30 to ICRB37). The event counter B1 (TCNTB1) is not cleared at occurrence of any of external event inputs 1B to 1I.

Which register ICRB30-37 captures is selected with the channel select bit (CHSELB).

CHSELB	Register of a backup target
00	TCNTB1
01	TCNTB1S1
10	TCNTB1S2
11 (Prohibited)	—

ICRB30 to ICRB37 can be read in 8-bit units only.

ICRB30 to ICRB37 are initialized to 00_H after reset.

30.6.2.20 OCRB0 — Output Compare Register B0

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OCB0[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCB0[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.58 OCRB0 Register Contents

Bit Position	Bit Name	Function
31 to 0	OCB0 Output Compare B0	These bits store 32-bit data to be compared with TCNTB0.

The output compare register B0 (OCRB0) is a 32-bit readable/writable register that is constantly compared with the input edge interval measuring counter B0 (TCNTB0). When the timer B channel select bits (CHSELB[1:0]) are set to “00” and the OCRB0 value matches the TCNTB0 value, an interrupt request due to this compare match can be output to the CPU and the CMFB0 bit in timer status register B (TSRB) is set to 1.

OCRB0 can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

OCRB0 is initialized to FFFF FFFF_H after reset.

30.6.2.21 OCRB0S1 — Output Compare Register B0S1

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OCB0S1[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCB0S1[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.59 OCRB0S1 Register Contents

Bit Position	Bit Name	Function
31 to 0	OCB0S1 Output Compare B0S1	These bits store 32-bit data to be compared with TCNTB0S1.

The output compare register B0S1 (OCRB0S1) is a 32-bit readable/writable register that is constantly compared with the input edge interval measuring counter B0S1 (TCNTB0S1). When the OCRB0S1 value matches the TCNTB0S1 value at CHSELB = 01_B, an interrupt request due to this compare match can be output to the CPU and the CMFB0 bit in timer status register B (TSRB) is set to 1.

OCRB0S1 can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

OCRB0S1 is initialized to FFFF FFFF_H after reset.

30.6.2.22 OCRB0S2 — Output Compare Register B0S2

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OCB0S2[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCB0S2[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.60 OCRB0S2 Register Contents

Bit Position	Bit Name	Function
31 to 0	OCB0S2 Output Compare B0S2	These bits store 32-bit data to be compared with TCNTB0S2.

The output compare register B0S2 (OCRB0S2) is a 32-bit readable/writable register that is constantly compared with the input edge interval measuring counter B0S2 (TCNTB0S2). When the OCRB0S2 value matches the TCNTB0S2 value at CHSELB = 10_B, an interrupt request due to this compare match can be output to the CPU and the CMFB0 bit in timer status register B (TSRB) is set to 1.

OCRB0S2 can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

OCRB0S2 is initialized to FFFF FFFF_H after reset.

30.6.2.23 TCNTB1 — Event Counter B1

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CNTB1[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.61 TCNTB1 Register Contents

Bit Position	Bit Name	Function
7 to 0	CNTB1	Event Count B1 These bits store 8-bit counter value.

The event counter B1 (TCNTB1) is an 8-bit readable/writable register that counts external event 1_0. According to the setting for the CLRB1 and CLRB1SEL bits in the timer control register B (TCRB), the TCNTB1 value can be cleared to 01_H at the next event input timing after a compare match between TCNTB1 and OCB10 or between TCNTB6 and OCB6 occurs.

Unless count operation is enabled by the TBE bit in the ATU master enable register (ATUENR), count operation is not performed even if an external event 1_0 is input.

TCNTB1 can be read or written in 8-bit units only.

TCNTB1 can be written only when TSECRB.EVSEQENB is “0”. Writing to TCNTB1 when TSECRB.EVSEQENB is “1” is prohibited.

TCNTB1 is initialized to 00_H after reset.

30.6.2.24 TCNTB1S1 — Event Counter B1S1

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CNTB1S1[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.62 TCNTB1S1 Register Contents

Bit Position	Bit Name	Function
7 to 0	CNTB1S1	Event Count B1S1 These bits store 8-bit counter value.

The event counter B1S1 (TCNTB1S1) is an 8-bit readable/writable register that counts external event 1_1. According to the setting for the CLRBS1 and CLRBS1SELS1 bits in the timer control register B (TCRB), the TCNTB1S1 value can be cleared to 01_H at the next event input timing after a compare match between TCNTB1S1 and OCRB10 or between TCNTB6 and OCRB6 occurs.

Unless count operation is enabled by the TBE bit in the ATU master enable register (ATUENR), count operation is not performed even if an external event 1_1 is input.

TCNTB1S1 can be read or written in 8-bit units only.

TCNTB1S1 can be written only when TSECRB.EVSEQENB is “0”. Writing to TCNTB1S1 when TSECRB.EVSEQENB is “1” is prohibited.

TCNTB1S1 is initialized to 00_H after reset.

30.6.2.25 TCNTB1S2 — Event Counter B1S2

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CNTB1S2[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.63 TCNTB1S2 Register Contents

Bit Position	Bit Name	Function
7 to 0	CNTB1S2	Event Count B1S2 These bits store 8-bit counter value.

The event counter B1S2 (TCNTB1S2) is an 8-bit readable/writable register that counts external event 1_2. According to the setting for the CLR B1S2 and CLR B1SEL S2 bits in the timer control register B (TCRB), the TCNTB1S2 value can be cleared to 01_H at the next event input timing after a compare match between TCNTB1S2 and OCRB10 or between TCNTB6 and OCRB6 occurs.

Unless count operation is enabled by the TBE bit in the ATU master enable register (ATUENR), count operation is not performed even if an external event 1_2 is input.

TCNTB1S2 can be read or written in 8-bit units only.

TCNTB1S2 can be written only when TSECRB.EVSEQENB is “0”. Writing to TCNTB1S2 when TSECRB.EVSEQENB is “1” is prohibited.

TCNTB1S2 is initialized to 00_H after reset.

30.6.2.26 OCRB1 — Output Compare Register B1

Value after reset: FF_H

Bit	7	6	5	4	3	2	1	0
	OCB1[7:0]							
Value after reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.64 OCRB1 Register Contents

Bit Position	Bit Name	Function
7 to 0	OCB1	Output Compare B1 These bits store 8-bit data to be compared with TCNTB1.

The output compare register B1 (OCRB1) is an 8-bit readable/writable register that is constantly compared with the event counter B1 (TCNTB1). When the timer B channel select bits (CHSELB[1:0]) are set to “00” and the OCRB1 value matches the TCNTB1 value, an interrupt request due to this compare match can be output to the CPU and the CMFB1 bit in timer status register B (TSRB) is set to 1 on the next PCLK.

The input capture register B1 (ICRB1) value is transferred to input capture register B2 (ICRB2) and ICRB1 is cleared at this compare match timing.

OCRB1 can be read or written in 8-bit units only.

OCRB1 is initialized to FF_H after reset.

30.6.2.27 OCRB1S1 — Output Compare Register B1S1

Value after reset: FF_H

Bit	7	6	5	4	3	2	1	0
	OCB1S1[7:0]							
Value after reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.65 OCRB1S1 Register Contents

Bit Position	Bit Name	Function
7 to 0	OCB1S1	Output Compare B1S1 These bits store 8-bit data to be compared with TCNTB1S1.

The output compare register B1S1 (OCRB1S1) is an 8-bit readable/writable register that is constantly compared with the event counter B1S1 (TCNTB1S1). When the timer B channel select bits (CHSELB[1:0]) are set to “01” and the OCRB1S1 value matches the TCNTB1S1 value, an interrupt request due to this compare match can be output to the CPU and the CMFB1 bit in timer status register B (TSRB) is set to 1 on the next PCLK.

The input capture register B1S1 (ICRB1S1) value is transferred to input capture register B2S1 (ICRB2S1) and ICRB1S1 is cleared at this compare match timing.

OCRB1S1 can be read or written in 8-bit units only.

OCRB1S1 is initialized to FF_H after reset.

30.6.2.28 OCRB1S2 — Output Compare Register B1S2

Value after reset: FF_H

Bit	7	6	5	4	3	2	1	0
	OCB1S2[7:0]							
Value after reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.66 OCRB1S2 Register Contents

Bit Position	Bit Name	Function
7 to 0	OCB1S2	Output Compare B1S2 These bits store 8-bit data to be compared with TCNTB1S2.

The output compare register B1S2 (OCRB1S2) is an 8-bit readable/writable register that is constantly compared with the event counter B1S2 (TCNTB1S2). When the timer B channel select bits (CHSELB[1:0]) are set to “10” and the OCRB1S2 value matches the TCNTB1S2 value, an interrupt request due to this compare match can be output to the CPU and the CMFB1 bit in timer status register B (TSRB) is set to 1 on the next PCLK.

The input capture register B1S2 (ICRB1S2) value is transferred to input capture register B2S2 (ICRB2S2) and ICRB1S2 is cleared at this compare match timing.

OCRB1S2 can be read or written in 8-bit units only.

OCRB1S2 is initialized to FF_H after reset.

30.6.2.29 OCRB10 — Output Compare Register B10

Value after reset: FF_H

Bit	7	6	5	4	3	2	1	0
	OCB10[7:0]							
Value after reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.67 OCRB10 Register Contents

Bit Position	Bit Name	Function
7 to 0	OCB10	Output Compare B10 These bits store 8-bit data to be compared with TCNTB1.

The output compare register B10 (OCRB10) is an 8-bit readable/writable register that is constantly compared with the event counter B1 (TCNTB1). When the timer B channel select bits (CHSELB[1:0]) are set to “00” and the OCRB10 value matches the TCNTB1 value, an interrupt request due to this compare match can be output to the CPU and the CMFB10 bit in timer status register B (TSRB) is set to 1 on the next PCLK.

According to the setting for the CLRB1 and CLRB1SEL bits in timer control register B (TCRB), the TCNTB1 value can be cleared to 01_H at the next event input timing after this compare match occurs. Furthermore, according to the setting for the CLRB3 and CLRB3SEL bits in timer control register B (TCRB), the TCNTB3 value can be cleared to 0000 0000_H at the next PCLK timing after this compare match occurs.

OCRB10 can be read or written in 8-bit units only.

OCRB10 is initialized to FF_H after reset.

30.6.2.30 OCRB10S1 — Output Compare Register B10S1

Value after reset: FF_H

Bit	7	6	5	4	3	2	1	0
	OCB10S1[7:0]							
Value after reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.68 OCRB10S1 Register Contents

Bit Position	Bit Name	Function
7 to 0	OCB10S1	Output Compare B10S1 These bits store 8-bit data to be compared with TCNTB1S1.

The output compare register B10S1 (OCRB10S1) is an 8-bit readable/writable register that is constantly compared with the event counter B1S1 (TCNTB1S1). When the timer B channel select bits (CHSELB[1:0]) are set to “01” and the OCRB10S1 value matches the TCNTB1S1 value, an interrupt request due to this compare match can be output to the CPU and the CMFB10 bit in timer status register B (TSRB) is set to 1 on the next PCLK.

According to the setting for the CLRB1S1 and CLRB1SELS1 bits in timer control register BS1 (TCRBS1), the TCNTB1S1 value can be cleared to 01_H at the next event input timing after this compare match occurs.

OCRB10S1 can be read or written in 8-bit units only.

OCRB10S1 is initialized to FF_H after reset.

30.6.2.31 OCRB10S2 — Output Compare Register B10S2

Value after reset: FF_H

Bit	7	6	5	4	3	2	1	0
	OCB10S2[7:0]							
Value after reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.69 OCRB10S2 Register Contents

Bit Position	Bit Name	Function
7 to 0	OCB10S2	Output Compare B10S2 These bits store 8-bit data to be compared with TCNTB1S2.

The output compare register B10S2 (OCRB10S2) is an 8-bit readable/writable register that is constantly compared with the event counter B1S2 (TCNTB1S2). When the timer B channel select bits (CHSELB[1:0]) are set to “10” and the OCRB10S2 value matches the TCNTB1S2 value, an interrupt request due to this compare match can be output to the CPU and the CMFB10 bit in timer status register B (TSRB) is set to 1 on the next PCLK.

According to the setting for the CLRB1S2 and CLRB1SELS2 bits in timer control register BS2 (TCRBS2), the TCNTB1S2 value can be cleared to 01_H at the next event input timing after this compare match occurs.

OCRB10S2 can be read or written in 8-bit units only.

OCRB10S2 is initialized to FF_H after reset.

30.6.2.32 OCRB11 — Output Compare Register B11

Value after reset: FF_H

Bit	7	6	5	4	3	2	1	0
	OCB11[7:0]							
Value after reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.70 OCRB11 Register Contents

Bit Position	Bit Name	Function
7 to 0	OCB11	Output Compare B11 These bits store 8-bit data to be compared.

The output compare register B11 (OCRB11) is an 8-bit readable/writable register. According to the setting for the PIMRSEL bit in timer control register B (TCRB), when the timer B channel select bits (CHSELB[1:0]) are set to “00”, this register can operate by using PIMR2 for down-count values to be loaded to RLDB and for down-count values of TCNTB2 during a compare match between TCNTB1 and OCRB11. PIMR1 is always used for up-count values of TCNTB3.

When the timer B channel select bits (CHSELB[1:0]) are set to “00” and the TCNTB1 value matches the OCRB11 value, an interrupt request due to this compare match can be output to the CPU and the CMFB11 bit in timer status register B (TSRB) is set to 1 on the next PCLK.

OCRB11 can be read or written in 8-bit units only.

OCRB11 is initialized to FF_H after reset.

30.6.2.33 OCRB11S1 — Output Compare Register B11S1

Value after reset: FF_H

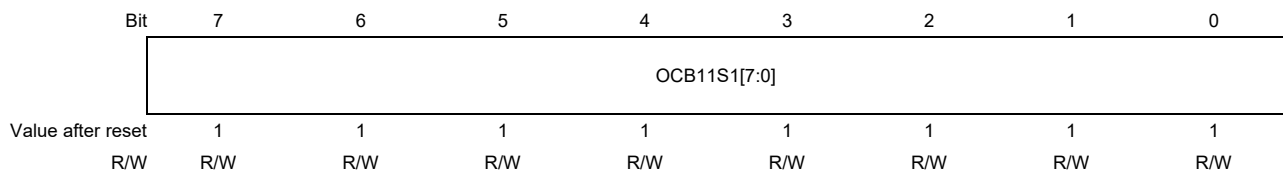


Table 30.71 OCRB11S1 Register Contents

Bit Position	Bit Name	Function
7 to 0	OCB11S1	Output Compare B11S1 These bits store 8-bit data to be compared.

The output compare register B11S1 (OCRB11S1) is an 8-bit readable/writable register. According to the setting for the PIMRSEL bit in timer control register B (TCRB), when the timer B channel select bits (CHSELB[1:0]) are set to “01”, this register can operate by using PIMR2 for down-count values to be loaded to RLDB and for down-count values of TCNTB2 during a compare match between TCNTB1S1 and OCRB11S1. PIMR1 is always used for up-count values of TCNTB3.

When the timer B channel select bits (CHSELB[1:0]) are set to “01” and the TCNTB1S1 value matches the OCRB11S1 value, an interrupt request due to this compare match can be output to the CPU and the CMFB11S1 bit in timer status register B (TSRB) is set to 1 on the next PCLK.

OCRB11S1 can be read or written in 8-bit units only.

OCRB11S1 is initialized to FF_H after reset.

30.6.2.34 OCRB11S2 — Output Compare Register B11S2

Value after reset: FF_H

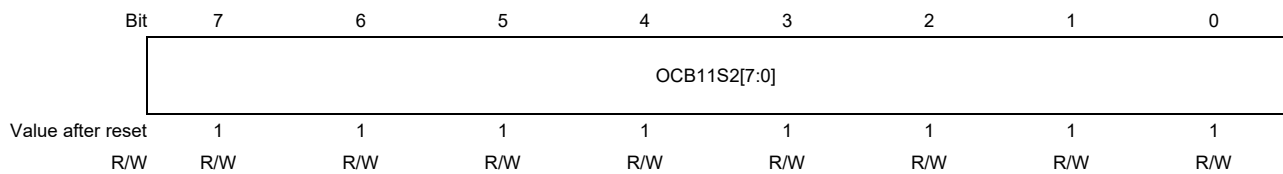


Table 30.72 OCRB11S2 Register Contents

Bit Position	Bit Name	Function
7 to 0	OCB11S2	Output Compare B11S2 These bits store 8-bit data to be compared.

The output compare register B11S2 (OCRB11S2) is an 8-bit readable/writable register. According to the setting for the PIMRSEL bit in timer control register B (TCRB), when the timer B channel select bits (CHSELB[1:0]) are set to “10”, this register can operate by using PIMR2 for down-count values to be loaded to RLDB and for down-count values of TCNTB2 during a compare match between TCNTB1S2 and OCRB11S2. PIMR1 is always used for up-count values of TCNTB3.

When the timer B channel select bits (CHSELB[1:0]) are set to “10” and the TCNTB1S2 value matches the OCRB11S2 value, an interrupt request due to this compare match can be output to the CPU and the CMFB11S2 bit in timer status register B (TSRB) is set to 1 on the next PCLK.

OCRB11S2 can be read or written in 8-bit units only.

OCRB11S2 is initialized to FF_H after reset.

30.6.2.35 OCRB12 — Output Compare Register B12

Value after reset: FF_H

Bit	7	6	5	4	3	2	1	0
	OCB12[7:0]							
Value after reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.73 OCRB12 Register Contents

Bit Position	Bit Name	Function
7 to 0	OCB12	Output Compare B12 These bits store 8-bit data to be compared.

The output compare register B12 (OCRB12) is an 8-bit readable/writable register. When the timer B channel select bits (CHSELB[1:0]) are set to “00”, values of ICRB0 and RECRB1 to RECRB6 can be backed up in RBURB0 to RBURB6 at the timing of event compare match between the event counter B1 (TCNTB1) and this register.

CHSELB[1:0] are set to “00” and the TCNTB1 value matches the OCRB12 value, an interrupt request due to this compare match can be output to the CPU and the CMFB12 bit in timer status register B (TSRB) is set to 1 on the next PCLK.

OCRB12 can be read or written in 8-bit units only.

OCRB12 is initialized to FF_H after reset.

30.6.2.36 OCRB12S1 — Output Compare Register B12S1

Value after reset: FF_H

Bit	7	6	5	4	3	2	1	0
	OCB12S1[7:0]							
Value after reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.74 OCRB12S1 Register Contents

Bit Position	Bit Name	Function
7 to 0	OCB12S1	Output Compare B12S1 These bits store 8-bit data to be compared.

The output compare register B12S1 (OCRB12S1) is an 8-bit readable/writable register. When the timer B channel select bits (CHSELB[1:0]) are set to “01”, values of ICRB0S1 and RECRB1 to RECRB6 can be backed up in RBURB0 to RBURB6 at the timing of event compare match between the event counter B1S1 (TCNTB1S1) and this register.

CHSELB[1:0] are set to “01” and the TCNTB1S1 value matches the OCRB12S1 value, an interrupt request due to this compare match can be output to the CPU and the CMFB12 bit in timer status register B (TSRB) is set to 1 on the next PCLK.

OCRB12S1 can be read or written in 8-bit units only.

OCRB12S1 is initialized to FF_H after reset.

30.6.2.37 OCRB12S2 — Output Compare Register B12S2

Value after reset: FF_H

Bit	7	6	5	4	3	2	1	0
	OCB12S2[7:0]							
Value after reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.75 OCRB12S2 Register Contents

Bit Position	Bit Name	Function
7 to 0	OCB12S2	Output Compare B12S2 These bits store 8-bit data to be compared.

The output compare register B12S2 (OCRB12S2) is an 8-bit readable/writable register. When the timer B channel select bits (CHSELB[1:0]) are set to “10”, values of ICRB0S2 and RECRB1 to RECRB6 can be backed up in RBURB0 to RBURB6 at the timing of event compare match between the event counter B1S2 (TCNTB1S2) and this register.

CHSELB[1:0] are set to “10” and the TCNTB1S1 value matches the OCRB12S2 value, an interrupt request due to this compare match can be output to the CPU and the CMFB12 bit in timer status register B (TSRB) is set to 1 on the next PCLK.

OCRB12S2 can be read or written in 8-bit units only.

OCRB12S2 is initialized to FF_H after reset.

30.6.2.38 ICRB1 — Input Capture Register B1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICB1[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICB1[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.76 ICRB1 Register Contents

Bit Position	Bit Name	Function
31 to 0	ICB1	Input Capture B1 These bits store 32-bit input capture value.

ICRB1 is a 32-bit read-only register. ICRB1 accumulates and stores the input inter-edge measuring counter B0 (TCNTB0) value in ICRB1 at the input timing of the external event 1_0. ICRB1 is cleared to 0000 0000_H on compare match between the event counter B1 (TCNTB1) and output compare register B1 (OCRB1).

ICRB1 can be read in 32-bit units only. Do not read 8-bit or 16-bit data for this register.

ICRB1 is initialized to 0000 0000_H after reset.

30.6.2.39 ICRB1S1 — Input Capture Register B1S1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICB1S1[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICB1S1[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.77 ICRB1S1 Register Contents

Bit Position	Bit Name	Function
31 to 0	ICB1S1	Input Capture B1S1 These bits store 32-bit input capture value.

ICRB1S1 is a 32-bit read-only register. ICRB1S1 accumulates and stores the input inter-edge measuring counter B0S1 (TCNTB0S1) value in ICRB1S1 at the input timing of the external event 1_1. ICRB1S1 is cleared to 0000 0000_H on compare match between the event counter B1S1 (TCNTB1S1) and output compare register B1S1 (OCRB1S1).

ICRB1S1 can be read in 32-bit units only. Do not read 8-bit or 16-bit data for this register.

ICRB1S1 is initialized to 0000 0000_H after reset.

30.6.2.40 ICRB1S2 — Input Capture Register B1S2

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICB1S2[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICB1S2[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.78 ICRB1S2 Register Contents

Bit Position	Bit Name	Function
31 to 0	ICB1S2	Input Capture B1S2 These bits store 32-bit input capture value.

ICRB1S2 is a 32-bit read-only register. ICRB1S2 accumulates and stores the input inter-edge measuring counter B0S2 (TCNTB0S2) value in ICRB1S2 at the input timing of the external event 1_2. ICRB1S2 is cleared to 0000 0000_H on compare match between the event counter B1S2 (TCNTB1S2) and output compare register B1S2 (OCRB1S2).

ICRB1S2 can be read in 32-bit units only. Do not read 8-bit or 16-bit data for this register.

ICRB1S2 is initialized to 0000 0000_H after reset.

30.6.2.41 ICRB2 — Input Capture Register B2

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICB2[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICB2[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.79 ICRB2 Register Contents

Bit Position	Bit Name	Function
31 to 0	ICB2	Input Capture B2 These bits store 32-bit input capture value.

ICRB2 is a 32-bit read-only register. ICRB1 keeps a running total of the value in input capture register B1 (ICRB1) on compare match between event counter B1 (TCNTB1) and output compare register B1 (OCRB1).

ICRB2 can be read in 32-bit units only. Do not read 8-bit or 16-bit data for this register.

ICRB2 is initialized to 0000 0000_H after reset.

30.6.2.42 ICRB2S1 — Input Capture Register B2S1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICB2S1[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICB2S1[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.80 ICRB2S1 Register Contents

Bit Position	Bit Name	Function
31 to 0	ICB2S1	Input Capture B2S1 These bits store 32-bit input capture value.

ICRB2S1 is a 32-bit read-only register. ICRB1S1 keeps a running total of the value in input capture register B1S1 (ICRB1S1) on compare match between event counter B1S1 (TCNTB1S1) and output compare register B1S1 (OCRB1S1).

ICRB2S1 can be read in 32-bit units only. Do not read 8-bit or 16-bit data for this register.

ICRB2S1 is initialized to 0000 0000_H after reset.

30.6.2.43 ICRB2S2 — Input Capture Register B2S2

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICB2S2[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICB2S2[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.81 ICRB2S2 Register Contents

Bit Position	Bit Name	Function
31 to 0	ICB2S2	Input Capture B2S2 These bits store 32-bit input capture value.

ICRB2S2 is a 32-bit read-only register. ICRB1S2 keeps a running total of the value in input capture register B1S2 (ICRB1S2) on compare match between event counter B1S2 (TCNTB1S2) and output compare register B1S2 (OCRB1S2).

ICRB2S2 can be read in 32-bit units only. Do not read 8-bit or 16-bit data for this register.

ICRB2S2 is initialized to 0000 0000_H after reset.

30.6.2.44 LDB — Load Register B

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	LDVAL[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LDVAL[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.82 LDB Register Contents

Bit Position	Bit Name	Function
31 to 24	—	These bits are not used. Fix these bits to “0”.
23 to 0	LDVAL	Load Value These bits store 24-bit data to be loaded to TCNTB2 and RLDB.

LDB is a 32-bit readable/writable register that is mapped to the lower-order 24 bits on a 32-bit boundary. The lower 24 bits are available.

When the LDSEL bit in timer I/O control register B (TIORB) is set to 1, the value in this register is loaded to reloadable counter B2 (TCNTB2) and reload register B (RLDB).

LDB can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

LDB is initialized to 0000 0000_H after reset.

30.6.2.45 RLDB — Reload Register B

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RLDVAL[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RLDVAL[15:8]								—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 30.83 RLDB Register Contents

Bit Position	Bit Name	Function
31 to 8	RLDVAL	Reload Value These bits store 24-bit reload value.
7 to 0	—	These bits are not used. Fix these bits to “0”.

RLDB is a 32-bit readable/writable register that is aligned with a longword boundary. The upper 24 bits are available.

This register is updated when the externally input event selection (AGCK) is detected while the LDEN bit in timer I/O control register B (TIORB) is cleared to 0.

The value in input capture register B0 (ICRB0)*¹ or load register B (LDB) minus the value in the PIMR bits in PIM is used for updating. According to the setting for the PIMRSEL bit in timer control register B (TCRB) and a compare match between TCNTB1 and OCRB11*², PIMR2 is used as the PIMR value. In other cases, PIMR1 is used as the PIMR value. ICRB0*¹ or LDB is set by the LDSEL bit in TIORB. For subtraction on ICRB0*¹ and PIMR, the lower 24 bits of ICRB0*¹ and PIM which is zero-extended.

The value in this register is added to the value in TCNTB2 on the first counter clock after the value in bits reload count B2 (CNTB2) is equal to or less than the value in the pulse interval multiplier register (PIM).

RLDB can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

RLDB is initialized to 0000 0000_H after reset.

Initial setting is required for RLDB before starting timer B counter operation. For details, see **Section 30.6.3, Details of Operation**.

Note 1. CHSELB is used for ICRB0 at 00_B, and CHSELB is used for ICRB0S1 at 01_B, and CHSELB is used for ICRB0S2 at 10_B.

Note 2. CHSELB moves at 00_B for a compare match of TCNTB1 and OCRB11, and CHSELB moves at 01_B for a compare match of TCNTB1S1 and OCRB11S1, and CHSELB moves at 10_B for a compare match of TCNTB1S2 and OCRB11S2.

30.6.2.46 TCNTB2 — Reloadable Counter B2

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTB2[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTB2[15:8]								—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 30.84 TCNTB2 Register Contents

Bit Position	Bit Name	Function
31 to 8	CNTB2	Reload Counter B2 These bits store 24-bit reload counter value.
7 to 0	—	These bits are not used. Fix these bits to "0".

TCNTB2 is a down counter mapped to 24 bits in a 32-bit readable and writable register and functions as a counter driven by the clock selected by the clock selection B bits (CKSELB0 to CKSELB2) in timer control register B (TCRB). Each decrementation by the value set in the pulse interval multiplier register (PIMR). According to the setting for the PIMRSEL bit in timer control register B (TCRB) and in response to a compare match between TCNTB1 and OCB11*¹, PIMR2 is used as the PIMR value. In other cases, PIMR1 is used as the PIMR value.

When the TBE bit in ATU master enable register (ATUENR) is set to 1, counting is not performed. Even if the TBE bit is cleared to 0, this counter is not cleared.

The TCNTB2 value is updated when the externally input event selection (AGCK) is detected while the LEDN bit in timer I/O control register B (TIORB) is cleared to 0.

The value in input capture register B0 (ICRB0)*² or load register B (LDB) is used for updating. ICRB0*² or LDB can be set by the LDSEL bit in TIORB.

The value in reload register B (RLDB) is added to the value stored in this counter on the first counter clock cycle after the value in this counter is equal to or less than the value in PIM. The frequency-multiplied clock (AGCK1) is output as a train of single pulses equal in width to the cycle of the PCLK clock on reloading.

TCNTB2 can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

TCNTB2 is initialized to 0000 0000_H after reset.

Initial setting is required for TCNTB2 before starting timer B counter operation. For details, see **Section 30.6.3, Details of Operation**.

Note 1. CHSELB uses PIMR2 by a compare match of TCNTB1 and OCB11 at 00_B, the compare match by which CHSELB is TCNTB1S1 and OCB11S1 at 01_B, the compare match of TCNTB1S2 and OCB11S2 at 10_B.

Note 2. CHSELB uses ICRB0 as renewal data at 00_B. When CHSELB is 01_B, ICRB0S1 is used, and CHSELB uses ICRB0S2 as renewal data at 10_B.

30.6.2.47 PIMR1 — Pulse Interval Multiplier Register 1

Value after reset: 0001_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PIM1[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.85 PIMR1 Register Contents

Bit Position	Bit Name	Function
15 to 12	—	These bits are not used. Fix these bits to “0”.
11 to 0	PIM1	Pulse Interval Multiplier 1 These bits set the multiplication ratio for the frequency-multiplied clock. The settable value ranges from 1 to 4095.

PIMR1 is a 16-bit readable/writable register that sets the multiplication ratio of the externally input event for generation of the frequency-multiplied clock.

The settable value ranges from 1 (001_H) to 4095 (FFF_H). Do not set the PIM bits to 000_H. If the PIM bits are set to 000_H, operation cannot be guaranteed.

The value in this register is used in various registers: as the step size in decrementation of reloadable counter B2 (TCNTB2); for calculation of the value to be input to reload register B (RLDB); and for calculation of the value to be input to corrected event counter B3 (TCNTB3). However, when a compare match between TCNTB1 and OCRB11*¹ occurs with the PIMRSEL bit in timer control register B (TCRB) set to 1, PIMR2 is used to calculate the TCNTB2 down-count value and the RLDB input value. PIMR1 is always used to calculate the TCNTB3 input value

PIMR1 can be read or written in 16-bit units only. Do not write or read 8-bit data for this register.

PIMR1 is initialized to 0001_H after reset.

Note 1. CHSELB uses PIMR2 by a compare match of TCNTB1 and OCRB11 at 00_B, the compare match by which CHSELB is TCNTB1S1 and OCRB11S1 at 01_B, the compare match of TCNTB1S2 and OCRB11S2 at 10_B.

30.6.2.48 PIMR2 — Pulse Interval Multiplier Register 2

Value after reset: 0001_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PIM2[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.86 PIMR2 Register Contents

Bit Position	Bit Name	Function
15 to 12	—	These bits are not used. Fix these bits to “0”.
11 to 0	PIM2	Pulse Interval Multiplier 2 These bits set the multiplication ratio for the frequency-multiplied clock. The settable value ranges from 1 to 4095.

The pulse interval multiplier register 2 (PIMR2) is a 16-bit readable/writable register that is mapped to the lower 12 bits. PIMR2 sets the multiplication ratio of the frequency-multiplied clock when tooth is missing.

The settable value ranges from 1 (001_H) to 4095 (FFF_H). Do not set the PIM2 bits to 000_H. If the PIM2 bits are set to 000_H, operation cannot be guaranteed.

By setting the PIMRSEL bit in timer control register B (TCRB), the reload counter B2 (TCNTB2) performs down-counting with the set PIMR2 value (PIM2) (not with the PIMR1 value) while the TCNTB1 value is equal to the OCRB11 value*¹. The PIMR2 value is also used to calculate the reload register B (RLDB) input value, but is not used to calculate the correction event counter B3 (TCNTB3) input value. The multiplication ratio of the frequency-multiplied clock at missing tooth is corrected by setting a value of (N (number of missing teeth) + 1) times for the PIMR1 value. When the number of missing teeth is 2, for example, a value of (PIMR1 value × 3) is set for PIMR2.

PIMR2 can be read or written in 16-bit units only. Do not write or read 8-bit data for this register.

PIMR2 is initialized to 0001_H after reset.

Note 1. CHSELB uses PIMR2 by a compare match of TCNTB1 and OCRB11 at 00_B, the compare match by which CHSELB is TCNTB1S1 and OCRB11S1 at 01_B, the compare match of TCNTB1S2 and OCRB11S2 at 10_B.

30.6.2.49 TCNTB6 — Multiplied Clock Counter B6

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTB6[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTB6[15:12]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.87 TCNTB6 Register Contents

Bit Position	Bit Name	Function
31 to 12	CNTB6	Frequency-Multiplied Clock Count B6 These bit store 20-bit counter value driven by the frequency-multiplied clock.
11 to 0	—	These bits are not used. Fix these bits to "0".

The multiplied clock counter B6 (TCNTB6) is a 20-bit up-counter mapped in the 32-bit readable/ writable register. TCNTB6 performs up-counting on the frequency-multiplied clock (AGCK1). This counter is cleared to 0000_H when the external input event selection (AGCK) is detected.

When the TBE bit in ATU master enable register (ATUENR) is set to 1, counting is not performed. Even if the TBE bit is cleared to 0 and counting up is stopped, the counter value is not cleared.

TCNTB6 can be read or written in 32-bit units only. Do not write or read 16-bit data for this register.

TCNTB6 is initialized to 0000 0000_H after reset.

30.6.2.50 ICRB6 — Input Capture Register B6

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICB6[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICB6[15:12]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.88 ICRB6 Register Contents

Bit Position	Bit Name	Function
31 to 12	ICB6	Input Capture B6 These bits store 20-bit input capture value.
11 to 0	—	These bits are not used. Fix these bits to "0".

The input capture register B6 (ICRB6) is a 32-bit read-only register that is mapped to upper 20 bits. No value can be written to this register.

When an external event selection (AGCK) input is detected, ICRB6 stores the TCNTB6 value at the next clock (PCLK) timing. The value stored in ICRB6 is compared with the TCNTB6M value at the next external event selection input timing. When the ICRB6 value is larger than the TCNTB6M value, an interrupt request can be output.

ICRB6 can be read in 32-bit units only.

ICRB6 is initialized to 0000 0000_H after reset.

30.6.2.51 RARB6 — Multiplication Setting Register B6

Value after reset: 40_H

Bit	7	6	5	4	3	2	1	0
	RARB6[7:0]							
Value after reset	0	1	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.89 RARB6 Register Contents

Bit Position	Bit Name	Function
7 to 0	RARB6	Multiplication Setting Register B6 These bits set the TCNTB6M count value.

The multiplication setting register B6 (RARB6) is an 8-bit readable/writable register to set an up-count value of the frequency-multiplied clock counter B6M (TCNTB6M). RARB6 handles the lower 6 bits as fixed point. A multiplication of 1 to 3.984375 can be set with a resolution of 1/64.

RARB6 can be read or written in 8-bit units only.

RARB6 is initialized to 40_H after reset.

30.6.2.52 TCNTB6M — Frequency-Multiplied Clock Counter B6M

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTB6M[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTB6M[15:6]										—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Table 30.90 TCNTB6M Register Contents

Bit Position	Bit Name	Function
31 to 6	CNTB6M	Frequency-Multiplied Clock Count B6M 26-bit frequency-multiplied clock count value
5 to 0	—	These bits are not used. Fix these bits to “0”.

The frequency-multiplied clock counter B6M (TCNTB6M) is a 32-bit readable/writable register (up-counter) that is mapped to upper 26 bits. TCNTB6M performs count-up operation with a count value of multiplication that is set in multiplication setting register B6 (RARB6) for the frequency-multiplied clock (AGCK1). TCNTB6M handles the lower 6 bits (b11 to b6) as fixed point, and can count within a range of 1 to 1048575.984375 with a resolution of 1/64. Each time an external event selection (AGCK) is input, the TCNTB6M value is compared with the ICRB6 value. When ICRB6 is larger than TCNTB6M, an interrupt request can be output. TCNTB6M is initialized to 0000 0000_H at the first clock (PCLK) timing after occurrence of an external event selection input.

Setting the TBE bit in the ATU master enable register (ATUENR) to 1 starts counting of this counter. Clearing the TBE bit to 0 stops counting but the counter value is not cleared.

TCNTB6M can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

TCNTB6M is initialized to 0000 0000_H after reset.

30.6.2.53 OCRB6 — Output Compare Register B6

Value after reset: FFFF F000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OCB6[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCB6[15:12]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.91 OCRB6 Register Contents

Bit Position	Bit Name	Function
31 to 12	OCB6	Output Compare B6 These bits store 20-bit data to be compared
11 to 0	—	These bits are not used. Fix these bits to "0".

OCRB6 is a 32-bit readable/writable register in which output compare B6 (OCB6) is mapped to upper 20 bits in 32 bits. Whether to perform compare match between multiplied-clock counter B6 and OCRB6 can be selected by the IOB6 setting of timer I/O control register B (TIORB).

The OCRB6 value is compared with the TCNTB6 value. When input of the frequency-multiplied clock signal (AGCK1) leads to the values matching, a missing tooth detection trigger is generated, which leads to setting of the CMFB6 bit in timer status register B (TSRB) to 1 and can also cause the output of an interrupt request for the CPU.

According to the setting for the CLRB1 and CLRB1SEL bits in the timer control register B (TCRB) and this compare match occurrence, the TCNTB1, TCNT1S1 and TCNT1S2 value can be cleared to 01_H at the next event input timing. Furthermore, according to the setting for the CLRB3 and CLRB3SEL bits in the timer control register B (TCRB) and this compare match occurrence, the TCNTB3 value can be cleared to 0000 0000_H at the next PCLK timing.

OCRB6 can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

OCRB6 is initialized to FFFF F000_H after reset.

30.6.2.54 OCRB7 — Output Compare Register B7

Value after reset: FFFF F000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OCB7[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCB7[15:12]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.92 OCRB7 Register Contents

Bit Position	Bit Name	Function
31 to 12	OCB7	Output Compare B7 These bit store 20-bit data to be compared.
11 to 0	—	These bits are not used. Fix these bits to “0”.

OCRB7 is a 32-bit readable/writable register in which output compare B7 (OCB7) is mapped to upper 20 bits in 32 bits.

The OCRB7 value is compared with the TCNTB6 value. When the frequency-multiplied clock (AGCK1) is input with these values matching, a compare match occurs. When the IREGB6 bits in timer interrupt control register B (TICRB) are set to 10, the CMFB6 bit in timer status register B (TSRB) is cleared to 0 at an occurrence of this compare match.

There is no status flag or interrupt request that indicates occurrence of a compare match of OCRB7.

OCRB7 can be read or written in 32-bit units only. Do not write or read 16-bit data for this register.

OCRB7 is initialized to FFFF F000_H after reset.

30.6.2.55 OCRB20 to OCRB43 — Output Compare Registers B20 to 43

Value after reset: FF_H

Bit	7	6	5	4	3	2	1	0
	OCB20-43[7:0]							
Value after reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.93 OCRB20 to OCRB43 Register Contents

Bit Position	Bit Name	Function
7 to 0	OCB20 to OCB43	Output Compare B20 to B43 These bits store 8-bit value to be compared.

The output compare registers B20 to B43 (OCRB20 to OCRB43) are 8-bit readable/writable register.

During timer B operation, one of OCRB20 to OCRB43 that is selected by the sequencer is constantly compared with TCNTB1, and if its value matches the TCNTB1 value, the sequencer transits at the next external event selection (AGCK).

There is no status flag or interrupt request that indicates occurrence of a compare match of OCRB20 to OCRB43.

OCRB20 to OCR43 can be read or written in 8-bit units only.

OCRB20 to OCRB43 are initialized to FF_H after reset.

The register that is compared with TCNTB1*¹ functions as following according to the status of the sequencer.

If the sequencer status bit (SEQB) in the timer sequencer register (TSEQRB) is “x”, the output compare register OCRB (20 + x – 1) is compared with TCNTB1 (x = 1 to 24).

While the sequencer is running, it is prohibited to rewrite the compare registers corresponding to the sequence numbers enabled by the timer sequencer enable bits TSEQNEN[23:0].

Compare register/Compare match	Transition of sequencer
OCRB20 compare match	The sequencer transits at the first external event selection (AGCK) after compare match. Example: State 0 → State 1
:	
OCRB43 compare match	

Note 1. When CHSELB is 00_B, TCNTB1 counter compares, and when CHSELB is 01_B, TCNTB1S1 counter compares, and when CHSELB is 10_B, TCNTB1S2 counter compares.

30.6.2.56 TSEQCRB — Timer Sequencer Control Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	EVSEQENB
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 30.94 TSEQCRB Register Contents

Bit Position	Bit Name	Function
7 to 1	—	These bits are not used. Fix these bits to “0”.
0	EVSEQENB	Event Input Cycle Correction Sequencer Enable 0: Sequencer operation is disabled. Correction is not performed. 1: Sequencer operation is enabled.

The timer sequencer control register (TSEQRB) is an 8-bit readable/writable register.

The ECSEQENB bit enables the sequencer operation for correcting the timer B event input cycle. If ECSEQENB is changed from 1 to 0, the sequencer inside is initialized after next external event selection (AGCK) input timing.

TSEQRB can be read or written in 8-bit units only.

TSEQRB is initialized to 00_H after reset.

30.6.2.57 TSEQRB — Timer Sequencer Register

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	FONS	—	—	SEQB[4:0]				
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 30.95 TSEQRB Register Contents

Bit Position	Bit Name	Function
7	FONS	Correction Formula Enable Status 0: Correction is disabled. 1: Correction is enabled.
6, 5	—	These bits are not used. Fix these bits to “0”.
4 to 0	SEQB	Sequencer Status 00000: The sequencer is in State 0. : 11000: The sequencer is in State 24.

The timer sequencer register (TSEQRB) is an 8-bit read-only register that generates AGCK cycle according to the sequencer state.

TSEQRB can be read or written in 8-bit units only.

TSEQRB is initialized to 00_H after reset.

30.6.2.58 TSEQENB0 to TSEQENB2 — Timer Sequencer Enable Registers 0 to 2

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TSEQENB23-16							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSEQENB15-8								TSEQENB7-0							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.96 TSEQENB0 to TSEQENB2 Register Contents

Bit Position	Bit Name	Function
31 to 24	—	These bits are not used. Fix these bits to “0”.
23 to 16	TSEQENB x (x = 23 to 16)	Sequencer Enable 0: Do not transit to the target sequencer state. 1: Transit to the target sequencer state and perform correction.
15 to 8	TSEQENB x (x = 15 to 8)	Sequencer Enable 0: Do not transit to the target sequencer state. 1: Transit to the target sequencer state and perform correction.
7 to 0	TSEQENB x (x = 7 to 0)	Sequencer Enable 0: Do not transit to the target sequencer state. 1: Transit to the target sequencer state and perform correction.

Timer sequencer enable registers 0 to 2 (TSEQENB0 to TSEQENB2) are 8-bit readable/writable registers that control transition of the sequencer to the state upon detection of a compare match between TCNTB1*¹ and OCRB20 to OCRB43.

TSEQENB0 to TSEQENB2 can be read or written in 8-bit units only.

TSEQENB0 to TSEQENB2 are initialized to 00_H after reset.

Note 1. When CHSELB is 00_B, TCNTB1 counter compares, and when CHSELB is 01_B, TCNTB1S1 counter compares, and when CHSELB is 10_B, TCNTB1S2 counter compares.

30.6.2.59 TEPCFB0 to TEPCFB23 — Timer Event Cycle Correction Formula Definition Registers B0 to 23

Value after reset: 0300_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	HALFE Nx	—	COEFFx[2:0]		—	CRFIT1x[2:0]			—	CRFIT2x[2:0]			
Value after reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

CAUTION

While the sequencer is running, it is prohibited to rewrite the timer event cycle correction formula definition registers corresponding to the sequence numbers enabled by the timer sequencer enable bits TSEQNEN[23:0].

Table 30.97 TEPCFB0 to TEPCFB23 Register Contents

Bit Position	Bit Name	Function
15 to 13	—	These bits are not used. Fix these bits to "0".
12	HALFENx	Event Input Cycle Correction Formula 1/2 Multiplication Enable 0: The event input cycle correction formula is not multiplied by 1/2. (Default) 1: The event input cycle correction formula is multiplied by 1/2.
11	—	This bit is not used. Fix this bit to "0".
10 to 8	COEFFx	Correction Formula Coefficient 000: Coefficient $K(x) = 0.25$ 001: Coefficient $K(x) = 0.5$ 010: Coefficient $K(x) = 0.75$ 011: Coefficient $K(x) = 1.0$ (Default) 100: Coefficient $K(x) = 1.25$ 101: Coefficient $K(x) = 1.5$ 110: Coefficient $K(x) = 1.75$ 111: Coefficient $K(x) = 2.0$
7	—	This bit is not used. Fix this bit to "0".
6 to 4	CRFIT1x	Event Input Cycle Correction Formula 2nd Term Cycle 000: Use ICRB0 for the 2nd term of correction formula. 001: Use RECRB1 for the 2nd term of correction formula. 010: Use RECRB2 for the 2nd term of correction formula. 011: Use RECRB3 for the 2nd term of correction formula. 100: Use RECRB4 for the 2nd term of correction formula. 101: Use RECRB5 for the 2nd term of correction formula. 110: Use RECRB6 for the 2nd term of correction formula. 111: Setting prohibited
3	—	This bit is not used. Fix this bit to "0".
2 to 0	CRFIT2x	Event Input Cycle Correction Formula 3rd Term Cycle 000: Use ICRB0 for the 3rd term of correction formula. 001: Use RECRB1 for the 3rd term of correction formula. 010: Use RECRB2 for the 3rd term of correction formula. 011: Use RECRB3 for the 3rd term of correction formula. 100: Use RECRB4 for the 3rd term of correction formula. 101: Use RECRB5 for the 3rd term of correction formula. 110: Use RECRB6 for the 3rd term of correction formula. 111: Setting prohibited

Timer event cycle correction formula definition registers B0 to B23 (TEPCFB0 to TEPCFB23) are 16-bit readable/writable registers.

TEPCFB is initialized to 0300_H after reset.

30.6.2.60 TCNTB3 — Correction Event Counter B3

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTB3[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTB3[15:12]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.98 TCNTB3 Register Contents

Bit Position	Bit Name	Function
31 to 12	CNTB3	Correction Event Counter B3 20-bit event counter value
11 to 0	—	These bits are not used. Fix these bits to “0”.

Correction event counter B3 (TCNTB3) is a 20-bit up-counter that is mapped a 32-bit readable/writable register.

When ex external event selection (AGCK) is input, TCNTB3 transfers the counter value to correction counter B4 (TCNTB4) and then starts up counting using the value of pulse interval multiplier register 1 (PIMR1). The value of pulse interval multiplier register 2 (PIMR2) is not used for up-counting of TCNTB3.

TCNTB3 does not perform counting if the TBE bit of ATU master enable register (ATUENR) is not set to “1”. The counter value is not cleared if TBE is cleared to “0”.

According to the setting for the CLRB3 and CLRB3SEL bits in the timer control register B (TCRB), the TCNTB3 value can be cleared to 0000 0000_H at the next PCLK timing after a compare match*¹ between TCNTB1 and OCRB10 or between TCNTB6 and OCRB6 occurs.

TCNTB3 can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

TCNTB3 is initialized to 0000 0000_H after reset.

Note 1. CHSELB uses by compare match of TCNTB1 and OCRB11 at 00_B, the compare match by which CHSELB is TCNTB1S1 and OCRB10S1 at 01_B, the compare match of TCNTB1S2 and OCRB10S2 at 10_B.

30.6.2.61 OCRB8 — Output Compare Register B8

Value after reset: FFFF F000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OCB8[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCB8[15:12]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.99 OCRB8 Register Contents

Bit Position	Bit Name	Function
31 to 12	OCB8	Output Compare B8 These bits store 20-bit value to be compared with TCNTB3.
11 to 0	—	These bits are not used. Fix these bits to "0".

Output Compare Register B8 (OCRB8) is a 32-bit readable/writable register in which output compare B8 (OCB8) bits are mapped to the upper 20 bits.

The OCRB8 value is compared with the TCNTB3 value. When TCNTB3 clear (due to a compare match between OCRB8 and TCNTB3) is enabled by bit 4 and bit 3 in the timer control register B (TCRB), the TCNTB3 value is cleared to 0 at the first PCLK timing after a compare match between the OCRB8 value and the TCNTB3 value. OCRB8 has no compare match status flag or interrupt

OCRB8 can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

OCRB8 is initialized to FFFF F000_H after reset.

30.6.2.62 TEPCFENB0 to TEPCFENB2 — Timer Event Cycle Correction Enable Registers 0 to 2

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	FON23-16							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FON15-8								FON7-0							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.100 TEPCFENB0 to TEPCFENB2 TEPCFENB0 to TEPCFENB2

Bit Position	Bit Name	Function
31 to 24	—	These bits are not used. Fix these bits to “0”.
23 to 16	FONx	Event Input Cycle Correction Formula x Correction Enable 0: The correction formula is disabled in sequencer state x. (AGCK1 multiplication cycle is calculated using the ICRB0 value only.) 1: The correction formula is enabled in sequencer state x.
15 to 8	FONx	Event Input Cycle Correction Formula x Correction Enable 0: The correction formula is disabled in sequencer state x. (AGCK1 multiplication cycle is calculated using the ICRB0 value only.) 1: The correction formula is enabled in sequencer state x.
7 to 0	FONx	Event Input Cycle Correction Formula x Correction Enable 0: The correction formula is disabled in sequencer state x. (AGCK1 multiplication cycle is calculated using the ICRB0 value only.) 1: The correction formula is enabled in sequencer state x.

Timer event cycle correction enable registers 0 to 2 (TEPCFENB0 to TEPCFENB2) are 8-bit readable/writable registers. TEPCFENB0 to TEPCFENB2 enables and disables the correction formula that is set by the TEPCFBx and TEPCOEFFBx registers in sequencer state x indicated by the TSEQRB register (x = 0 to 23).

TEPCFENB0 to TEPCFENB2 can be read or written in 8-bit units only.

TEPCFENB0 to TEPCFENB2 are initialized to 00_H after reset.

30.6.2.63 TEPCVALRB0 — Timer Event Input Cycle Correction Result Retention Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEPCVALRB0[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEPCVALRB0[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.101 TEPCVALRB0 Register Contents

Bit Position	Bit Name	Function
31-0	TEPCVALB0	Timer Event Input Cycle Correction Result Retention 0 These bits store 32-bit timer event input cycle correction result.

The timer event input cycle correction result retention register (TEPCVALRB0) is a 32-bit read-only register. TEPCVALRB0 captures the ICRB0 register value*¹ that is corrected according to the setting in timer event cycle correction formula definition registers B0 to B23 (TEPCFB0 to TEPCFB23) and timer event cycle correction formula enable register B0 to B2 (TEPCFENB0 to TEPCFENB2).

TEPCVALRB0 can be read in 32-bit units only.

TEPCVALRB0 is initialized to 0000 0000_H after reset.

Note 1. CHSELB revises ICRB0 register value at 00_B. CHSELB revises ICRB0S1 register value at 01_B also CHSELB revises ICRB0S2 register value at 10_B.

30.6.2.64 TEPCRECRB1 — Timer Event Input Cycle Correction Result Log Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEPCRECRB1[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEPCRECRB1[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.102 TEPCRECRB1 Register Contents

Bit Position	Bit Name	Function
31 to 0	TEPCRECRB1	Timer Event Input Cycle Correction Log 1 These bits store 32-bit event captured value.

The timer event input cycle correction result log register (TEPCRECRB1) is a 32-bit read-only register that captures the TEPCVALRB0 register value when a timer B external event selection (AGCK) is input.

TEPCRECRB1 can be read in 32-bit units only.

TEPCRECRB1 is initialized to 0000 0000_H after reset.

30.6.2.65 TCNTB4 — Multiplied-and-Corrected Clock Counter B4

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTB4[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTB4[15:12]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.103 TCNTB4 Register Contents

Bit Position	Bit Name	Function
31 to 12	CNTB4	Multiplied-and-Corrected Clock Counter B4 These bits store 20-bit frequency-multiplied clock count value.
11 to 0	—	These bits are not used. Fix these bits to “0”.

The multiplied-and-corrected clock counter B4 (TCNTB4) is a 20-bit up-counter mapped in the 32-bit readable/writable register.

The value in TCNTB3 is loaded to this counter when the externally input event selection (AGCK) is detected.

This up-counter is driven by the frequency-multiplied clock (AGCK1) output by reload counter B2 (TCNTB2) and is cleared to 00000_H when the externally input event selection is detected while CNTB3 = 00000_H.

When the values in this counter and TCNTB3 match with the CCS bit in timer I/O control register B (TIORB) set to 1, counting is stopped*¹. When the CCS bit is set to 0, counting is continued.

When the TBE bit in ATU master enable register (ATUENR) is set to 1, counting is not performed. Even if the TBE bit is cleared to 0 and counting is stopped, the counter value is not cleared.

TCNTB4 can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

TCNTB4 is initialized to 0000 0000_H after reset.

Note 1. When CU4SEL is 0, and then count up to the value of TCNTB3 matches TCNTB4. When CU4SEL is 1, and then count up to the value of “TCNTB3 - 1” matches TCNTB4.

30.6.2.66 TCNTB5 — Multiplied-and-Corrected Clock Generating Counter B5

Value after reset: 0000 1000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTB5[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTB5[15:12]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.104 TCNTB5 Register Contents

Bit Position	Bit Name	Function
31 to 12	CNTB5	Multiplied-and-Corrected Clock Generation Count B5 These bit store 20-bit multiplied-and-corrected clock count value.
11 to 0	—	These bits are not used. Fix these bits to “0”.

The multiplied-and-corrected clock generating counter B5 (TCNTB5) is a 20-bit up-counter mapped in the 32-bit readable/writable register.

This counter is enabled by the TBE bit in ATU master enable register (ATUENR) and the count control B5 bit (CTCNTB5). Incrementation of this counter is driven by the PCLK as long as the counter value is less than the value in multiplied-and-corrected clock counter B4 (TCNTB4).

Incrementation is stopped when the counter value matches the value in correcting counter clearing register (TCCLRB).

The value in this counter is corrected and cleared when the externally input event selection (AGCK) is detected while CNTB3 = 0000_H. The value for the clearing depends on the counter value.

- When TCNTB5 = TCCLRB

This counter is cleared. The corrected counter clear flag register B (TCCLFRB) retains 0.

- When TCNTB5 ≠ TCCLRB

The TCCLFB bit of correction counter clear flag register (TCCLFRB) is set to “1”, and TCNTB5 continues counting automatically. When the counter value reaches the value in TCCLRB, TCNTB5 is cleared and the TCCLFB bit is cleared to “0”.

The clear value of TCNTB5 is different depending on setting of the CU4SEL bit of “Multiplied-and-Corrected Clock Generating Counter B4 Control Register (TCNT4CRB)”.

When the CU4SEL bit is set to “0”, the value is cleared to 0000 1000_H.

When the CU4SEL bit is set to “1”, the value is cleared to 0000 0000_H.

Every incrementation of this counter, a single pulse of the multiplied-and-corrected clock (AGCKM) is output. The clock can be output on clock-bus line 5 by setting the CB5SEL bit in clock bus control register (CBCNT) to 1. The output of the clock is temporarily stopped by altering the setting in bit CTCNTB5 in the TIORB during counting.

TCNTB5 can be read or written in 32-bit units only. Do not write or read 16-bit data for this register.

TCNTB5 is initialized to 0000 1000_H after reset.

Initial setting is required for TCNTB5 (make the contents of TCNTB5 and TCCLR5 the same) before starting timer B counter operation. For details, see **Section 30.6.3, Details of Operation**.

30.6.2.67 TCCLFRB — Correcting Counter Clear Flag Register B

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCCLFB
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 30.105 TCCLFRB Register Contents

Bit Position	Bit Name	Function
7 to 1	—	These bits are not used. Fix these bits to “0”.
0	TCCLFB	Correcting Counter Clear Flag B 0: TCNTB5 is not automatically counting up. (Default) 1: TCNTB5 is automatically counting up until it matches the TCCLRB value.

The correcting counter clear flag register B (TCCLFRB) is an 8-bit read-only register that indicates whether the function of correcting TCNTB5 (the automatic up-counting function that continues counting until TCNTB5 = TCCLRB) is enabled or disabled.

TCCLFRB can be read in 8-bit units only.

TCCLFRB is initialized to 00_H after reset.

Correcting Counter Clear Flag B (TCCLFB)

This status flag shows that the function of correcting TCNTB5 is enabled and the automatic up-counting function continues until TCNTB5 = TCCLRB. When 1 is read from this flag, it shows that TCNTB5 is automatically up-counting until TCNTB5 = TCCLRB. This flag cannot be set to 1 or 0 by the software.

- Setting (to 1) conditions
 - When an external event selection is input with TCNTB3 = 0000_H and TCNTB5 ≠ TCCLRB while TCNTB5 is automatically up-counting until it reaches the TCCLRB value
- Clearing (to 0) condition
 - When TCNTB5 value has reached the TCCLRB value after counting with the TCCLFB bit set to 1, the TCNTB5 value has matched the TCCLRB value, and TCNTB5 has been cleared*¹

Note 1. CU4SEL clears TCNTB5 in 0000 1000_H at 0. CU4SEL clears TCNTB5 in 0000 0000_H at 1.

30.6.2.68 TCCLRB — Correcting Counter Clearing Register B

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CCLRB[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCLRB[15:12]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.106 TCCLRB Register Contents

Bit Position	Bit Name	Function
31 to 12	CCLRB	Correcting Counter Clear B These bits store 20-bit correcting counter clear value.
11 to 0	—	These bits are not used. Fix these bits to “0”.

The correcting counter clearing register B (TCCLRB) is a 32-bit readable/writable register mapped to upper 20 bits in 32 bits.

TCCLRB is constantly compared with TCNTB5. When they match, TCNTB5 is stopped and a counter clearing trigger is output to timer D. TCNT1Dx and TCNT2Dx in timer D are separately cleared by setting the corresponding counter clearing enable bit in timer control register Dx (TCRDx).

TCCLRB can be read or written in 32-bit units only. Do not write or read 8-bit or 16-bit data for this register.

TCCLRB is initialized to 0000 0000_H after reset.

30.6.3 Details of Operation

30.6.3.1 Edge Interval Measuring Function and Edge Input Stopping Function

For timer B, input-capture and compare-match operations are unconditionally performed with 32-bit input capture register B0 (ICRB0) and 32-bit output compare register B0 (OCRB0), respectively.

These registers are connected to input edge-interval measuring counter B0 (TCNTB0).

Operation of timer B is started by setting the TBE bit in the ATU master enable register (ATUENR).

ICRB0 captures the TCNTB0 value when an external-event 1_0 is input via timer A. At the same time, TCNTB0 is set to 0000 0001_H. An interrupt request can be output to the CPU at this capture timing*¹. This enables measurement of the time of edge-to-edge interval.

The value captured in ICRB0*¹ is transferred to the frequency-multiplied clock generating block to be used as an input value to the reload counter B2 (TCNTB2) and the reload register B (RLDB).

Furthermore, up to six captured values (ICRB0*¹) can be retained in the record registers B1 to B6 (RECRB1 to RECRB6) at the external input event selection (AGCK) timing. Values retained in ICRB0*¹ and RECRB1 to RECRB6 are backed up in the record backup registers B0 to B6 (RBURB0 to RBURB6) at the timing of B12 (OCRB12) event compare match*² between TCNTB1 and the output compare register B12 (OCRB12).

When TCNTB0 reaches the value set in OCRB0, an interrupt request due to this compare match can be output*³. This interrupt indicates that active edge input has stopped for at least time equivalent to the setting in OCRB0.

- Note 1.** CHSELB [1:0] outputs interrupt by capture timing of ICRB0 at 00_B, and a flag (IMFB0) is set. CHSELB [1:0] outputs interrupt by capture timing of ICRB0S1 at 01_B, and a flag (IMFB0) is set. CHSELB [1:0] outputs interrupt by capture timing of ICRB0S2 at 10_B, and a flag (IMFB0) is set.
- Note 2.** CHSELB [1:0] is a compare match of TCNTB1 and OCRB12 at 00_B, and it's backup in RBURB0-6. CHSELB [1:0] is a compare match of TCNTB1 and OCRB12S1 at 01_B, and it's backup in RBURB0-6. CHSELB [1:0] is a compare match of TCNTB1 and OCRB12S2 at 10_B, and it's backup in RBURB0-6.
- Note 3.** CHSELB [1:0] is a compare match of TCNTB0 and OCRB0 and outputs interrupt at 00_B, and a flag (CMFB0) is set. CHSELB [1:0] is a compare match of TCNTB0S1 and OCRB0S1 and outputs interrupt at 01_B, and a flag (CMFB0) is set. CHSELB [1:0] is a compare match of TCNTB0S2 and OCRB0S2 and outputs interrupt at 10_B, and a flag (CMFB0) is set.

Figure 30.18 shows input-capture and compare-match operation of TCNTB0 at CHSELB[1:0] = 00_B.

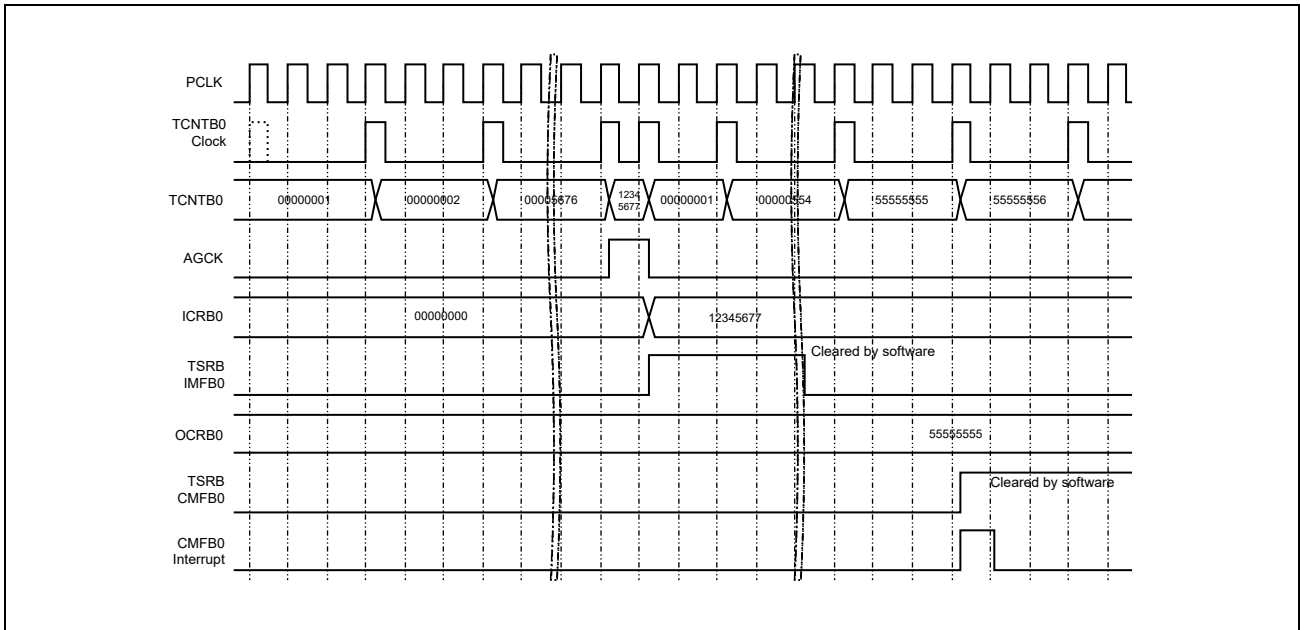


Figure 30.18 Input-Capture and Compare-Match Operation of TCNTB0 at CHSELB[1:0] = 00_B

Event counter B1 (TCNTB1) counts the input of signals indicating external events 1_0. When a predetermined value is set in output compare register B1 (OCRB1) and TCNTB1 reaches that value, a compare-match occurs. When a value is set in output compare register B10 (OCRB10) by the setting of the TCNTB1 clear setting bit (CLRB1) in the timer control register B (TCRB), TCNTB1 can be cleared at the first event input timing after an event compare match between TCNTB1 and OCRB10. Furthermore, by the setting of the TCNTB3 clear setting bit (CLRB3SEL) in the timer control register B (TCRB), TCNTB3 can be cleared at a compare match between TCNTB1 and OCRB10*¹. Input capture register B2 (ICRB2) can capture input capture register B1 (ICRB1) at the timing of a compare match between TCNTB1 and OCRB1. This compare match*² makes it possible to output an interrupt request to the CPU, allowing detection of edge input stop of external event. In addition, event counter B1 (TCNTB1) has a function for compare match with output compare register B11 (OCRB11). When the pulse interval multiplier select bit (PIMRSELR) is set to 1, TCNTB1 can operate by using PIMR2 for down-count values to be loaded to RLDB and for down-count values of TCNTB2 at a compare match between TCNTB1 and OCRB11*³. When the TCNTB1 value matches the OCRB11 value, an interrupt request due to this compare match*³ can be output to the CPU.

The external event signal 1_0 drives the capturing of TCNTB0 values in ICRB1. Moreover, latching of ICRB1 in ICRB2 on matches between TCNTB1 and OCRB1 can also be selected. This enables the measurement of multiple edge-to-edge intervals.

Registers ICRB30 to ICRB37 capture the TCNTB1*⁴ value by using the external event input 1B to 1I as a trigger. ICRB30 corresponds to external event input 1B and ICRB31 to ICRB37 correspond to respective external event input 1C to 1I.

- Note 1.** When CHSELB [1:0] is 00_B, TCNTB3 can be cleared by a compare match of TCNTB1 and OCRB10. When CHSELB [1:0] is 01_B, TCNTB3 can be cleared by a compare match of TCNTB1S1 and OCRB10S1. When CHSELB [1:0] is 10_B, TCNTB3 can be cleared by a compare match of TCNTB1S2 and OCRB10S2.
- Note 2.** When CHSELB [1:0] is 00_B, interrupt request can be output by a comparing match of TCNTB1 and OCRB1. When CHSELB [1:0] is 01_B, interrupt request can be output by a comparing match of TCNTB1S1 and OCRB1S1. When CHSELB [1:0] is 10_B, interrupt request can be output by a comparing match of TCNTB1S2 and OCRB1S2.
- Note 3.** When CHSELB [1:0] is 00_B, PIMR2 can be used by a compare match of TCNTB1 and OCRB11, and interrupt request can be output. When CHSELB [1:0] is 01_B, PIMR2 can be used by a compare match of TCNTB1S1 and OCRB11S1, and interrupt request can be output. When CHSELB [1:0] is 10_B, PIMR2 can be used by a compare match of TCNTB1S2 and OCRB11S2, and interrupt request can be output.
- Note 4.** When CHSELB [1:0] is 00_B, the value of TCNTB1 can be captured in ICRB30-37. When CHSELB [1:0] is 01_B, the value of TCNTB1S1 can be captured in ICRB30-37. When CHSELB [1:0] is 10_B, the value of TCNTB1S2 can be captured in ICRB30-37.

Figure 30.19 shows compare-match operation of TCNTB1 and capture operation of ICRB1 and ICRB2 at CHSELB[1:0] = 00_B, and **Figure 30.20** shows capture operation of TCNTB1 and ICRB3x.

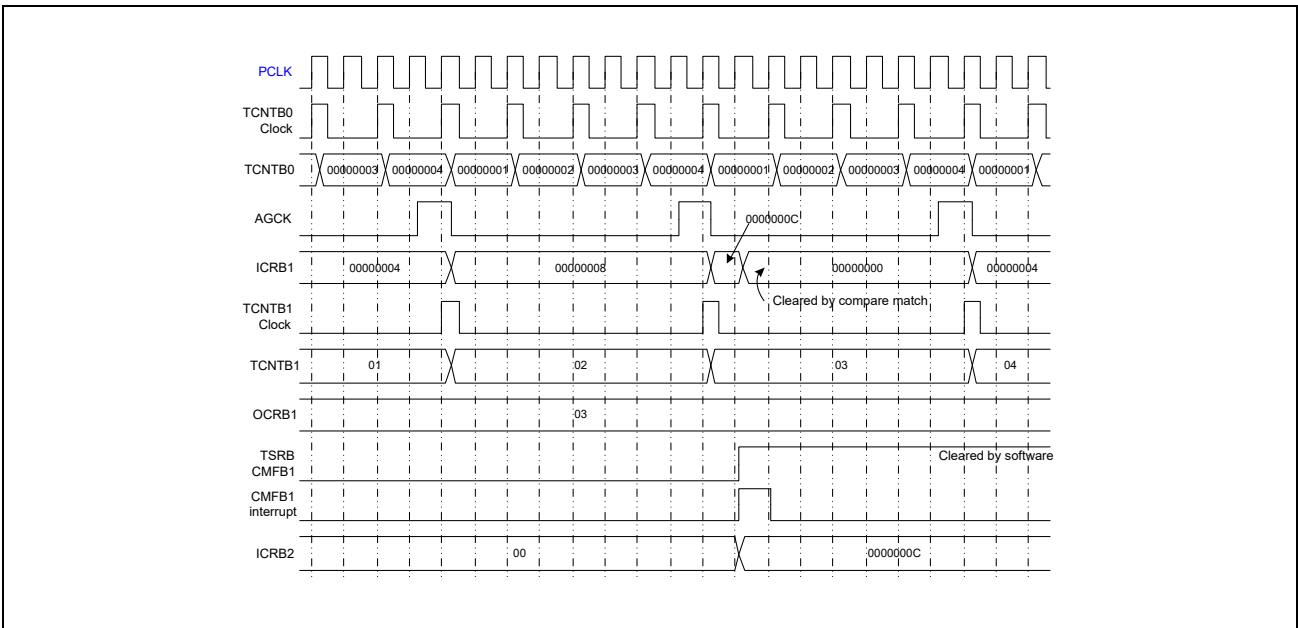


Figure 30.19 Compare-Match Operation of TCNTB1 and Capture Operation of ICRB1 and ICRB2 at CHSELB[1:0] = 00_B

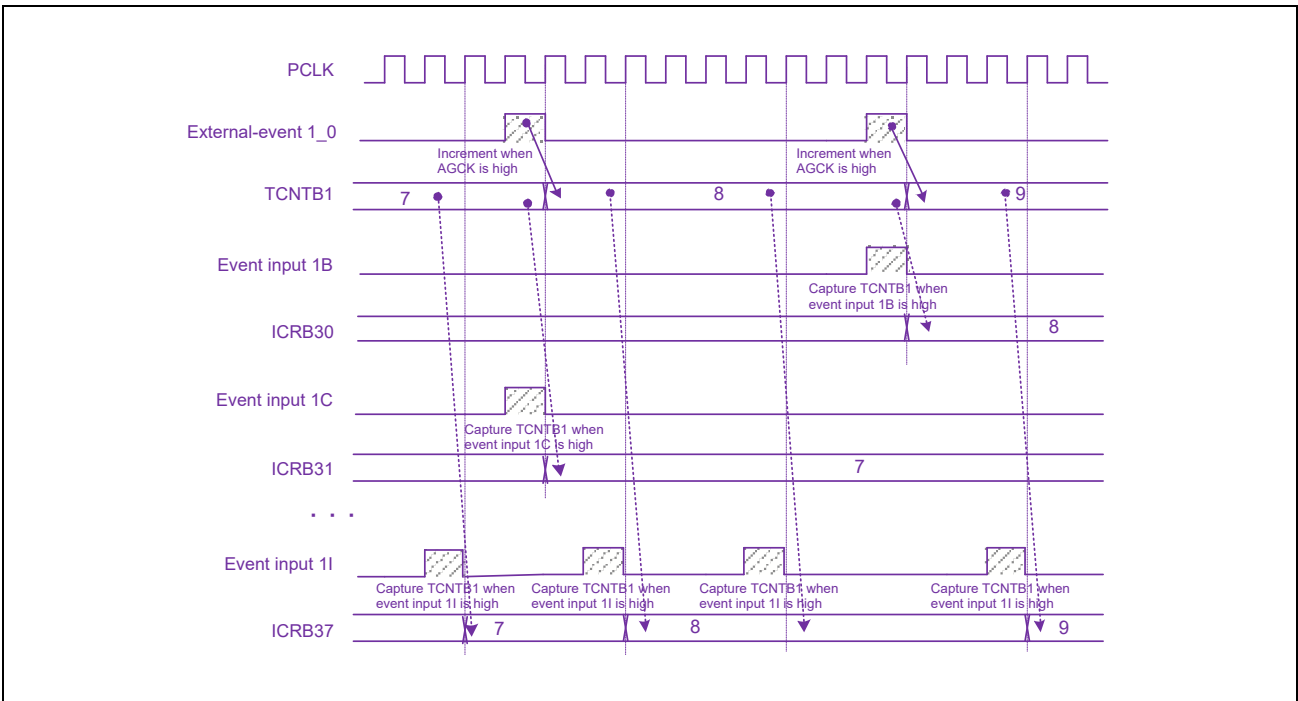


Figure 30.20 Count Operation of TCNTB1 and Capture Operation of ICRB3x at CHSELB[1:0] = 00_B

Upon detection of a compare match between event counter B1 (TCNTB1)*¹ and output compare register (OCRB20 to OCRB43), the state of the sequencer that control correction for edge interval measurement values is determined. The condition for transiting to the corresponding sequencer state upon compare match between TCNTB1*¹ and OCRB20 to OCRB43 is controlled by the SEQEN23 to SEQEN0 bits. IF the same value is set for OCRB20 to OCRB43, the transition caused by a compare match with upper OCRB (OCRB43 < OCRB42 < ... < OCRB21 < OCRB20) takes priority. According to the TEPCFB20 to TEPCFB43 set values for the sequencer state after transition, the correction value is calculated and the result is saved in timer event input cycle correction result retention register (TEPCVALRB0). The TEPCVALRB0 value is saved into timer event input cycle correction result log register (TEPCRECRB1) at the external event selection (AGCK) detection timing. The value of ICRB0*² or TEPCVALRB0 is supplied to the frequency-multiplied clock generator, according to the EVSEQENB bit setting.

Note 1. When CHSELB [1:0] is 00_B, a comparing match with TCNTB1 is put into effect. When CHSELB [1:0] is 01_B, a comparing match with TCNTB1S1 is put into effect. When CHSELB [1:0] is 10_B, a comparing match with TCNTB1S2 is put into effect.

Note 2. When CHSELB [1:0] is 00_B, the value of ICRB0 is communicated to frequency-multiplying clock generator. When CHSELB [1:0] is 01_B, the value of ICRB0S1 is communicated to frequency-multiplying clock generator. When CHSELB [1:0] is 10_B, the value of ICRB0S2 is communicated to frequency-multiplying clock generator.

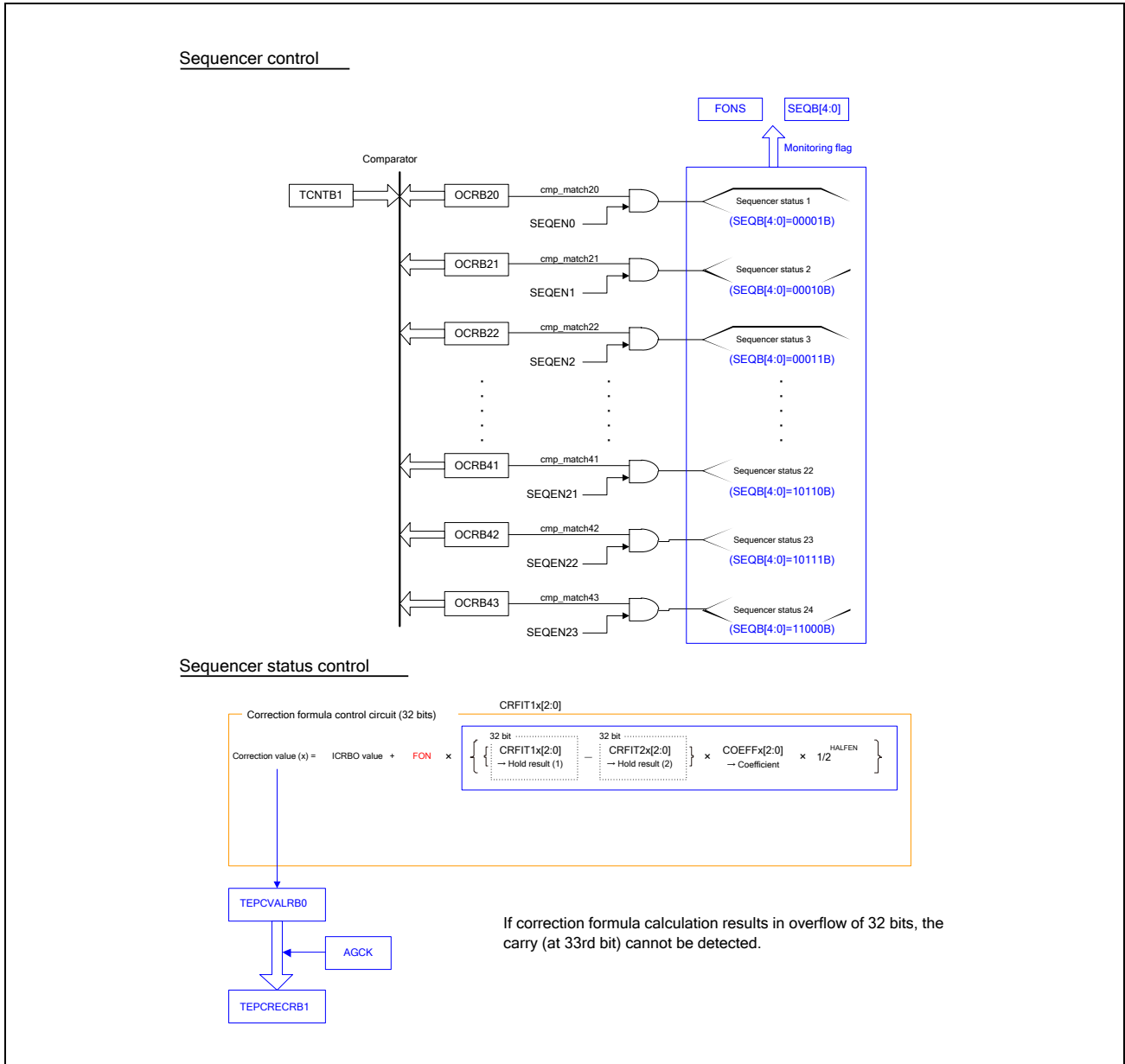


Figure 30.21 Operation upon Compare Match between TCNTB1 and OCRB20-43 and Sequencer State Transition at CHSELB[1:0] = 00_B

NOTE

TCNTB1 and ICRB0 are used by sequencer control of the figure above because CHSELB [1:0] is 00_B but TCNTB1S1 and ICRB0S1 use the time when CHSELB [1:0] is 01_B and TCNTB2S2 and ICRB2S2 use the time when CHSELB [1:0] is 10_B.

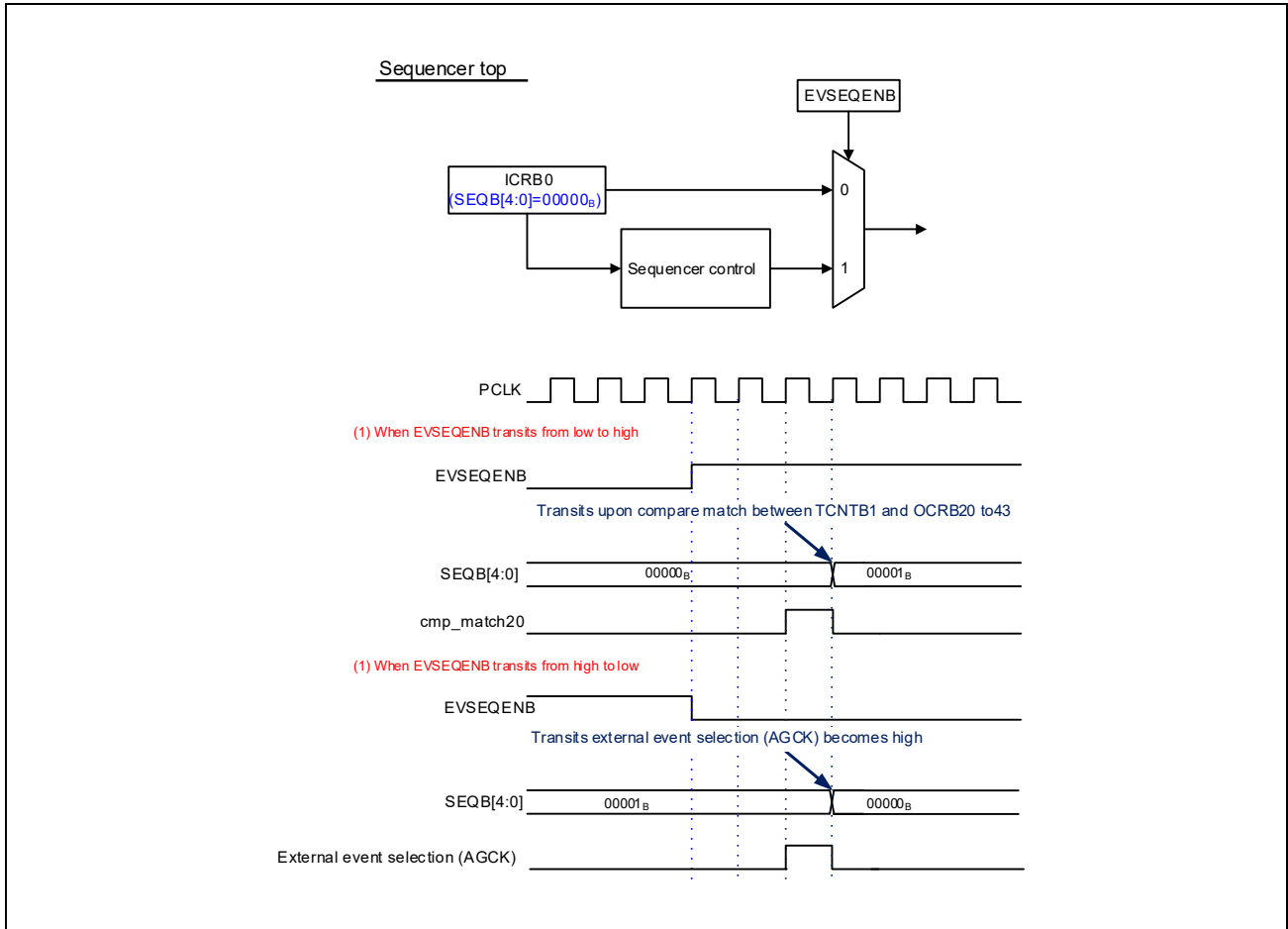


Figure 30.22 Switching between ICRB0 and TEPVALRB0 by EVSEQENB at CHSELB[1:0] = 00_B

NOTE

ICRB0 is used by control of the figure above because CHSELB [1:0] is 00_B but ICRB0S1 is used by the time when CHSELB [1:0] is 01_B and ICRBS2 is used by the time when CHSELB [1:0] is 10_B.

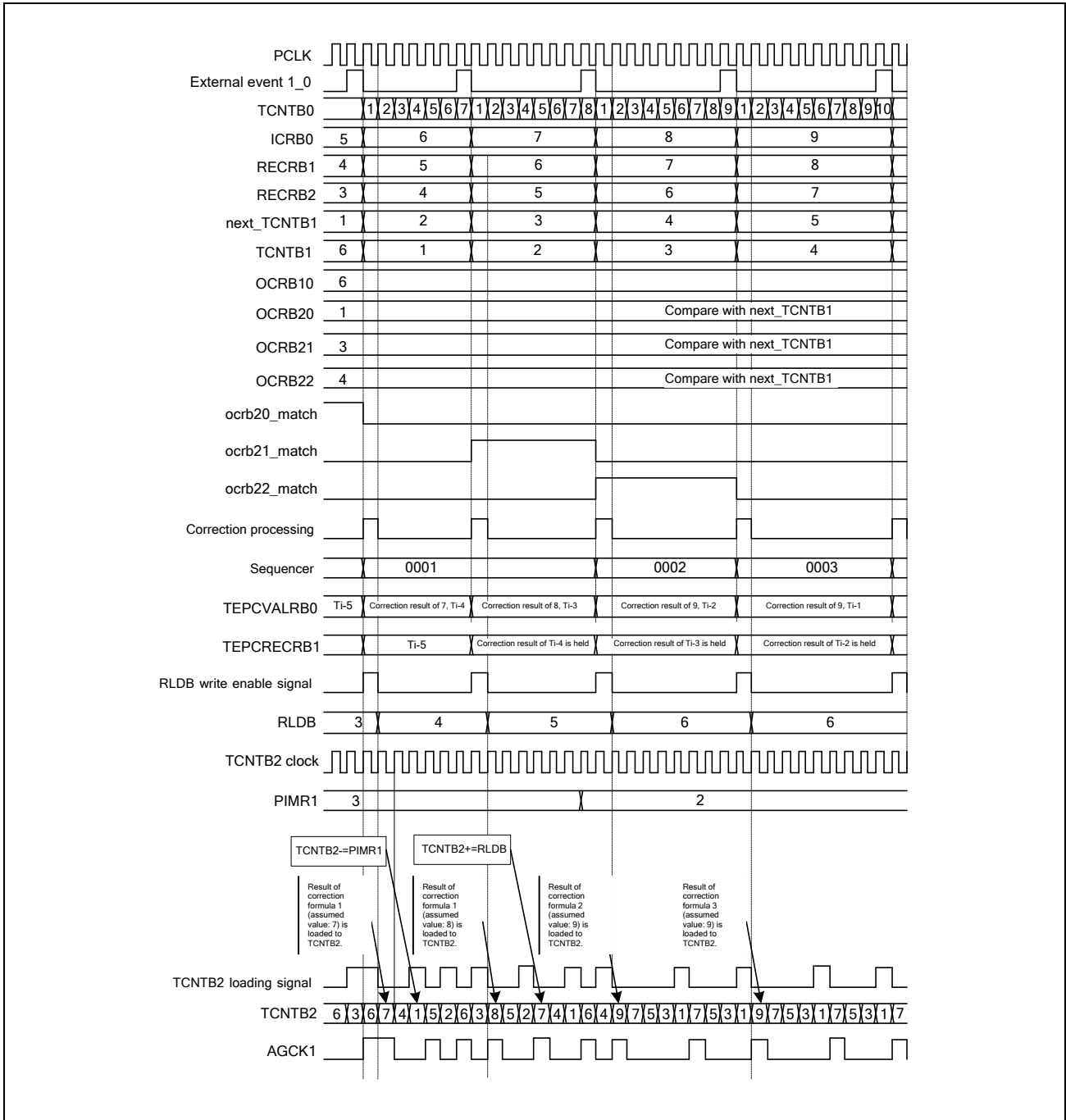


Figure 30.23 Operation of Compare Match between TCNTB1 and OCRB20-43 and Sequencer Control at CHSELB[1:0] = 00_B

30.6.3.2 Frequency-Multiplied Clock Generator

The frequency-multiplied clock generator generates a clock signal (AGCK1) for use within timer B. The cycle of this clock signal is obtained by dividing the intervals between selected transitions of the external-event selection (AGCK) input by the value in the pulse interval multiplier register (PIMR1 or PIMR2).

When the LDEN bit in the TIORB is 0, On the selected transition of the external-event selection (AGCK) input signal, the lower 24 bits of the value captured in ICRB0*¹ of the edge-interval measuring block are transferred to 24-bit reloadable counter B2 (TCNTB2). At the same time, the value transferred to TCNTB2 minus the value in PIMR (PIM1 or PIM2) is transferred to 24-bit reload register B (RLDB). When the LDEN bit in the TIORB is 1, the values of TCNTB2 and RLDB are not updated on ICRB0*¹ input capture.

When the LDSEL bit in timer I/O control register B (TIORB) is set to 1, the value in load register B (LDB) instead of that in ICRB0*¹ can be transferred to TCNTB2 and RLDB. When the pulse interval multiplier select bit (PIMRSELR) is set to 1, the above operation is performed using the PIMR2 value instead of PIMR1 value while TCNTB1 matches OCRB11*².

24-bit reloadable counter B2 (TCNTB2) is driven to count down by the clock selected by the CKSELB bits in timer control register B (TCRB). Each decrementation is by the value set in PIMR (TCNTB2- = PIM). When the value in the down counter is less than or equal to that of the PIM bits, RLDB is automatically read out into TCNTB2 and added to the count value (TCNTB2 + the RLDB value), and which again starts to count down. A single pulse of the multiplied clock signal (AGCK1) is output in synchronization with the reloading of TCNTB2. The pulse width is equal to the cycle of the PCLK.

Compare match operation by frequency-multiplied clock (AGCK1) can be performed by using 20-bit output compare register B6 (OCRB6) and 20-bit frequency-multiplied clock counter B6 (TCNTB6).

Incrementation of TCNTB6 on the assertion of AGCK1 is unconditional. The values in TCNTB6 and output compare register B6 (OCRB6) are tested for matches, and an interrupt request will be generated to the CPU when the values match. The IREGB6 bits in TICRB can be set so that the interrupt is generated on the match, on the first AGCK pulse (external event) after the match, or on the second AGCK pulse (external event) after the match.

When an external event selection input (AGCK) is detected, ICRB6 stores the TCNTB6 value at the next clock (PCLK) timing. The frequency-multiplied clock counter B6M (TCNTB6M) performs count-up operation with a count value of multiplication that is set in multiplication setting register B6 (RARB6) for the frequency-multiplied clock (AGCK1). Each time an external event selection (AGCK) is input, the TCNTB6M value is compared with the ICRB6 value. When the ICRB6 value is larger than the TCNTB6M value as a result of comparison, an interrupt request can be output.

Since AGCK1 is generated with reference to the previous edge-to-edge interval, if two consecutive edge intervals differ significantly, the clock will not be generated correctly. To correct this error, AGCK1 is corrected by the frequency-multiplied clock signal corrector, whose description is provided in the next section, and is to be converted into AGCKM.

Figure 30.24 and **Figure 30.25** show counting operations with reloading and output of the frequency-multiplied clock and **Figure 30.26** and **Figure 30.27** show the generation of interrupt requests on matches between TCNTB6 and CMFB6, enabled or disabled by the setting in IREG. **Figure 30.28** shows the Switching Timing of PIMR1 and PIMR2. **Figure 30.29** shows comparison of TCNTB6M and ICRB6 operations and interrupt request output.

Note 1. When CHSELB [1:0] is 00_B, ICRB0 is used. When CHSELB [1:0] is 01_B, ICRB0S1 is used. When CHSELB [1:0] is 10_B, ICRB0S2 is used.

Note 2. When CHSELB [1:0] is 00_B, PIMR2 is used by a compare match of TCNTB1 and OCRB11. When CHSELB [1:0] is 01_B, PIMR2 is used by a compare match of TCNTB1S1 and OCRB11S1. When CHSELB [1:0] is 10_B, PIMR2 is used by a compare match of TCNTB1S2 and OCRB11S2.

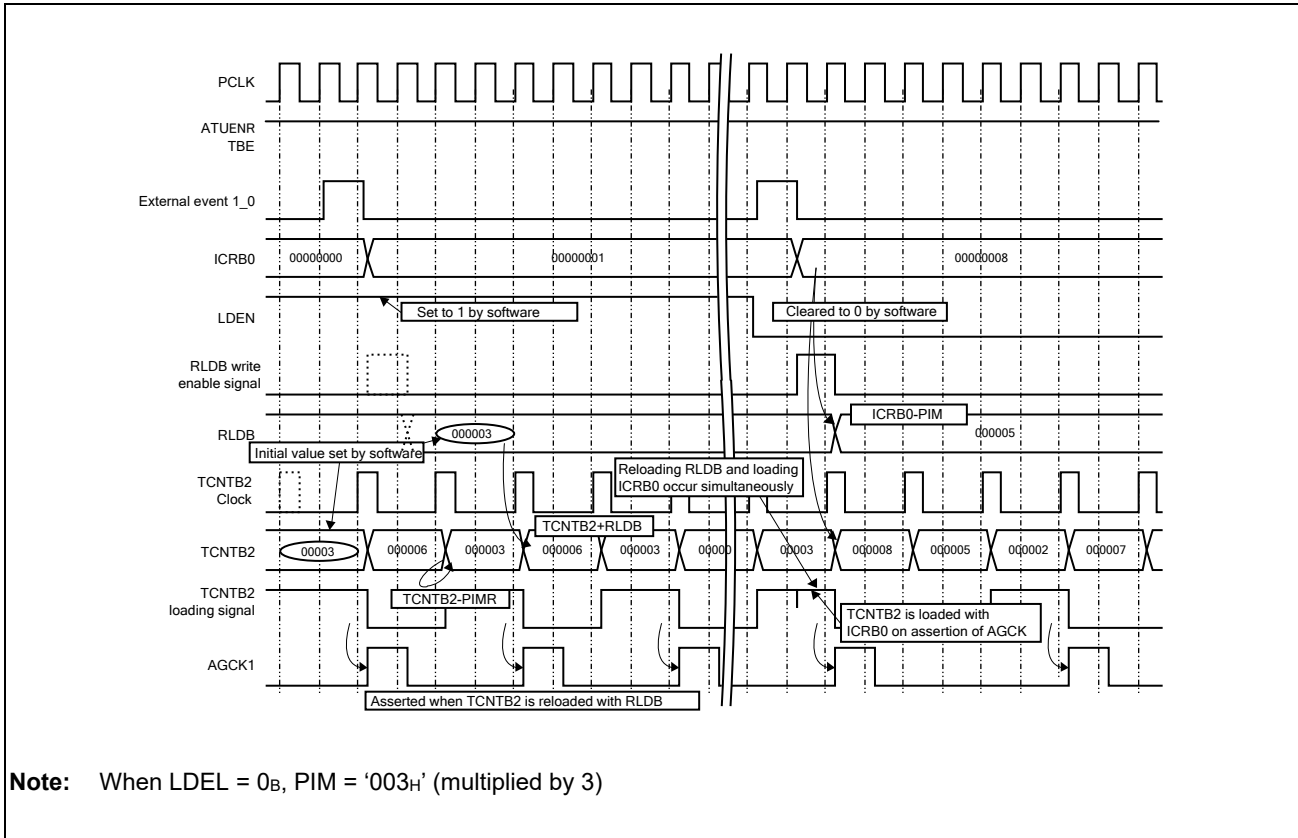


Figure 30.24 Counting Operations with Reloading and Output of Frequency - Multiplied Clock at CHSELB[1:0] = 00_B (1)

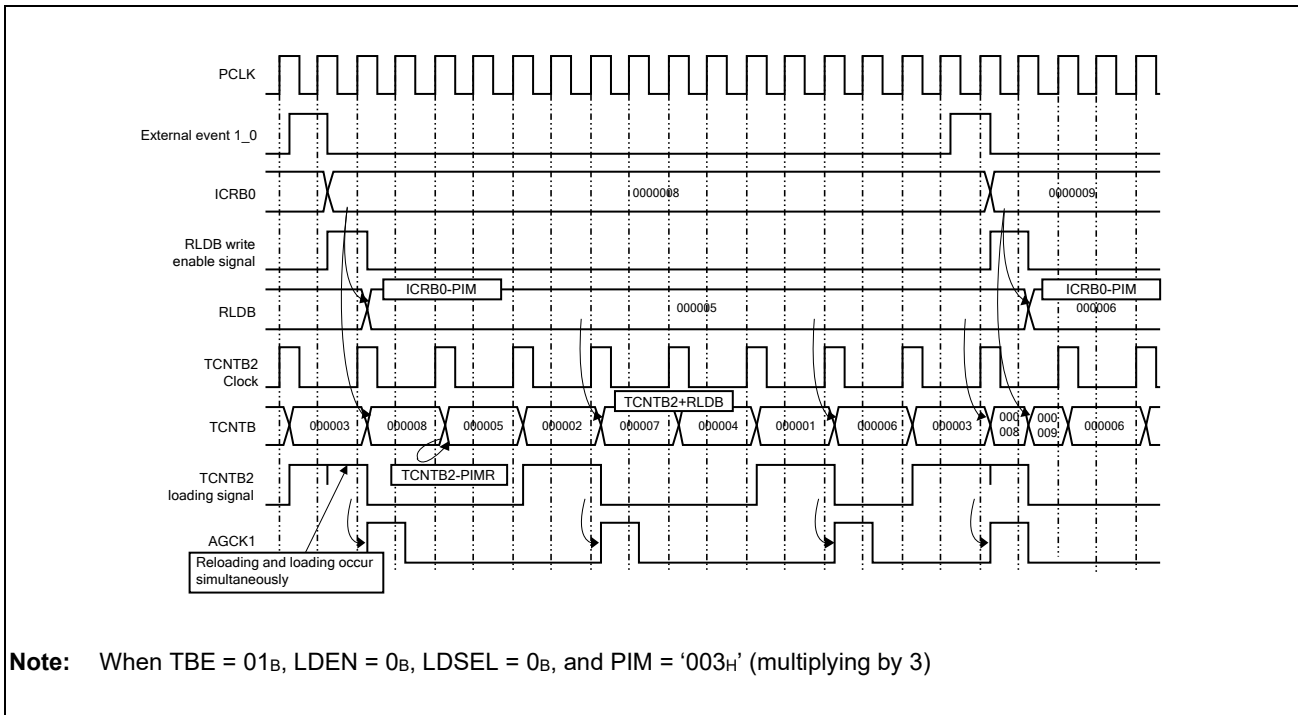


Figure 30.25 Reload Count Operation and Multiplied-Clock Output at CHSELB[1:0] = 00_B (2)

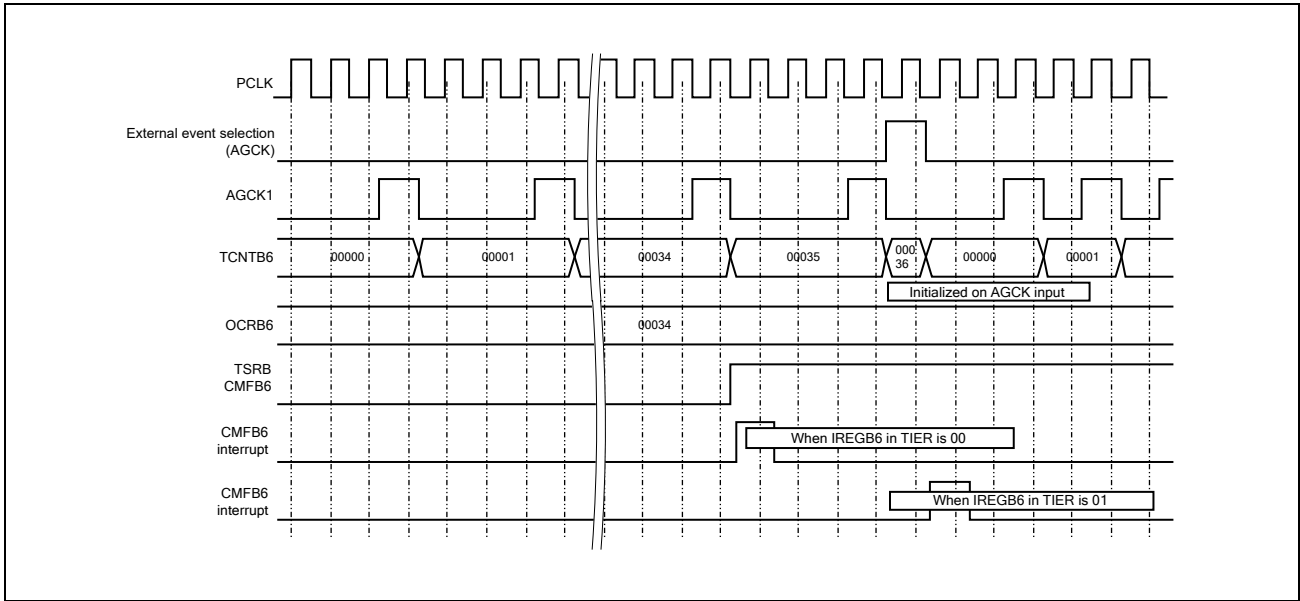


Figure 30.26 Compare-Match Operation of TCNTB6 and Output of CMFB6 Interrupt (IREGB6 = 00, 01)

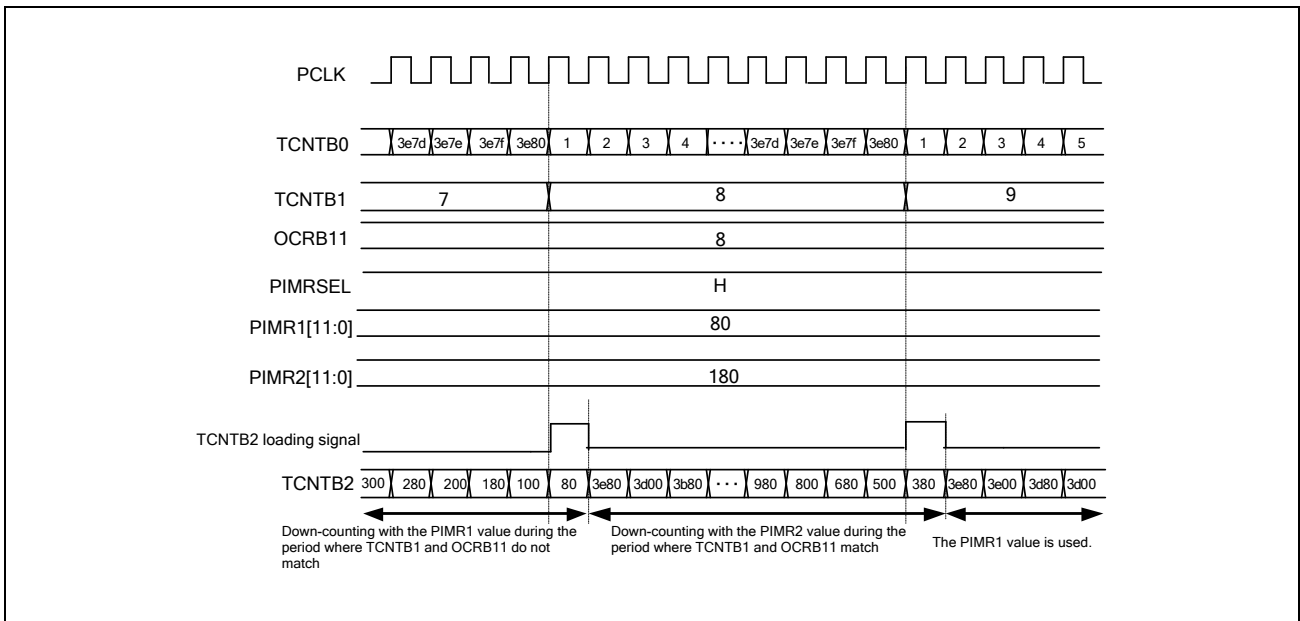


Figure 30.27 Switch Timing between PIMR1 and PIMR2 at CHSELB[1:0] = 00B

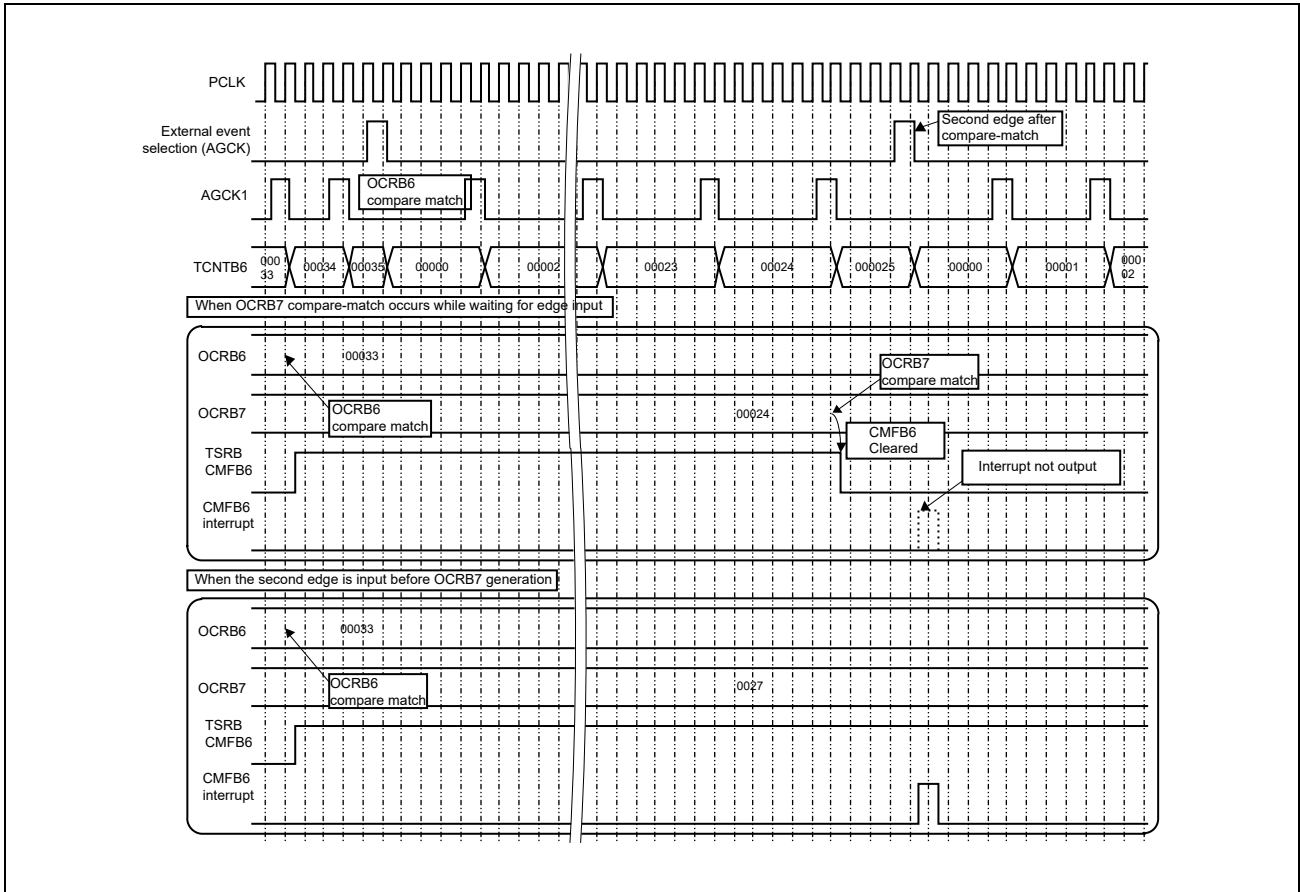


Figure 30.28 CMTB6 Interrupt Output when IREGB6 = 10

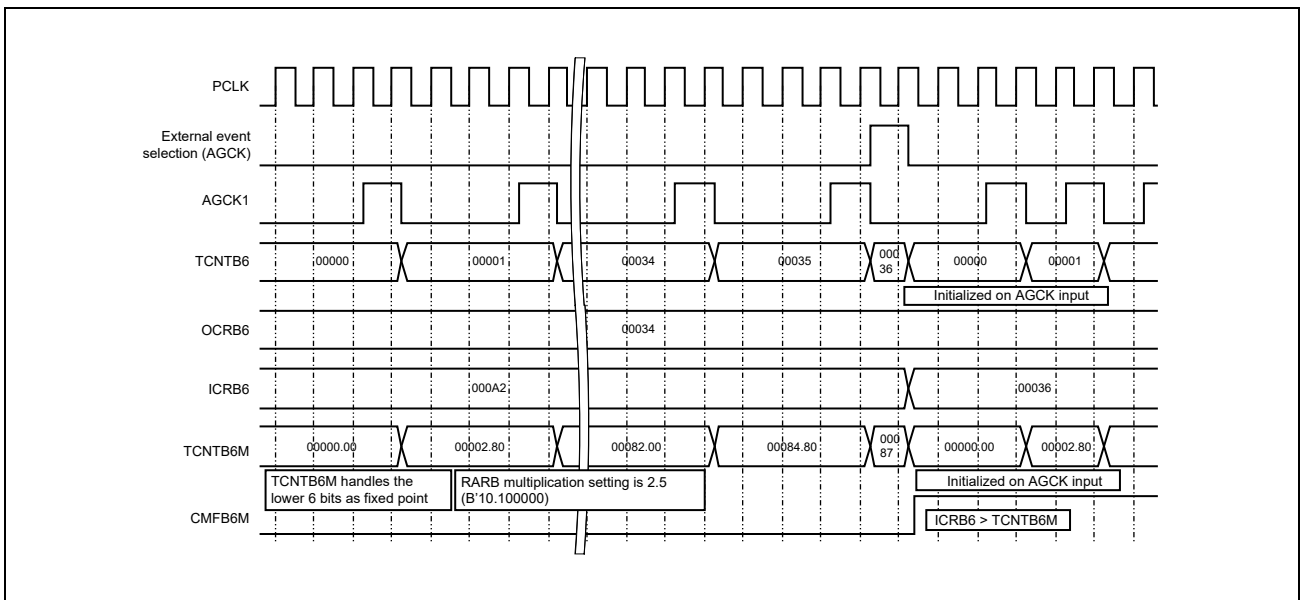


Figure 30.29 Example of Operations of TCNTB6M and ICRB6

30.6.3.3 Frequency-Multiplied Clock Signal Corrector

The frequency-multiplied clock signal which is generated by dividing the intervals between external event selection (AGCK) inputs by the multiplication ratio set in the PIM bits in PIMR can be corrected by using 20-bit correcting event counter B3 (TCNTB3), 20-bit multiplied-and-corrected clock counter B4 (TCNTB4), 20-bit multiplied-and-corrected clock generating counter B5 (TCNTB5), and correcting counter clearing register B (TCCLRb).

TCNTB3 is a 20-bit up-counter that is driven by the external event selection (AGCK) input. On the selected transition of the external event selection (AGCK) signal, the value in TCNTB3 is transferred to TCNTB4, after which TCNTB3 is incremented by the value in the PIM bits ($TCNTB3+ = PIM$). According to the setting for the CLRb3SEL bit (TCNTB3 clear select bit) in the timer control register B (TCRB), TCNTB3 can be cleared by a compare match between TCNTB1 and OCRB10*¹, between TCNTB6 and OCRB6, or between TCNTB3 and OCRB8.

TCNTB4 is a 20-bit up-counter that is driven by the multiplied clock signal (AGCK1). TCNTB3 is loaded to TCNTB4 with the external event selection (AGCK) input as a trigger, and incrementation of TCNTB4 is driven by the AGCK1 input.

The counter correcting select bit (CCS) in TIORB controls counting by TCNTB4. When CU4SEL is set to “0”, movement or a stop of TCNTB4 up count can be select at $TCNTB3 = TCNTB4$, and when CU4SEL is set to “1”, movement or a stop of TCNTB4 up count can be select at $(TCNTB3-1) = TCNTB4$.

TCNTB5 is a 20-bit up-counter that is driven by the PCLK clock, meaning that it operates at a high speed. TCNTB5 is constantly compared with TCNTB4 and is incremented as long as its value is lower than that in TCNTB4. Each time TCNTB5 is actually incremented, it produces a single pulse whose width is equal to one cycle of the PCLK clock. Pulse of the peripheral clock signal, namely the multiplied-and -corrected clock signal (AGCKM), for which output on clock-bus line 5 can be selected by bit CB5SEL in the clock bus control register (CBCNT). The AGCKM signal is then available on clock-bus line 5 as a source to drive counting by other timers.

As state above, TCNTB5 is not incremented when its value is greater that in TCNTB4 (for example, after TCNTB3 has been loaded to TCNTB4), TCNTB5 can also be disabled by the count control B5 (CTCNTB5) bit in timer I/O control register B (TIORB). This halts the output of the AGCKM signal.

As long as its value is lower than TCNTB4, TCNTB5 is incremented until it reaches the value in correcting counter clearing register B (TCCLRb). This function allows you to check the automatic up- count status by using correcting counter clear flag register B (TCCLFRb). Incrementation of TCNTB5 then stops, regardless of the relation between its value and that of TCNTB4. In addition, timer counters 1Dx and 2Dx of timer D (TCNT1Dx and TCNT2Dx) can be separately cleared by this match as a trigger when the corresponding counter clearing enable bit (C1CEDx/C2CEDx) in timer control register Dx (TCRDx) is set to 1.

TCNTB4 is unconditionally cleared to 0000 0000_H when a pulse of the external-event selection signal (AGCK) is input while TCNTB3 = 0000 0000_H. TCNTB5 is unconditionally set to 0000 1000_H at CU4SEL = 0, or cleared to 0000 0000_H at CU4SEL = 1 when a pulse of the external-event selection signal (AGCK) is input while TCNTB3 = 0000 0000_H.

However, when TCNTB5 has not reached TCCLRb, TCNTB5 is incremented until it reaches TCCLRb. After that, it is set to 0000 1000_H.

Figure 30.30 shows operations of TCNTB3 and TCNTB4, **Figure 30.31** shows operation when TCNTB5 is being started up, **Figure 30.32** shows TCNTB5 operation with correction at the end of a cycle.

Note 1. When CHSELB [1:0] is 00_B, TCNTB3 can be cleared by a comparing match of TCNTB1 and OCRB10. When CHSELB [1:0] is 01_B, TCNTB3 can be cleared by a comparing match of TCNTB1S1 and OCRB10S1. When CHSELB [1:0] is 10_B, TCNTB3 can be cleared by a comparing match of TCNTB1S2 and OCRB10S2.

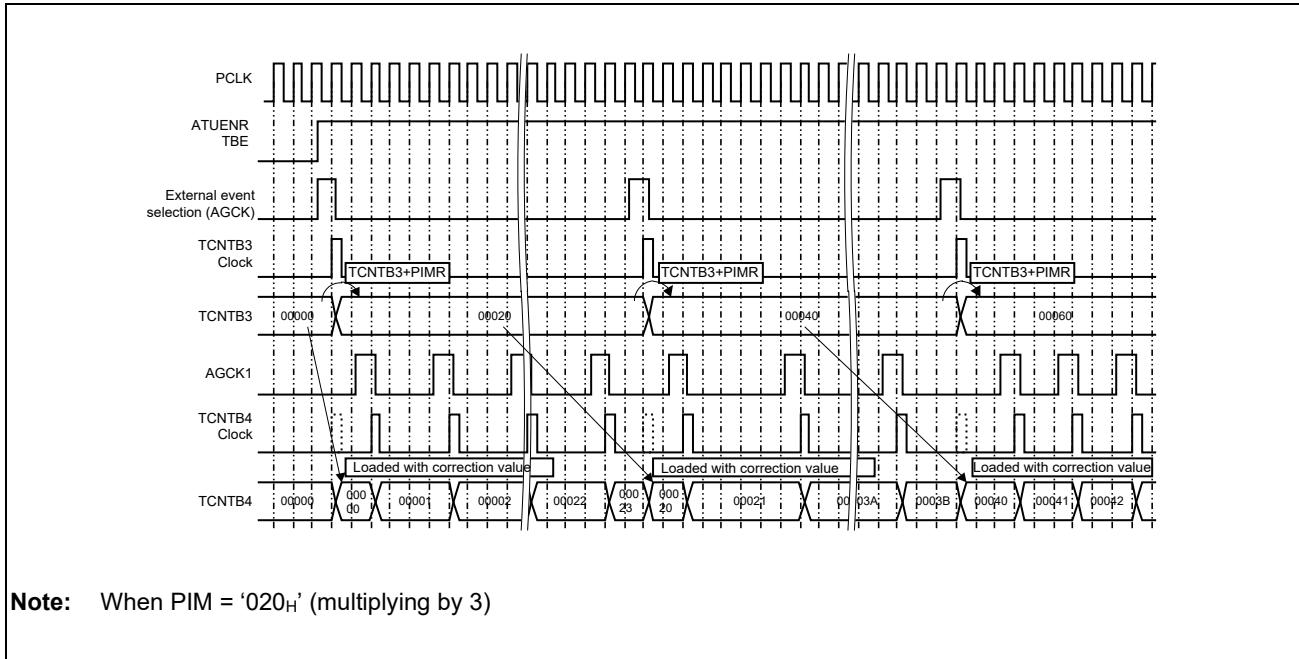


Figure 30.30 Operation of TCNTB3 and TCNTB4 (at CU4SEL = 0)

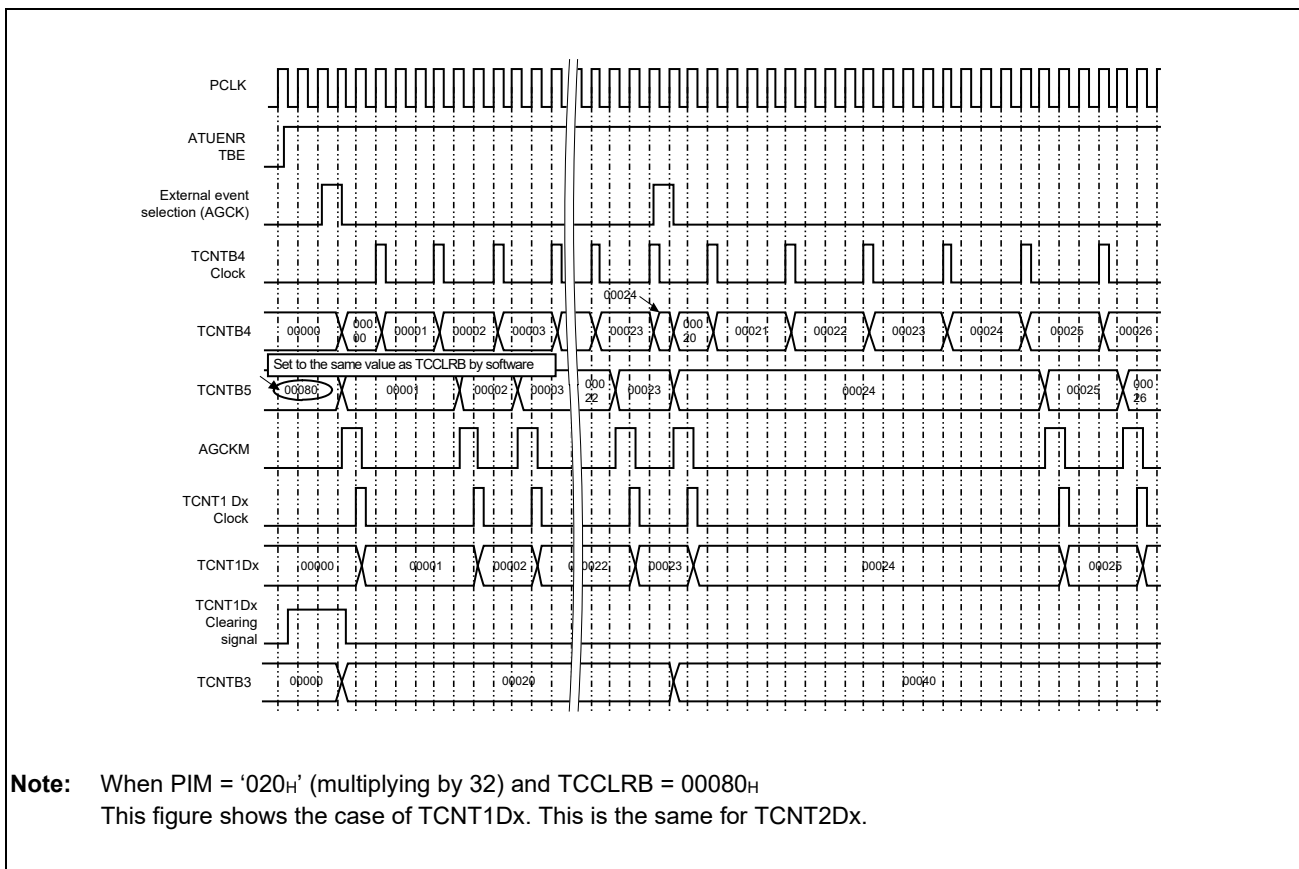


Figure 30.31 TCNTB5 Operation (when CU4SEL = 0 at Start-Up)

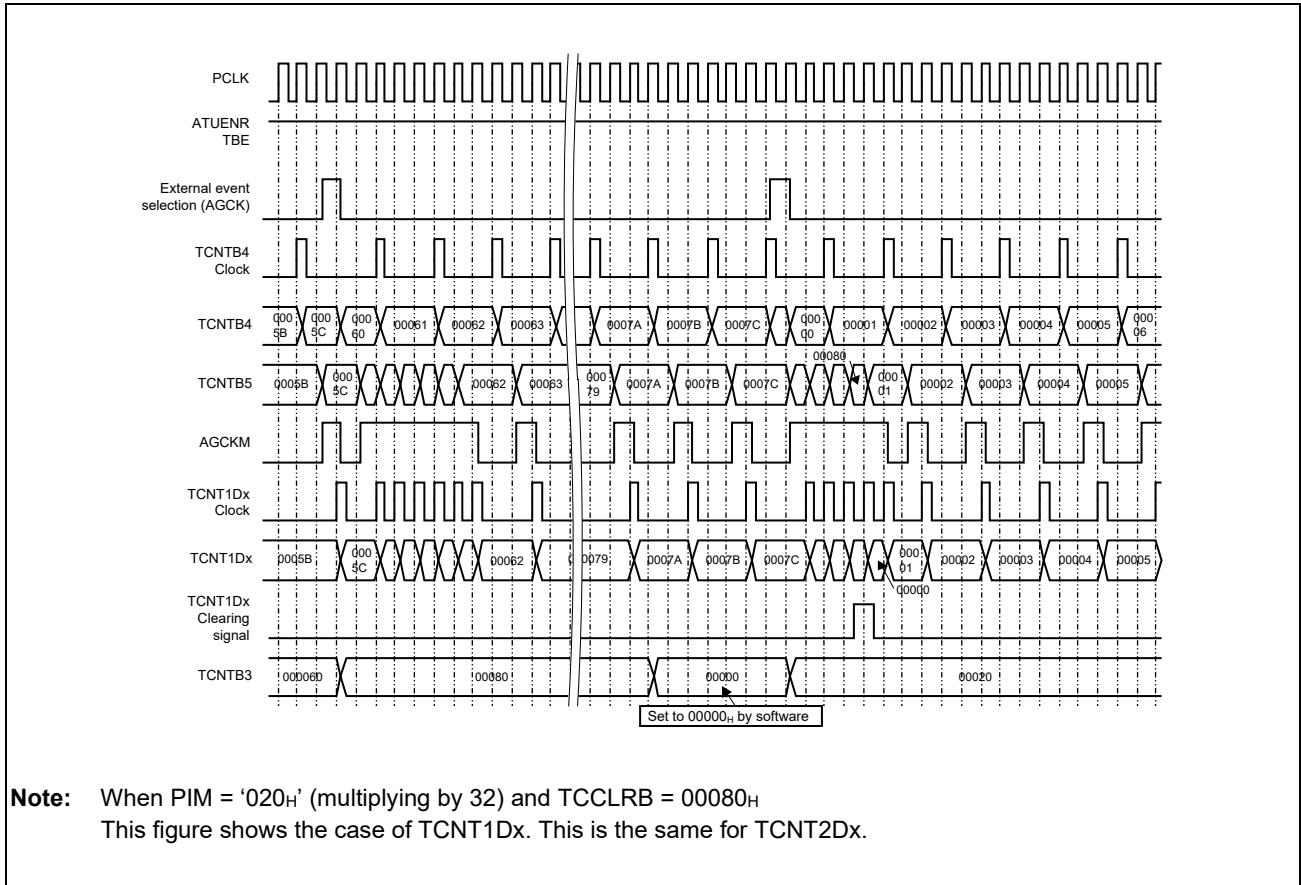


Figure 30.32 Operation of TCNTB5 (when CU4SEL = 0, with Correction at End of Cycle)

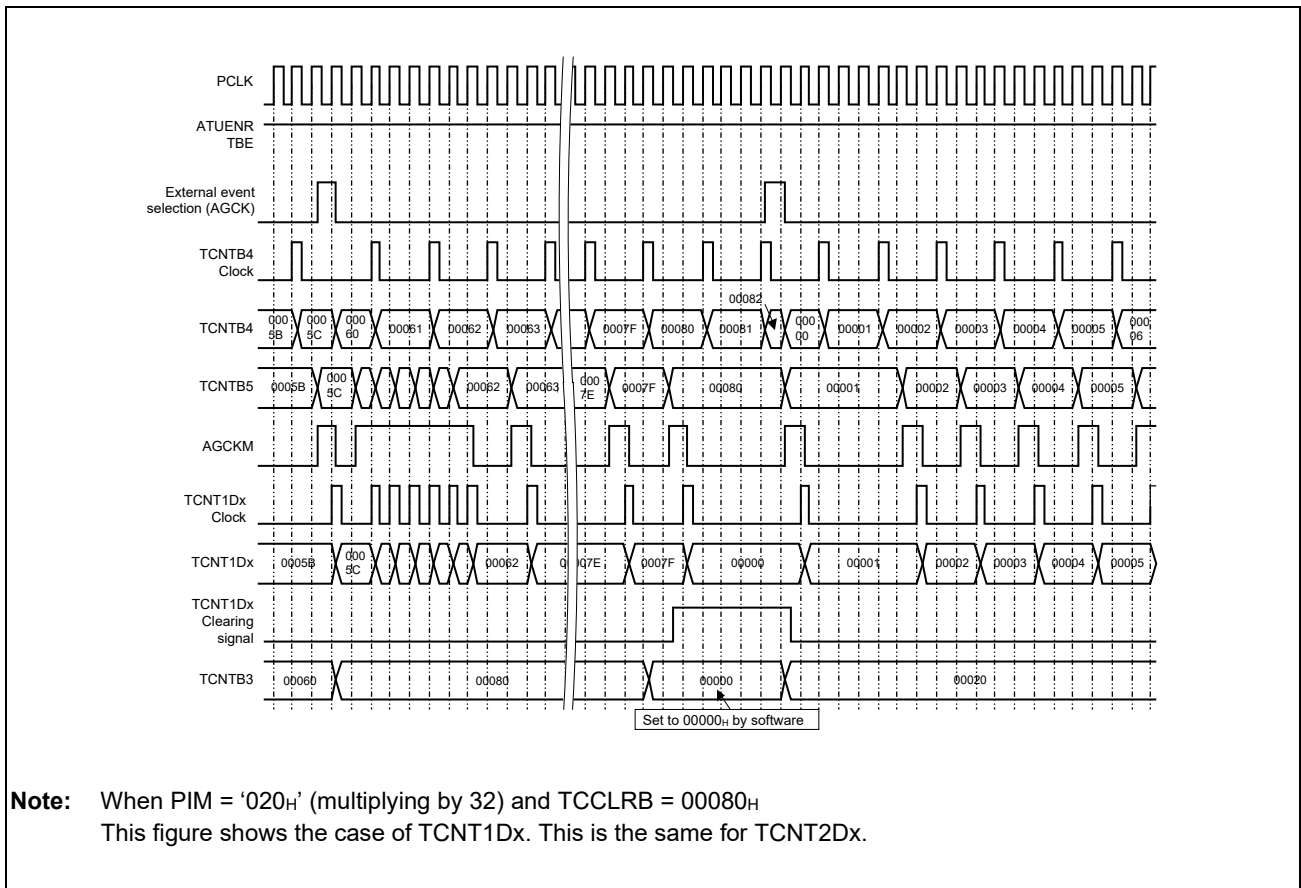


Figure 30.33 Operation of TCNTB5 (when CU4SEL = 0 without Correction at End of Cycle)

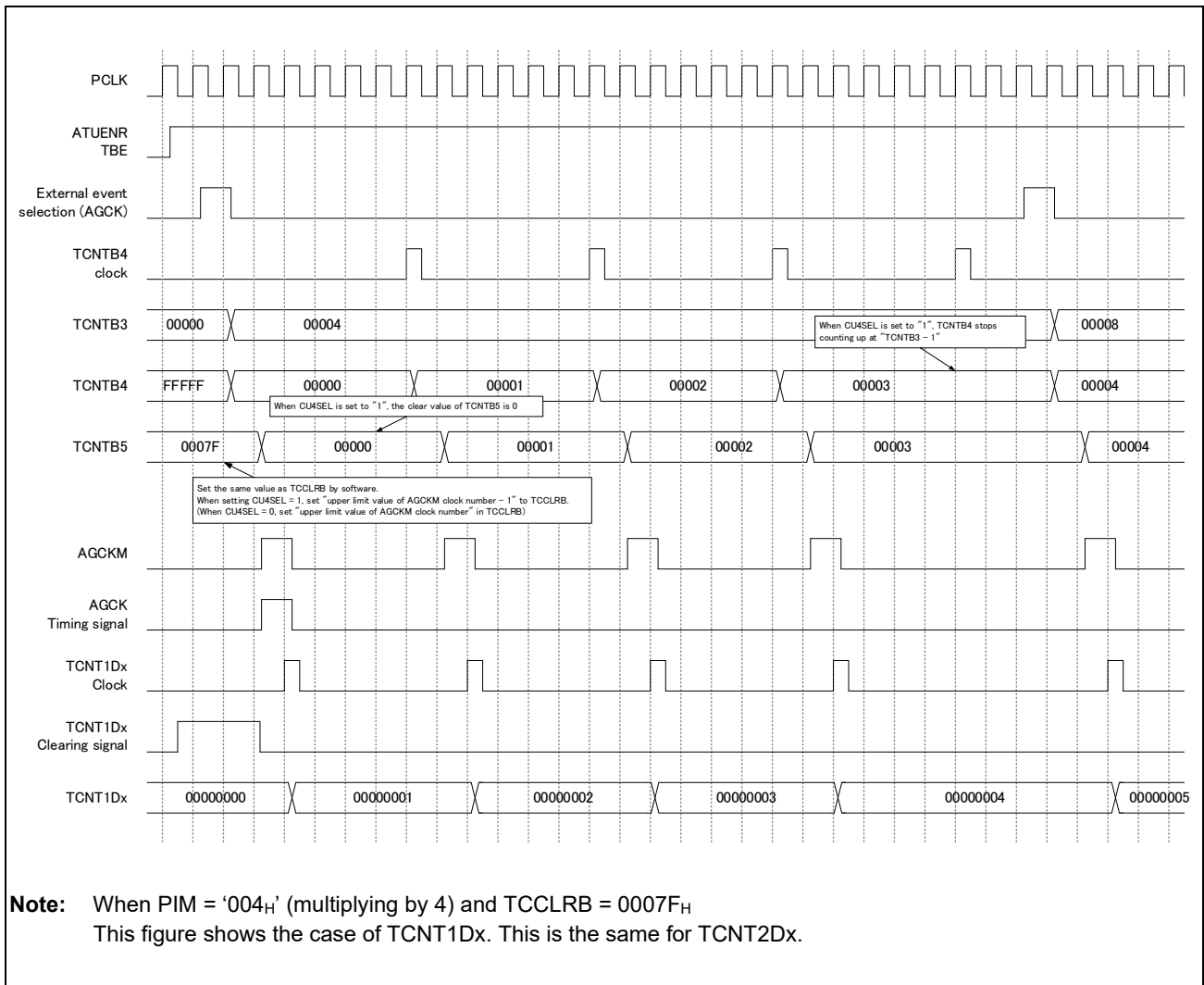


Figure 30.34 TCNTB3, TCNTB4, TCNTB5 Operation (when CU4SEL = 1 at Start-Up)

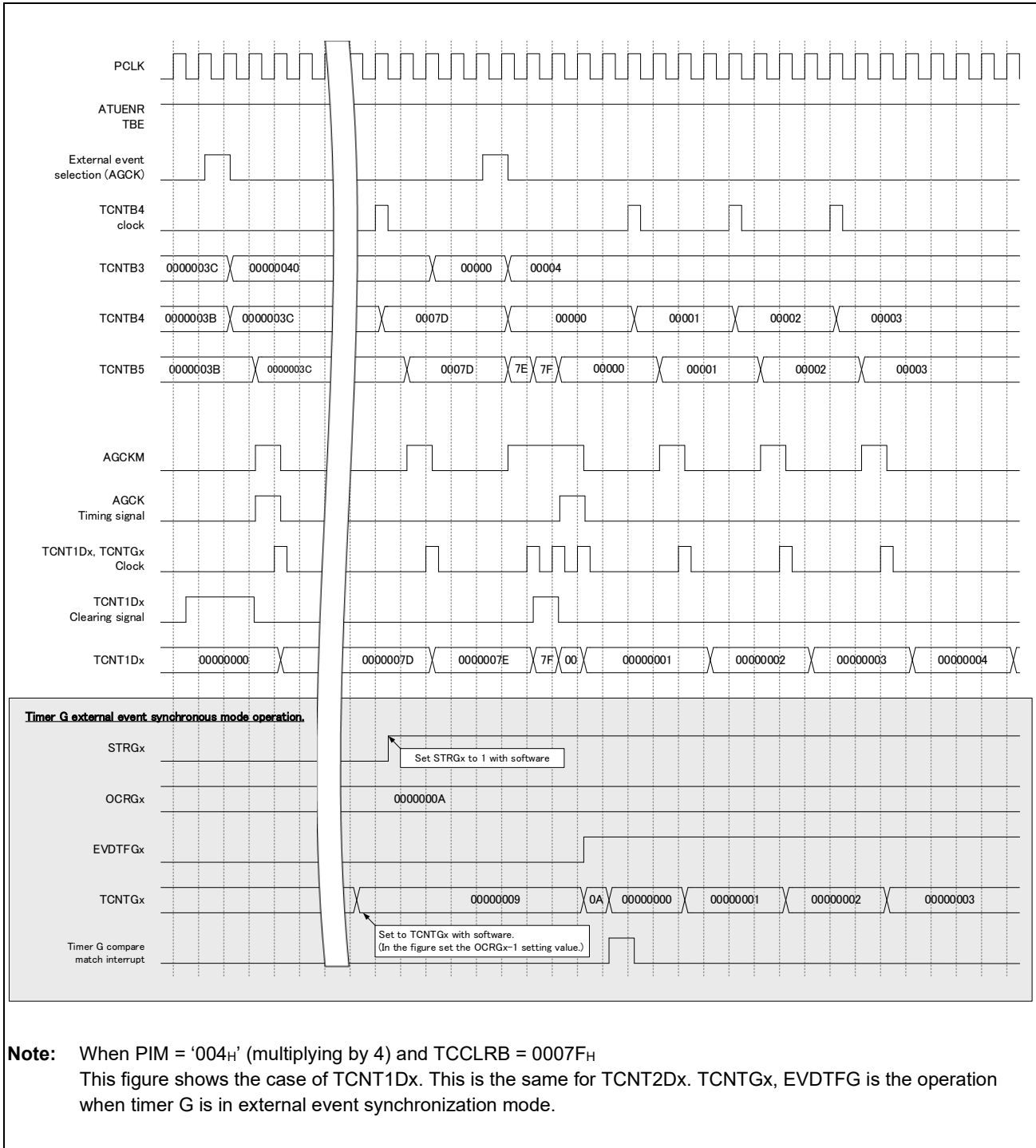


Figure 30.35 Operation of TCNTB3, TCNTB4, TCNTB5 (when CU4SEL = 1, with Correction at End of Cycle)

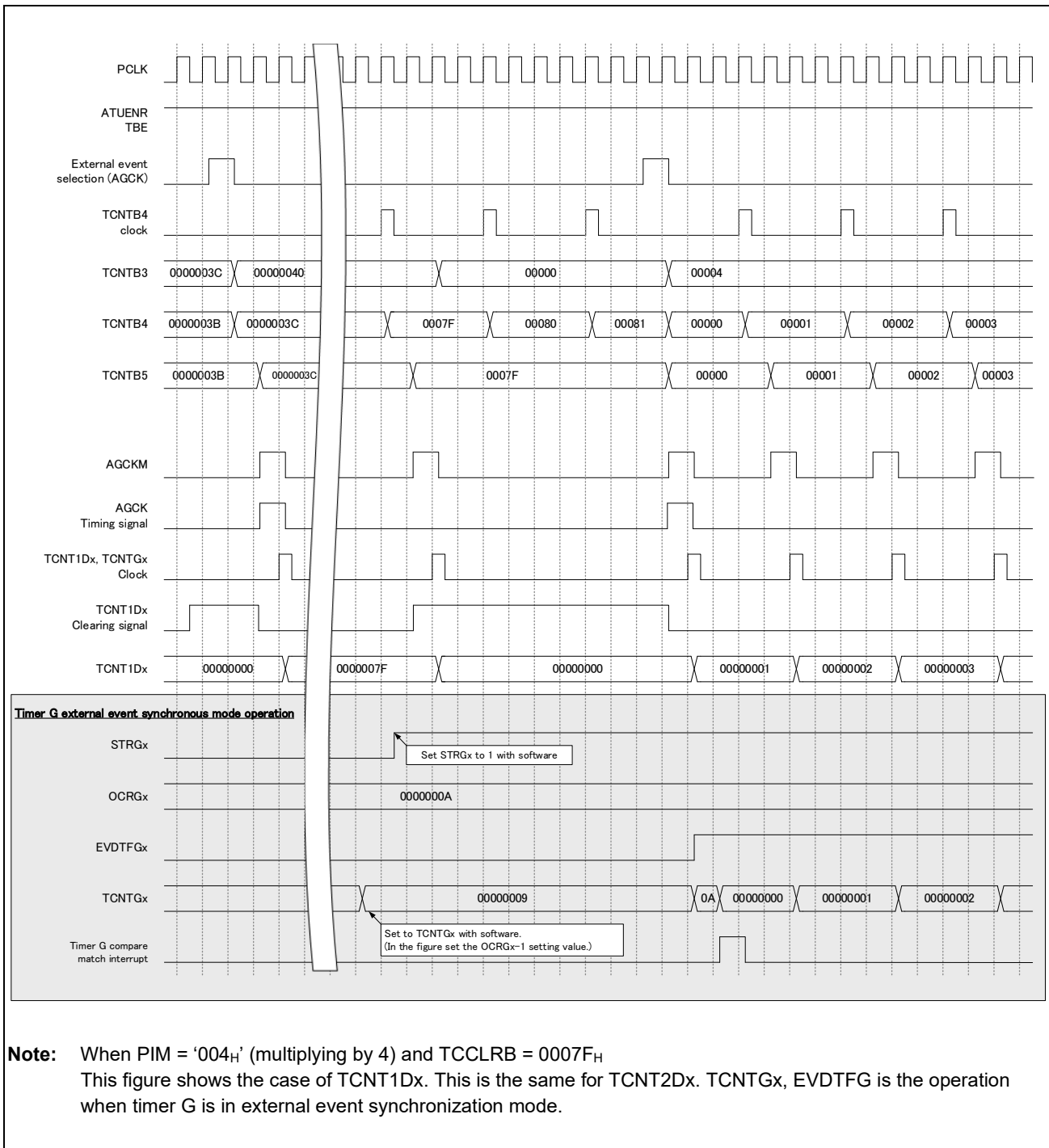


Figure 30.36 Operation of TCNTB3, TCNTB4, TCNTB5 (when CU4SEL = 1 without Correction at End of Cycle)

30.7 Timer C

30.7.1 Overview of Operation

The timer C block is a general-purpose timer consisting of SB_C subblocks having the same function. Each subblock consists of four channels.

Each timer C subblock can achieve the following functions.

- Input capture and output compare matches
- Choice of rising edge, falling edge, or both edge sensing as the edge of input capture trigger signal
- Capturing at the timing of event output 1_0, 2A, or 2B from timer A when used as input capture
- Output of a waveform on compare match. Choice of a logical one, a logical zero, a toggled output, or a one-shot pulse output by setting a register
- Output of an interrupt request on capture or compare match
- Output of an interrupt request on timer counter overflow
- Clearing counter (TCNTCx) on compare match (two types)
 - The counter can be cleared on compare match of GRCx0 in PWM mode (not supported by GRCx1 to GRCx3)
 - When the counter upper-limit setting function is used in any mode other than PWM mode, the counter can be cleared on compare match of CUCRCx.
- Output of forced compare match by setting the forced compare match bit
- There is noise cancellation of at most 0.21 sec or a noise cancellation function by the angle clock in each input terminal.
- A noise cancellation mode setting by the subblock/channel unit is possible.
- High/Low/toggle corrugated output is possible by a comparing match.
- The range comparison can be used for compare match output by the compare match register by setting the range comparison value setting register.

Timer C subblock consists of a 32-bit counter (TCNTCx), four 32-bit general registers (GRCxy), four 32-bit output compare registers (OCRCxy), a 32-bit counter upper-limit setting compare register (CUCRCx), four range comparison value setting registers (RCRCxy), four 32-bit general mirror registers (GMRCxy), and controller. The general registers (GRCxy) can be used for input capture/compare match operations and input capture trigger input and output compare output signals (TIOCxy) are available.

The initial output value on the TIOCxy pin is 0 for output compare operation. During operation, the previous state is reflected. An input/output mode of a TIOCxy terminal is selectable by setting of TIORCx.IOCxy.

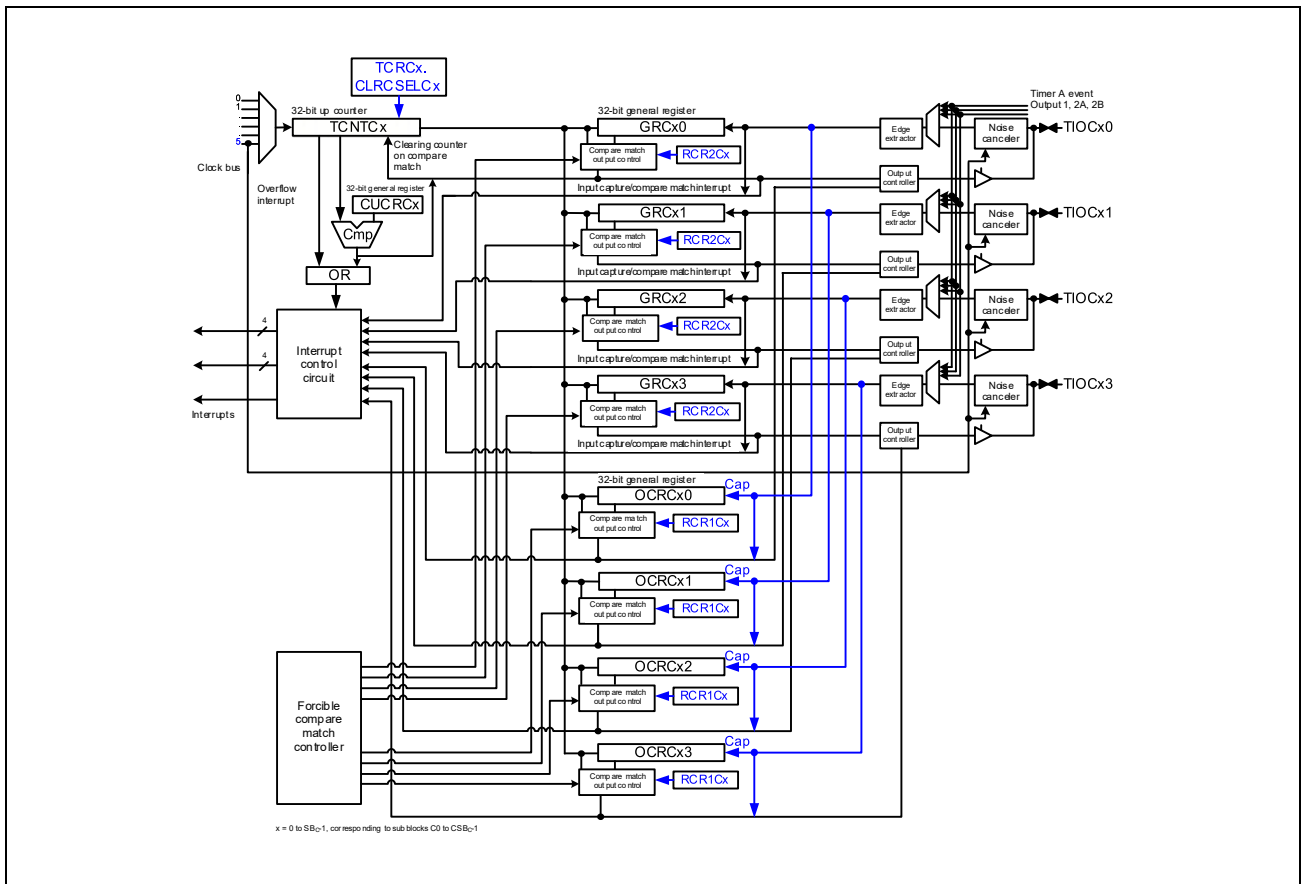


Figure 30.37 Block Diagram of Timer C Subblock

30.7.2 Registers Related to Timer C

30.7.2.1 TSTRC — Timer Start Register C

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	STRC1 0	STRC9	STRC8	STRC7	STRC6	STRC5	STRC4	STRC3	STRC2	STRC1	STRC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.107 TSTRC Register Contents

Bit Position	Bit Name	Function
15 to SB _C	—	These bits are not used. Fix these bits to “0”.
SB _C -1 to 0	STRC _x	Counter C _x start 0: TCNTC _x is disabled 1: TCNTC _x is enabled

Note: Start bits are provided for subblocks C0 to CSB_C-1 respectively.

TSTRC is an 8-bit or a 16-bit readable/writable register that enables and disables timer counter (TCNTC_x) in SB_C subblocks C_x (x = 0 to SB_C-1). When the both the STRC bits in this register and the TCE bit in ATU master enable register (ATUENR) are set to 1, counting is enabled.

TSTRC is initialized to 00_H after reset.

Counter C_x Start (STRC_x)

These bits enable and disable timer counter C_x (TCNTC_x) in a subblock.

When bit STRC_x is cleared to 0, TCNTC_x is stopped. While TCNTC_x is stopped, the previous counter value is retained and TCNTC_x is resumed from the value when this bit is set to 1 again.

Even if the counter C_x start bit is set to 1, counting does not start unless the timer C_x master enable bit in the ATU master enable register (ATUENR) is set to 1.

CAUTION

The prescalers run independently of the setting of the STRC_x bit and are not initialized at the start of TCNTC_x. Therefore, during the time between the activation and the start of actual count operation by the above counter, hardware-related uncertainty shorter than the period of selected count source (resolution) accompanies.

30.7.2.2 NCCRCx — Noise Canceler Control Register Cx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NCKCx3	NCKCx2	NCKCx1	NCKCx0	NCECx3	NCECx2	NCECx1	NCECx0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_C-1: Corresponding to subblocks C0 to CSB_C-1

Table 30.108 NCCRCx Register Contents

Bit Position	Bit Name	Function
7 to 4	NCKCxy	Noise canceler clock select These bits select the count source clock for noise canceler counter Cxy (NCNTCxy).
3 to 0	NCECxy	Noise canceler enable Cxy These bits set the TIOCxy input noise canceling function.

Note: x = 0 to SB_C-1, y = 0, 1, 2, 3

The noise canceler control register Cx (NCCRCx) is an 8-bit readable/writable register. The noise cancellation is performed on the input capture trigger signal input from pin TIOCxy in subblock Cx by setting this register. Three modes are available in noise cancellation and can be switched by the NCMC bit in the noise cancel mode register (NCMR). Noise canceler count clock or clock bus 5 can be selected as the count source by the noise canceler clock select bits (NCKCxy) in this register.

NCCRCx is initialized to 00_H after reset.

Noise Canceler Clock Select (NCKCxy)

These bits select the count source clock of noise canceler counter Cx (NCNTCx).

NCKCxy	Function
0	Noise canceler count clock (PCLK or PCLK/128) is selected as the count source clock of NCNTCx (Default)
1	Clock bus 5 is selected as the count source clock of NCNTCx

Note: x = 0 to SB_C-1, y = 0, 1, 2, 3

Selects the count source clock for noise cancellation counters C4 to C0 (NCNTC4 to NCNTC0).

Noise canceler count clock or clock bus 5 is selectable as the count source clock. As the noise canceler count clock, a clock of PCLK or PCLK/128 can be selected by the NCCSEL bit in the common control unit.

Noise canceler Enable Cxy (NCECxy)

These bits enable or disable the noise canceling function for input/output pins TIOCx0 to Cx3 of subblock Cx.

NCECxy	Function
0	The TIOCxy input noise canceling function is disabled (Default)
1	The TIOCxy input noise canceling function is enabled.

Note: x = 0 to SB_C-1, y = 0, 1, 2, 3

When a change in input signal level from TIOC_{xy} is detected with these bits set to 1, processing for premature-transition cancellation mode, minimum time-at-level cancellation mode, or level accumulation cancellation mode starts according to the settings of the noise cancellation mode register (NCMR) and the noise canceler clock select bit (NCKC) in the common control unit.

- In premature-transition cancellation mode

When a level change of the externally input signal is detected, the change is output as the signal whose noise is removed and the corresponding noise canceler counter C_{xy} (NCNTC_{xy}) is started for counting up. Subsequent level changes are masked until the value in the counter reaches the value in the noise canceler register C_{xy} (NCRC_{xy}). The level of the externally input signal is output on this compare match.

When these bits are cleared to 0 while the counter (NCNTC_{xy}) is being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes are also masked over this period.

- In minimum time-at-level cancellation mode

When a level change of the externally input signal is detected, the corresponding noise canceler counter C_{xy} (NCNTC_{xy}) is started for counting up. If subsequent level changes are not detected until the value in the counter reaches the value in the noise canceler register C_{xy} (NCRC_{xy}), the previously accepted level change is output as the signal whose noise is removed on compare match of the counter and noise canceler register. When the subsequent level change is detected before the values in the counter and noise canceler register match, all the changes are treated as noise. Therefore the signal whose noise is removed is not changed.

When these bits are cleared to 0 while the counter (NCNTC_{xy}) is being incremented, counting continues until the values in the counter and the noise canceler register match or a level change on the externally input signal is detected.

- In level accumulation cancellation mode

The corresponding noise canceler counter C_{xy} (NCNTC_{xy}) is started for counting up or down according to the input signal level (high level: up-count, low level: down-count). Up-count continues until the noise canceler counter value matches the noise canceler register C_{xy} (NCRC_{xy}) value. Down-count continues until the noise canceler counter value reaches 0000_H. When a compare match (NCRC_{xy}) occurs during up-counting, the noise canceler output is updated to 1. When a compare match (0000_H) occurs during down-counting, the noise canceler output is updated to 0.

In minimum time-at-level cancellation mode and premature-transition cancellation mode, clock PCLK is always used to detect input signal level changes regardless of the selected noise canceler clock. In level accumulation cancellation mode, the clock selected by the noise canceler clock select C_{xy} bits (NCKC_{xy}) is used for sampling the input level.

For details of operations for noise cancellation, see **Figure 30.1**, **Figure 30.2**, and **Figure 30.3**.

30.7.2.3 TCRCx — Timer Control Registers Cx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FOCMC _{x3}	FOCMC _{x2}	FOCMC _{x1}	FOCMC _{x0}	—	—	CLRCS ELCx	CLRCx	FCMCx 3	FCMCx 2	FCMCx 1	FCMCx 0	PWMx0	CKSELcx[2:0]		
Value after reset	0*1	0*1	0*1	0*1	0	0	0	0	0*1	0*1	0*1	0*1	0	0	0	0
R/W	W	W	W	W	R	R	R/W	R/W	W	W	W	W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_C-1, corresponding to subblocks C0 to CSB_C-1

Note 1. Value 0 cannot be written. The value is not retained when 1 is written. These bits are always read as 0.

Table 30.109 TCRCx Register Contents

Bit Position	Bit Name	Function
15 to 12	FOCMC _{xy}	Forced output compare match C _{xy} Forced compare match is performed by output compare register (OCRC _{xy}). ^{*1}
11 to 10	—	These bits are not used. When read, "0" is always returned. When writing, always write 0.
9	CLRCS ELCx	TCNTCx clear timing select bit C _x Select the timing for clearing TCNTCx. 0: Clear in sync with PCLK (equivalent to ATU-IV) 1: Clear in sync with the count clock
8	CLRCx	TCNTCx clear C _x Enable and disable clearing of the timer counters.
7 to 4	FCMC _{xy}	Forced compare match C _{xy} Forced compare match is performed by general register (GRC _{xy}). ^{*1}
3	PWMx0	PWM mode Whether to operate subblock C _x in PWM mode is selected.
2 to 0	CKSELcx[2:0]	Clock select bits The count clock is selected from clock bus 0 to 5.

Note: x = 0 to SB_C-1. Corresponding to subblocks C0 to CSB_C-1.

y = 0, 1, 2, 3: Corresponding to general registers C_{x0}, C_{x1}, C_{x2}, and C_{x3}.

Note 1. Do not use forced compare match in input capture mode.

The timer control registers C_x (TCRC_x) are 8-bit/16-bit readable/writable registers that select the counter clock for subblocks C0 to CSB_C-1, set operating mode, set forced compare matches, and enable/ disable clearing of timer counters.

Even if the respective "Forced output compare match C_{xy}(FOCMC_{xy})" and "Forced compare match C_{xy}(FCMC_{xy})" are made effective when using "Output Compare Registers C_{xy}(OCRC_{xy})" and "Timer General Registers C_{xy}(GRC_{xy})" as a capture register, a forced compare matches doesn't occur.

TCNTCx Clock Select (CKSELCx[2:0])

These bits select the counter clock of subblock Cx. Counters (TCNTCx) in subblock Cx are driven by the clock selected in these bits.

CKSELCx			Function	
0	0	0	Counters are driven by clock-bus line 0	(Default)
0	0	1	Counters are driven by clock-bus line 1	
0	1	0	Counters are driven by clock-bus line 2	
0	1	1	Counters are driven by clock-bus line 3	
1	0	0	Counters are driven by clock-bus line 4	
1	0	1	Counters are driven by clock-bus line 5	
1	1	0	Setting prohibited	
1	1	1	Setting prohibited	

Note: x = 0 to SB_C-1. Corresponding to subblocks C0 to CSB_C-1.

PWM Mode (PWMx0)

Setting this bit to 1 makes subblock Cx operate in PWM mode. In PWM mode, TCNTCx is cleared on compare match between TCNTCx and general register Cx0 (GRCx0). The setting of this bit is valid when GRCx0 functions as a compare match register.

When general register GRCxm (m = 1 to 3) functions as a compare match register, aIOCx0 setting value in the TIORCx is output to its output pins (TIOCx1, TIOCx2, TIOCx3) at the timing of clearing of the TCNTCx.

PWM output from pins TIOCx1 to TIOCx3 is enabled by setting a PWM cycle in GRCx0 and setting a duty cycle in GRCx1 to GRCx3.

CAUTION

To make the subblock operate in PWM mode, further settings are also required.

Select the compare match (IOCxy[3:0] = 0001_B or 0010_B) by the IOCxy bits in timer I/O control register C (TIORCx) for GRCx0 and GRCx1 to GRCx3 (for PWM output). The active polarity of PWM can be specified by the IOCxy[1:0] bits.

PWMx0	Function	
0	0: Subblock Cx does not operate in PWM mode	(Default)
1	1: Subblock Cx operates in PWM mode	

Note: x = 0 to SB_C-1. Corresponding to subblocks C0 to CSB_C-1.

When TCNTCx matches GRCx0 while this bit is set to 1, the counter is cleared to 0000 0000_H. However, when clearing the counter on compare match and incrementation occur simultaneously, TCNTCx is set to 0000 0001_H. This occurs when TCNTCx is driven by the clock whose frequency is equal to the PCLK clock.

In PWM mode, do not set from GRCx0 to GRCx3 to 0000 0000_H. If GRCx0 is set to 0000 0000_H, note that compare match occurs at illegal cycles.

Forced Compare Match Cxy (FCMCxy)

Setting these bits to 1 generates forced compare match when a general register Cxy (GRCxy) is used for compare match. Refer to **Figure 30.44**, **Figure 30.45**, **Figure 30.46**, and **Figure 30.47** as a use example.

FCMCxy	Function
0	No forced compare match is disabled on general register (GRCxy) (Default)
1	Forced compare match generated on general register (GRCxy)

Note: x = 0 to SB_C-1. Corresponding to subblocks C0 to CSB_C-1.
y = 0, 1, 2, 3: Corresponding to general registers Cx0, Cx1, Cx2, and Cx3.

When GRCxy is used as a comparing register at a comparing match mode (TIORCx.IOCxy [3:2] = 00_B), Setting the FCMCxy bit to 1 sets the IMFCxy bit (compare match flag) in TSRCx (timer status register Cx) to 1 and outputs a compare match interrupt to the CPU. Even when the IMFCxy bit is set to 1, a compare match interrupt is output. Furthermore, TIOCxy output data also changes as in the case of compare match. This operation is performed only once when the FCMCxy bit is set to 1. Writing 0 to these bits is disabled. If 1 is written to these bits, data is not retained. These bits are always read as 0.

Even if a forced compare match is established effectively, then a forced compare match doesn't generate an input capture mode (TIORCx.IOCxy [3:2] = 01_B or 11_B). A one shot pulse mode (TIORCx.IOCxy [3:2] = 10_B), then FCMCxy setting becomes effective after GRCxy setting with a one shot pulse mode programming procedure. Even if FCMCxy is made effective before GRCxy setting without following a one shot pulse mode programming procedure, a forced compare match doesn't occur. The change of state of TSRCx in the back writing in became effective where to FCMCxy is same as the time of a compare match mode.

The FCMCxy bit can be established only by the following condition.

Mode	TIORCx.IOCxy				TCRCx.PWMx0*1	The setting propriety of the FCMCxy bit
	[3]	[2]	[1]	[0]		
Compare match mode	0	0	0	0	0	Setting prohibited
	0	0	0	1	0	Possible to establish
					1	Setting prohibited
	0	0	1	0	0	Possible to establish
					1	Setting prohibited
	0	0	1	1	0	Possible to establish
Input capture mode	0	1	0/1	0/1	0	Setting prohibited
One shot pulse mode	1	0	0/1	0/1	0	Possible to establish
Input capture mode	1	1	0/1	0/1	0	Setting prohibited

Note 1. TCRCx.PWMx0 can set only at TIORCx.IOCxy = 0001_B or 0010_B.

TCNTCx Clear Cx (CLRCx)

When the TCNTCx clear bit is set to 1, the TCNTCx is cleared in synchronization with PCLK after occurrence of a compare match between timer counter Cx (TCNTCx) and counter upper limit value setting compare register (CUCRCx) or in synchronization with count clock input (according to the setting of CLRCSELx).

CLRCx	Function	
0	CUCRCx compare match is not used to clear TCNTCx.	(Default)
1	CUCRCx compare match is used to clear TCNTCx	

Note: x = 0 to SB_C-1. Corresponding to subblocks C0 to CSB_C-1. Supports timer counter Cx.

The upper limit value setting function can be enabled (CLRCx = 1) in other than PWM mode (PWMx0 = 0).

If the upper limit value setting function (CLRCx = 1) is used in PWM mode (PWMx0 = 1), timer counter Cx (TCNTCx) is cleared in a cycle other than PWM cycle, so the PWM output may become an unintended waveform.

The upper limit value of the counter is different depending on setting. Refer to the following table.

PWMx0	CLRCx	The upper limit value of the TCNTCx counter
1	0	(Setting prohibited) The value of GRCx0 (PWM cycle)
0	0	FFFF_FFFF _H
0	1	The value of CUCRCx (When the writing in value to CUCRCx is smaller than TCNTCx, it'll be FFFF_FFFF _H .)

TCNTCx Clear Timing Select Cx (CLRCSELCx)

Select the timing for clearing timer counter Cx (TCNTCx) after occurrence of a compare match between timer counter Cx (TCNTCx) and counter upper limit value setting compare register (CUCRCx).

CLRCSELCx	Function	
0	Clear in sync with PCLK (equivalent to ATU-IV)	(Default)
1	Clear in sync with count clock	

Note: x = 0 to SB_C-1. Corresponding to subblocks C0 to CSB_C-1 of timer counter Cx

CLRCSELCx can be set at all except for a PWM mode (PWMx0 = 0).

When comparing match occurrence with Timer Counter Cx(TCNTCx) and Counter Upper-Limit Setting Compare Registers Cx(CUCRCx) and a count up of TCNTCx occurred at the same time in case of CLRCSELx = 0, TCNTCx is cleared by 0000 0001_H. When a count up isn't simultaneous with a compare match, timer counter Cx (TCNTCx) is cleared by 0000 0000_H. Timer counter Cx (TCNTCx) doesn't depend on division ratio of a count clock in case of CLRCSELx = 1, and is cleared in 0.

PWMx0	The setting propriety of CLRCSELCx
0	It's possible to set it as 0 or 1.
1	Setting prohibited (The PWM cycle is prohibited to collapse)

Forced Output Compare Match Cxy (FOCMCxy)

When the output compare register (OCRCxy) is used as a compare match register, a compare match is forcibly generated by setting FOCMCxy to 1. Refer to **Figure 30.44**, **Figure 30.45**, **Figure 30.46**, and **Figure 30.47** as a use example.

FOCMCxy	Function
0	Forced compare match is disabled for the output compare register (OCRCxy). (Default)
1	Forced compare match is generated in the output compare register (OCRCxy).

Note: x = 0 to SB_C-1. Corresponding to subblocks C0 to CSB_C-1.
y = 0, 1, 2, 3: Corresponding to output compare registers Cx0 to Cx3.

When OCRCxy is used as a compare register by a compare match mode (TIORCx.IOCxy [3:2] = 00_B) and an input capture mode (TIORCx.IOCxy [3:2] = 01_B or 11_B), setting the FOCMCxy bit to 1 sets the OCMFCxy bit (compare match flag) in TSRCx (timer status register Cx) to 1 and outputs a compare match interrupt to the CPU. Even when the OCMFCxy bit is 1, a compare match interrupt is generated and the output from TIOCx changes as when a compare match occurs. This operation is performed only once when 1 is written to this bit. Writing 0 is disabled. If 1 is written to this bit, data is not retained. This bit is always read as 0.

Even if a forced compare match is established effectively when using OCRCxy as a capture register by an input capture mode (TIORCx.IOCxy [3:2] = 01_B or 11_B), a forced compare match doesn't occur. A one shot pulse mode (TIORCx.IOCxy [3:2] = 10_B), then FOCMCxy setting becomes effective after GRCxy setting with a one shot pulse mode programming procedure. Even if FOCMCxy is made effective before GRCxy setting without following a one shot pulse mode programming procedure, it becomes invalid.

Even if FOCMCxy is also made effective before GRCxy setting after a one shot pulsed output, it becomes invalid.

The change of state of after TSRCx writing in to FOCMCxy became effective where is same as a compare match mode and the time of an input capture.

The FOCMCxy bit can be established only by the following condition.

Mode	TIORDCx.IOCxy				TCRCx.PWMx0*1	The setting propriety of the FOCMCxy bit
	[3]	[2]	[1]	[0]		
Compare match mode	0	0	0	0	0	Possible to establish
	0	0	0	1	0	Possible to establish
					1	Setting prohibited
	0	0	1	0	0	Possible to establish
					1	Setting prohibited
0	0	1	1	0	Possible to establish	
Input capture mode	0	1	0	0	0	Possible to establish
	0	1	0	1	0	Setting prohibited
					0	Possible to establish
	0	1	1	1	0	Possible to establish
One shot pulse mode	1	0	0/1	0/1	0	Possible to establish
Input capture mode	1	1	0/1	0/1	0	Possible to establish

Note 1. TCRCx.PWMx0 can set only at TIORC.IOCxy = 0001_B or 0010_B

30.7.2.4 TSRCx — Timer Status Registers Cx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	OCMFC _{x3}	OCMFC _{x2}	OCMFC _{x1}	OCMFC _{x0}	—	—	—	OVFCx	IMFCx3	IMFCx2	IMFCx1	IMFCx0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SB_C-1, corresponding to subblocks C0 to CSB_C-1

Table 30.110 TSRCx Register Contents

Bit Position	Bit Name	Function
15-12	—	These bits are not used. When read, “0” is always returned. When writing, always write 0.
11 to 8	OCMFC _{xy}	Output compare match/input capture flag C _{xy} 1: Input capture or compare match with OCRC _{xy} has occurred. 0: Input capture or compare match with OCRC _{xy} has not occurred. (Default)
7 to 3	—	These bits are not used. When read, “0” is always returned. When writing, always write 0.
4	OVFCx	Overflow flag C _x 1: An overflow has occurred. 0: No overflow has occurred.
3 to 0	IMFC _{xy}	Input capture/compare match flag C _{xy} 1: Input capture and compare match by the GRC _{xy} register have occurred. 0: No input capture or compare match by the GRC _{xy} register has occurred.

Note: x = 0 to SB_C-1. Corresponding to subblocks C0 to CSB_C-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Timer status registers C_x (TSRC_x) are 8-bit/16-bit read-only registers that indicate occurrence of overflow of timer counter (TCNTC_x) in subblocks C_x (x = 0 to SB_C-1), occurrence of input capture or compare match of general registers (GRC_{xy}, y = 0 to 3), and occurrence of output compare match or input capture in output compare/input capture registers (OCRC_{xy}, y = 0 to 3).

These status flags indicate occurrence of an interrupt request. These flags can be cleared by setting corresponding bits in the timer status clear register (TSCRC_x). If an interrupt source occurs with one of these flags set to 1, an interrupt request is generated again. Even when clearing a flag by using the corresponding timer status clear register conflicts with setting the flag due to occurrence of an interrupt source, an interrupt request is generated.

TSRC_x is initialized to 0000_H after reset.

Output Compare Match/Input Capture Flag Cxy (OCMFCxy)

This status flag indicates occurrence of a compare match of the compare register xy (OCRCxy). This flag cannot be set to 1 or 0 by the software. Setting and clearing conditions are described below.

OCMFCxy	Function	
0	[Clearing condition in compare match mode, input capture mode, or one-shot pulse mode] Writing 1 to OCMFCCxy in timer status clear register C (TSCRC).	(Default)
1	[Setting conditions in compare match mode/input capture mode (IOCxy[3:0] is other than 0101 _B , 0110 _B) <ul style="list-style-type: none"> When the TCNTCx value is equal to the OCRCxy value Writing 1 to FOCMCxy (forced output compare match bit) in TCRcx When all except for 00_H is in the established state in Range Comparison Value Setting Register 1Cx (RCR1Cx), and TCNTCx is in the range set in RCR1Cx at the time of writing in OCRCxy. [Setting conditions in input capture mode] <ul style="list-style-type: none"> When the falling edge of TIOCxy is detected when IOCxy[3:0] = 0101_B When the rising edge of TIOCxy is detected when IOCxy[3:0] = 0110_B [Setting conditions in one-shot pulse mode] <ul style="list-style-type: none"> When the TCNTCx value is equal to the OCRCxy value after writing in GRCxy. Writing 1 to FOCMCxy (forced output compare match bit) in TCRcx after writing in GRCxy. When after writing in a GRCxy register in the state as which all except for 00_H is set by Range Comparison Value Setting Register 1Cx (RCR1Cx), TCNTCx writes in OCRCxy, and is in the range set in RCR1Cx. 	

Note: x = 0 to SB_C-1. Corresponding to subblocks C0 to CSB_C-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The next output compare processing is enabled even when the output compare match flag xy (OCMFCxy) is 1 (the flag is not cleared). At this time, 1 is overwritten to OCMFCxy.

Even if the output compare match status is cleared while the TCNTCx and OCMFCxy values are equal after an output compare match is detected, no status is set newly.

The value of TCNTCx, OCRCxy is in the identical state, and even if output comparing match status is cleared, status isn't set newly. Because the rising edge which is at the time of comparing match signal detection is detected and status is being set.

Overflow Flag Cx (OVFCx)

Indicates whether or not TCNTCx has overflowed. This flag cannot be set to 1 or 0 by software.

OVFCx	Function	
0	[Clearing condition] Writing 1 to OVFCx in timer status clear register C (TSCRC)	(Default)
1	[Setting condition] <ul style="list-style-type: none"> When the TCNTCx value overflows (FFFF FFFF_H to 0000 0000_H) When the counter is cleared with the upper-limit setting function enabled 	

Note: x = 0 to SB_C-1. Corresponding to subblocks C0 to CSB_C-1.

The overflow flag is set to 1 when the up-count clock is input with TCNTCx set to FFFF FFFF_H or at the next PCLK (when CLRCSELcx = 0) and at the next count (when CLRCSELcx = 1) clock after the counter value matches the upper-limit setting register value with the upper-limit setting function enabled. When writing 0000 0000_H to TCNTCx or TCNTCx is started from 0000 0000_H, this bit is not set to 1.

When writing to TCNTCx at the same time as incrementation while it is FFFF FFFF_H, this bit is set to 1. However, TCNTCx is started from the written value.

Input Capture/Compare Match Flag Cxy (IMFCxy)

These bits indicate whether or not input capture and compare match between general register (GRCxy) and TCNTCx has occurred. This flag cannot be set to 1 or 0 by software. Setting and clearing conditions are shown below.

IMFCxy	Function
0	[Clearing conditions in compare match mode, input capture mode, or one-shot pulse mode] (Default) Writing 1 to IMFCCxy in timer status clear register C (TSCRC)
1	It's set by the condition besides the TIORCx.IOCxy[1:0] = 00 _B . [Setting condition in input capture mode] (when TIORCx.IOCxy[3:2] = 01 _B or 11 _B) <ul style="list-style-type: none"> ● When the rising edge of TIOCxy is detected when IOCxy[3:0] = 0101_B ● When the falling edge of TIOCxy is detected when IOCxy[3:0] = 0110_B ● When the both edge of TIOCxy is detected when IOCxy[3:0] = 0111_B when edge detection. <ul style="list-style-type: none"> ● When Event1_0 from Timer A is detected when IOCxy[3:0] = 1101_B ● When Event2A from Timer A is detected when IOCxy[3:0] = 1110_B ● When Event2B from Timer A is detected when IOCxy[3:0] = 1111_B [Setting conditions in output compare mode] (when TIORCx.IOCxy[3:2] = 00 _B) <ul style="list-style-type: none"> ● When GRCxy functions as a compare match register and the values in TCNTCx and GRCxy match ● When the forced compare match bit (FCMCxy) in TCRCx is set to 1 ● When all except for 00_H is in the established state in "Range Comparison Value Setting Register 2Cx(RCR2Cx)", and TCNTCx is in the range set in RCR2Cx at the time of writing in GRCxy. [Setting conditions in one-shot pulse output mode] (when TIORCx.IOCxy[3:2] = 10 _B) <ul style="list-style-type: none"> ● When the TCNTCx value is equal to the GRCxy after (or at the same time when) a compare match of OCRCxy is generated while one-shot pulse is output ● When the forced compare match bit (FCMCxy) in TCRCx is set to 1 after (or at the same time when) a compare match of OCRCxy is generated while one-shot pulse is output ● When TCNTCx writes in GRCxy during a one shot pulsed output in the state as which all except for 00_H is set by RCR2Cx, and TCNTCx is in the range set in RCR2Cx.

Note: x = 0 to SB_C-1. Corresponding to subblocks C0 to CSB_C-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Even if these bits are set to 1 meaning that the flag has not been cleared, another input capture or output compare signal can be input. A value of 1 is written to these bits.

Even if the compare match flag is cleared to 0 while TCNTCx and GRCxy after the compare match is detected, these bits are not set to 1.

The value of TCNTCx and GRCxy is in the identical state because rising edge of a comparing match signal is detected and status is being set, and even if input capture comparing match status is cleared, status isn't set newly.

30.7.2.5 TSCRCx — Timer Status Clear Register Cx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	OCMFC Cx3	OCMFC Cx2	OCMFC Cx1	OCMFC Cx0	—	—	—	OVFCC x	IMFCCx 3	IMFCCx 2	IMFCCx 1	IMFCCx 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W	R	R	R	W	W	W	W	W

Note: x = 0 to SB_C-1, corresponding to subblocks C0 to CSB_C-1

Table 30.111 TSCRCx Register Contents

Bit Position	Bit Name	Function
15 to 12	—	These bits are not used. Fix these bits to 0.
11 to 8	OCMFCCxy	Output compare match flag clear Cxy enable 0: Disabled (Default) 1: OCMFCxy in timer status register Cx (TSRCx) is cleared to 0.
7 to 5	—	These bits are not used. Fix these bits to 0.
4	OVFCCx	Overflow flag clear Cx enable 0: Disabled (Default) 1: OVFCx in timer status register C (TSRCx) is cleared to 0.
3 to 0	IMFCCxy	Input capture flag clear Cxy enable 0: Disabled (Default) 1: IMFCxy in timer status register C (TSRCx) is cleared to 0.

Note: x = 0 to SB_C-1, corresponding to subblocks C0 to CSB_C-1. y = 0, 1, 2, 3, corresponding to channels 0 to 3.

TSCRCx is an 8-bit/16-bit readable/writable register to set clearing of flags at occurrence of an overflow or input capture or compare match. This register is always read as 0.

TSCRCx is initialized to 0000_H after reset.

Output Compare Match Flag Clear Cxy enable (OCMFCCxy)

Writing 1 to this register while the compare match flag (OCMFCxy) in the timer status register (TSRCx) is set to 1 clears OCMFCCxy to 0. This bit is always read as 0.

Overflow Flag Clear C enable (OVFCCx)

Writing 1 to this bit while overflow flag C (OVFCx) in the timer status register C (TSRCx) is set to 1 clears OVFCx to 0. This bit is always read as 0.

Input Capture Flag Clear Cxy Enable (IMFCCxy)

Writing 1 to this bit while input capture flag Cxy (IMFCxy) in the timer status register x (TSRCx) is set to 1 clears IMFCCxy to 0. This bit is always read as 0.

30.7.2.6 TIORCx — Timer I/O Control Registers Cx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IOCx3[3:0]				IOCx2[3:0]				IOCx1[3:0]				IOCx0[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_C-1, corresponding to subblocks C0 to CSB_C-1

Table 30.112 TIORCx Register Contents

Bit Position	Bit Name	Function
15 to 12,	IOCxy[3:0]	Settings of timer general register (GRCxy) and output compare registers (OCRCxy) functions
11 to 8,		Settings of input capture and compare match
7 to 4, 3 to 0		Settings of edge to be extracted at input capture Settings of output value at compare match

Note: x = 0 to SB_C-1, corresponding to subblocks C0 to CSB_C-1. y = 0, 1, 2, 3, corresponding to channels 0 to 3.

I/O Control Bit (IOCxy[3:0])

No.	mode	IOCxy				TCRCx	OCRCx	GRCx	Terminal TIOC	Function		
		[3]	[2]	[1]	[0]							
1	Compare match mode	0	0	0	0	0	Cmp ^{*5}	x	state retention	[When PWM = 1] GRCx0 is used as a cycle register and GRCx1-3 is used as a duty register. [When PWM = 0] The setting level is output from a TIOC in case of TCNTCx = GRCxy.		
2				0	1						0	
3												1
4				1	0						0	
5												1
6				1	1						0	
7	Input capture mode ^{*3}	0	1	0	0	0	Cmp ^{*5}	x	* It's used as an input terminal. * An output factor is state retention.	The capture function, being effective and when detecting an edge of a chosen TIOC, TCNTCx is captured. When a comparing match of OCRC is effective, it's compared with TCNTCx.		
8				0	1						Cap (TIOC↓)	Cap (TIOC↑)
9				1	0						Cap (TIOC↑)	Cap (TIOC↓)
10				1	1						Cmp	Cap (TIOC↑↓)
11	One-shot pulse mode	1	0	0	0	0	Cmp	Cmp	0 output (active high)	It's used for initialization of an output terminal. But, the output of a one shot pulse is impossible. A TIOC does state maintenance of a forced compare match until it's established. The level established in IOCxy is output from a TIOC after forced compare match setting. ^{*1}		
12				0	1						1 output (active low)	
13				1	0						One shot output (active high)	0 is output from TIOCxy output until compare match occurrence at the OCRCxy register. 1 is output from TIOCxy output from compare match occurrence at the OCRCxy register to compare match occurrence at the GRCxy register. 0 is output from TIOCxy output by compare match occurrence at the GRCxy register. ^{*2}
14				1	1						One shot output (active low)	1 is output from TIOCxy output until compare match occurrence at the OCRCxy register. 0 is output from TIOCxy output from compare match occurrence at the OCRCxy register to compare match occurrence at the GRCxy register. 1 is output from TIOCxy output by compare match occurrence at the GRCxy register. ^{*2}
15	Input capture mode	1	1	0	0	0	Cmp ^{*5}	x	* It's used as an input terminal. * An output factor is state retention.	The capture function, being effective and when detecting chosen Event, TCNTCx is captured. When a compare match of OCRC is effective, it's compared with TCNTCx.		
16				0	1						Cap (Event1)	
17				1	0						Cap (Event2A)	
18				1	1						Cap (Event2B)	

Note: x = 0 to SB_C-1, y = 0, 1, 2, 3.

Cmp: It can be used as a compare match register.^{*4}

Cap: It can be used as a capture register. () indicates a capture trigger.

x: don't use

- Note 1. If a compare match by OCRCxy and a compare match by GRCxy occur at the same time, the compare match by GRCxy takes precedence.
- Note 2. Before selecting one-shot pulse mode (IOCxy[3:2] = 10_B), set the PWMX0 bit to 0.
It is prohibited to set PWMX0 bit to 1.
(The circuit operates in PWM mode.)
- Note 3. Output operation of a one shot pulse is permitted by write to GRCxy. Set OCRCxy before writing in to GRCxy. The value of TCNTCx and OCRCxy is to be identical, and the active level is output. Set OCRCxy, GRCxy so that an OCRCxy compare match may occur or occur at the same time first than a GRCxy compare match after GRCxy writing in at the time of one shot pulsed output movement.
- Note 4. A range compare match by RCR1Cx, and RCR2Cx and a range compare match by FCMCx and FOCMCx are also included.
- Note 5. A compare match of TCNTCx and OCRC doesn't influence output of a TIOC by a mode except for a one shot pulse mode. When interrupt occurred to the time of a comparing match, that notifies outside of a comparing match's occurring.

The IOCxy[3:2] bits are used to set operating mode (compare match mode, input capture mode, or one-shot pulse mode). Change operating mode (IOCxy[3:2]) while the counter is not working.

Disable compare match and input capture (by setting IOCxy[1:0] to 00_B) before switching output mode for compare match mode (IOCxy[3:2] = 00_B) or switching edge detection/trigger for input capture mode (IOCxy[3:2] = 01_B or 11_B).

Change operating mode to one-shot pulse mode (IOCxy[3:2] = 10_B) and change active polarity (IOCxy[1:0]) in one-shot pulse mode (IOCxy[3:2] = 10_B) while the counter is not working.

When one-shot pulse mode (IOCxy[3:2] = 10_B) is used, initialize the external pins (TIOCxy) by forced compare match Cxy (FCMCxy) after operating mode is set, and then start counting.

When IOCxy[3:2] = x1_B and the mode setting is for input-capture, output operations do not proceed (the existing level on TIOCxy is retained).

The condition of the register who can establish it is different in each mode, so confirm each register condition to refer to the following table and establish it by TIORCx.IOCxy and TCRCx.PWMx0.

No.	mode	IOCxy				TCRCx	TCRCx	CLRCSELx	CLRCx	OCRCx	GRCx	CUCR	RCR1C	RCR2C	The upper limit value of TCNTCx									
		[3]	[2]	[1]	[0]																			
1	Compare match mode	0	0	0	0	0	0/1	0/1	Cmp	x	o	o	x	[When PWMx0 = 1] value of GRCx0										
2				0	1										0	o								
3				1	0										0		x	x	x	[When PWMx = 0, CLRCx = 0] FFFF_FFFF _H				
4				1	0										0						o	o	o	[When PWMx0 = 0, CLRCx = 1] value of CUCR*1
5				1	0										0									
6				1	1										0									
7	Input capture mode	0	1	0	0	0	0/1	0/1	Cmp	x	o	o	x	[When PWMx0 = 0, CLRCx = 0] FFFF_FFFF _H										
8				0	1										Cap	Cap								
9				1	0												Cmp	o	o	o				
10				1	1																			
11	One-shot pulse mode	1	0	0	0	0	0/1	0/1	x	x	o	o	o											
12				0	1																			
13				1	0									Cmp	Cmp									
14				1	1																			
15	Input capture mode	1	1	0	0	0	0/1	0/1	Cmp	x	o	o	x											
16				0	1									Cap										
17				1	0																			
18				1	1																			

Note: Cmp: It can be used as a compare match Cap: It can be used as a capture

o: Setting possible, x: Setting prohibited, 1: Setting possible, 0: Setting possible

Note 1. When the writing in value to CUCRC is smaller than TCNTCx, it'll be FFFF_FFFF_H.

30.7.2.7 TCNTCx — Timer Counters Cx

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTCx[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTCx[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_C-1, corresponding to subblocks C0 to CSB_C-1)

Timer counter C_x (TCNTCx) is 32-bit readable/writable registers driven by the input clock. These counters can be read from and written to while they are being run.

Timer counter C_x (TCNTCx) is started for counting by setting the bit in timer start register (TSTRC) to 1. The clock signal is selected by the clock select bits (CKSEL) in timer control register (TCRCx). A timer overflow causes an overflow interrupt request to be output to the CPU. TCNTCx is initialized to 0000 0000_H after reset.

The upper limit value of timer counter C_x (TCNTCx)

PWMx0	CLRCx	Upper limit value of timer counter C _x (TCNTCx)
1	0/1	GRCx0
0	0	FFFF_FFFF _H
0	1	CUCRCx*1

Note 1. When the writing in value to CUCRC is smaller than TCNTCx, it'll be FFFF_FFFF_H.

30.7.2.8 GRCxy — Timer General Registers Cxy

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GRCxy[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GRCxy[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_C-1: Corresponding to subblocks C0 to CSB_C-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

These timer general registers Cxy (GRCxy) are 32-bit readable/writable registers that function as the input capture register or output compare register. These functions are switched by setting timer I/O control register Cx (TIORCx).

- Operation in input capture mode

When GRCxy is used as an input capture register, it detects an external input capture signal, stores the TCNTCx value, and outputs an input capture interrupt request to the CPU. At this time, the IMFCxy bit in timer status register x (TSRCx) is set to 1. The edge to be detected is selected by the corresponding TIORCx.

Input capture is performed even if the counter is stopped (the TCE bit in ATUENR is cleared to 0 or the STRCx bit in TSTRC is cleared to 0). The value in the counter stopped is loaded to GRCxy.

- Operation in compare match mode

When GRCxy is used as an output compare register, the values in GRCxy and the timer counter (TCNTCx) are constantly compared. When these values match, TIOCx output is changed according to the method (0 output, 1 output, or toggle output) specified in TIORCx, a compare match interrupt request is output to the CPU. At this time, the IMFCxy bit in the timer status register (TSRCx) is set to 1. Initially, logical zero is output on the TIOCx pin (immediately after a reset or output compare modes are switched). During operation, the previous value is output. If these registers function as the output compare register for one-shot pulse output, one-shot pulse output starts by writing to GRCxy. Set OCRCxy before writing to GRCxy.

TCNTCx value is captured in GRCx at the time of range compare match occurrence by RCR2Cx. Regarding the range compare function, refer to **Section 30.7.3.6 Range Compare Function**.

- Operation in one-shot pulse mode

When GRCxy is used as an output compare register for one-shot pulse output, one-shot pulse output starts when a value is written to GRCxy. Be sure to set OCRCxy before writing to GRCxy.

The values in GRCxy and timer counter (TCNTCx) are compared after a compare match with OCRCxy is detected (or when GRCxy = OCRCxy).

Compare match of GRCxy inactivates one-shot pulse (compare match with OCRCxy activates one-shot pulse) and the IMFC bit in the timer status register (TSRCx) is set to 1. TCNTCx value is captured in GRCx at the time of range compare match occurrence by RCR2Cx. Regarding the range compare function, refer to **Section 30.7.3.6, Range Compare Function**.

Once the one-shot pulse is output, another output is not performed until new value is set to GRCxy. Also, the IMFC bit in the timer status register (TSRCx) is not changed.

When these registers are used as output compare registers for compare match or one-shot pulse output, if CLRSEL_{Cx} bit is set to 0 and the counter upper-limit setting function (see **Section 30.7.3.5, Counter Upper-Limit Setting Function**) is used, and moreover either clock-bus line 5 or a prescaler division ratio as 1/1 is selected for the count clock, the timer counter counts between 1 and the value of CUCRC_x and the counter value may not become 0000 0000_H. Therefore, do not set these registers to 0000 0000_H in such cases.

GRC_{xy} is initialized to FFFF FFFF_H after reset.

30.7.2.9 NCNTCxy — Noise Canceler Counters Cxy

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NCNTCxy[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.113 NCNTCxy Register Contents

Bit Position	Bit Name	Function
15 to 0	NCNTCxy	Noise cancel count Cxy These bits store an 16-bit count value

Note: x = 0 to SB_C-1. Corresponding to subblocks C0 to CSB_C-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Noise Canceler counters Cxy (NCNTCx0 to x3) are 16-bit readable/writable registers. NCNTCx0 to x3 are started by the external input pins (TIOCx0 to TIOCx3) as triggers when the noise canceler is enabled by the noise cancel enable bit (NCECx0 to x3) in timer I/O control register Cx (TIORCx). In level accumulation cancellation mode, these counters increment or decrement according to the external input level. Up-count or down-count after operation start is performed in synchronization with the noise canceler count clock or clock bus 5 supplied from the prescaler.

NCNTCxy continues counting regardless of the settings in the timer C enable bit (TCE) in ATU master enable register (ATUENR) and in TSTRC (regardless of the TCNTCx).

NCNTCxy operates in one of three modes (premature-transition cancellation mode, minimum time-at-level cancellation mode, or level accumulation cancellation mode) according to the settings of the noise cancellation mode select bit (NCMSEL) and the timer C noise cancellation mode bit (NCMC) in the noise cancellation mode register (NCMR) in the common control unit.

- In premature-transition cancellation mode

When a level change of the externally input signal (TIOCx_y) is detected while bit NCECx_y is set to 1 and NCNTCxy is stopped, NCNTCxy is started for counting up. This counter is cleared to 0000_H and stopped on the first edge of the PCLK after the value in NCNTCxy matches the value in noise cancel register (NCRCxy).

NCNTCxy is incremented regardless of the timer C enable bit (TCE) in ATU master enable register (ATUENR).

The first change is output as the signal whose noise is removed and the edge is to be extracted. Subsequent level changes are masked until the value in the counter reaches the value in the noise cancel register (NCRCxy). The level of the externally input signal is output on this compare match.

When the NCECx_y bit is cleared to 0 while the counter (NCRCxy) is being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes are also masked over this period.

- In minimum time-at-level cancellation mode

When a level change of the externally input signal (TIOCxy) is detected while the NCECxy bit is set to 1 and NCNTxy is stopped, NCNTxy is started for counting up. This counter is cleared to 0000_H and stopped on the first edge of the PCLK after the value in NCNTxy matches the value in noise cancel registers (NCRCxy) or after the level of the externally input signal (TIOCxy) is changed.

NCNTxy is incremented regardless of the timer C enable bit (TCE) in ATU master enable register (ATUENR).

When a level change of the externally input signal is detected, the corresponding noise canceler counter (NCNTxy) is started for counting up. If subsequent level changes are not detected until the value in the counter reaches the value in the noise canceler register (NCRCxy), the previously accepted level change is output as the signal whose noise is removed on compare match of the counter and noise canceler register. When the subsequent level change is detected before the values in the counter and noise canceler register match, all the changes is treated as noise. Therefore the signal whose noise is removed is not changed.

When the NCECxy bits are cleared to 0 while the counter (NCNTCxy) is being incremented, counting continues until the values in the counter and the noise canceler register match or a level change on the externally input signal is detected.

- In level accumulation cancellation mode

When the NCECxy bit is 1, NCNTCxy increments or decrements according to the input signal level. While the input level is high, NCNTCxy increments. When the counter value matches the noise cancel register (NCRCxy) value, up-counting stops on the next PCLK. While the input level is low, NCNTCxy decrements. When the counter value reaches 0000_H, down-counting stops on the next PCLK.

NCNTCxy counts regardless of the timer C enable bit (TCE) value in the ATU master enable register (ATUENR).

When the counter value matches the NCRCxy value during up-counting, the noise canceler output is updated to 1.

When the counter value reaches 0000_H during down-counting, the noise canceler output is updated to 0.

When the NCECxy bit is cleared during counting, the noise canceler counter stops working and the value changes from the noise canceler output to the input signal level at that time. For this reason, note that when clearing the NCECxy bit in level accumulation cancellation mode, edge detection may be made at this changeover.

NCNTCxy is initialized to 0000_H after reset.

30.7.2.10 NCRCxy — Noise Canceler Registers Cxy

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NCRCxy[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.114 NCRCxy Register Contents

Bit Position	Bit Name	Function
15 to 0	NCRCxy	Noise Canceling Time Cxy TIOcxy noise canceling period (16-bit comparison value)

Note: x = 0 to SB_C-1. Corresponding to subblocks C0 to CSB_C-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Noise cancel registers Cxy (NCSCx0 to Cx3) are 16-bit readable/writable registers that are provided in each subblock and set the upper limitations of noise cancel counters Cxy (NCNTCx0 to Cx3).

When PCLK/128 clock is selected as a noise canceler clock, noise with maximum length of 0.21 s (at PCLK = 40 MHz) can be canceled when this register is set to FFFF_H.

NCSCx0 to NCSCx3 operate in one of three modes (premature-transition cancellation mode, minimum time-at-level cancellation mode, or level accumulation cancellation mode) according to the settings of the noise cancellation mode select bit (NCMSEL) and the timer C noise cancellation mode bit (NCMC) in the noise cancellation mode register (NCRM) in the common control unit.

- In premature-transition cancellation mode
While NCNTCxxy is in count operation, the level change of the subsequent input signal is masked. Values in NCNTCxxy and NCRCxy are always compared. If a compare match occurs, the value in NCNTCxxy is cleared on the next PCLK, the count operation is stopped, and the masking of the input signal is canceled.
- In minimum time-at-level cancellation mode
While NCNTCxxy is in count operation, noise canceler processing waiting state is entered. Values in NCNTCxxy and NCRCxy are always compared. If a compare match occurs, the value in NCNTCxxy is cleared on the next PCLK, the count operation is stopped, and at the same time the noise canceler outputs the input signal that has passed through the noise canceling processing.
- In level accumulation cancellation mode
While NCNTCxxy is counting up, the NCNTCxxy value is compared with the NCRCxy value. When a compare match occurs, NCNTCxxy stops up-counting on the next PCLK. While NCNTCxxy is counting down, the NCNTCxxy value is compared with 0000_H.

NCRCx0 to NCRCx3 can be read and written only in 16-bit units.

NCRCx0 to NCRCx3 are initialized to 0000_H after reset.

30.7.2.11 OCRCxy — Output Compare Registers Cxy

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OCRCxy[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCRCxy[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_C-1: Corresponding to subblocks C0 to CSB_C-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Output compare registers Cxy (OCRCxy) are 32-bit readable/writable registers that function as an output compare register. These registers are used as a compare register in modes other than one-shot pulse mode.

- Operation in input capture mode:

When GRCxy is used as an input capture register, it detects an external input capture signal, stores the TCNTCx value, and outputs an input capture interrupt request to the CPU. At this time, the IMFCxy bit in timer status register x (TSRCx) is set to 1. The edge to be detected is selected by the corresponding TIORCx.

Input capture is performed even if the counter is stopped (the TCE bit in ATUENR is cleared to 0 or the STRCx bit in TSTRC is cleared to 0). The value in the counter stopped is loaded to GRCxy.

- Operation in compare match mode:

The values in OCRCxy and timer counter (TCNTCx) are always compared. When these values match, a compare match interrupt request is outputs to the CPU. At this time, the OCMFC bit in the timer status register (TSRCx) is set to 1. TCNTCx value is captured in OCRCx at the time of range compare match occurrence by RCR1Cx. Regarding the range compare function, refer to **Section 30.7.3.6, Range Compare Function**.

- Operation in one-shot pulse mode:

When GRCxy is used as an output compare register for one-shot pulse output, one-shot pulse output starts when a value is written to GRCxy. Be sure to set OCRCxy before writing to GRCxy.

When one-shot pulse output starts, the values in OCRCxy and timer counter (TCNTCx) are compared.

Compare match with OCRCxy activates one-shot pulse (compare match with GRCxy inactivates one-shot pulse) and the OCMFCxy bit in the timer status register (TSRCx) is set to 1. TCNTCx value is captured in OCRCx at the time of range compare match occurrence by RCR1Cx. Regarding the range compare function, refer to **Section 30.7.3.6, Range Compare Function**.

Once the one-shot pulse is output, another output is not performed until new value is set to GRCxy. Also, the OCMFCxy bit in the timer status register (TSRCx) is not changed. When these registers are used as output compare registers for compare match or one-shot pulse output, if the counter upper-limit setting function (see **Section 30.7.3.5, Counter Upper-Limit Setting Function**) is in use with clock-bus line 5, or with a division ratio of 1/1 selected for the prescaler, the timer counter counts between 1 and the value of CUCRCx and the counter value never reaches 0000 0000_H. Therefore, do not set these registers to 0000 0000_H in such cases.

OCRCxy is initialized to FFFF FFFF_H after reset.

30.7.2.12 OCMRCxy — Output Compare Mirror Registers Cxy

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OCMRCxy[31:16]																
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCMRCxy[15:0]																
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SBc-1: Corresponding to subblocks C0 to CSBc-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The output compare mirror registers Cxy (OCMRCxy) are 32-bit read-only registers that have a function mirroring the output compare registers Cxy (OCRCxy) for DMA transfer. Writing to this register is prohibited.

The mirroring function is provided to DMA-transfer the value captured to OCRCxy or GRCxy in input capture mode. During DMA transfer, read the values in OCMRCxy and GMRCxy consecutively in this order.

30.7.2.13 GMRCxy — Timer General Mirror Registers Cxy

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GMRCxy[31:16]																
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GMRCxy[15:0]																
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SB_C-1: Corresponding to subblocks C0 to CSB_C-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The timer general mirror registers Cxy (GMRCxy) are 32-bit read-only registers that have a function mirroring the timer general registers Cxy (GRCxy) for DMA transfer. Writing to this register is prohibited.

The mirroring function is provided to DMA-transfer the value captured to OCRCxy or GRCxy in input capture mode. During DMA transfer, read the values in OCMRCxy and GMRCxy consecutively in this order.

30.7.2.14 TIERCx — Timer Interrupt Enable Registers Cx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	OCRCE _{x3}	OCRCE _{x2}	OCRCE _{x1}	OCRCE _{x0}	—	—	—	OVCE _x	GRCE _{x3}	GRCE _{x2}	GRCE _{x1}	GRCE _{x0}
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_C-1: Corresponding to subblocks C0 to CSB_C-1

Table 30.115 TIERCx Register Contents

Bit Position	Bit Name	Function
15 to 12	—	These bits are not used. Fix these bits to 0.
11 to 8	OCRCE _{xy}	OCRC interrupt enable xy Interrupt request due to comparison and capture by OCRCE _{xy} 0: Interrupt request due to comparison and capture by OCRCE _{xy} is disabled. 1: Interrupt request due to comparison and capture by OCRCE _{xy} is enabled.
7 to 5	—	These bits are not used. Fix these bits to 0.
4	OVCE _x	Overflow interrupt enable x 0: Interrupt request due to overflow is disabled. 1: Interrupt request due to overflow is enabled.
3 to 0	GRCE _{xy}	GRC interrupt enable xy 0: Interrupt request due to comparison and capture by GRCE _{xy} is disabled. 1: Interrupt request due to comparison and capture by GRCE _{xy} is enabled.

Note: x = 0 to SB_C-1. Corresponding to subblocks C0 to CSB_C-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Timer interrupt enable registers C_x (TIERCx) are 8-bit/16-bit readable/writable registers used to set interrupt requests due to comparison in subblocks C0 to CSB_C-1.

Interrupt request C_{xy} due to comparison (OCRCE_{xy})

An interrupt is generated on compare match by the output compare register (OCRC_{xy}).

Interrupt request C_{xy} due to overflow (OVCE_x)

An interrupt is generated upon occurrence of an overflow.

Interrupt request C_{xy} due to comparison and capture (GRCE_{xy})

An interrupt is generated on compare match or capture operation by the general register GRC_{xy}.

30.7.2.15 CUCRCx — Counter Upper-Limit Setting Compare Registers Cx

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CUCRCx[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CUCRCx[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_C-1: Corresponding to subblocks C0 to CSB_C-1

Counter upper-limit value compare registers C_x (CUCRC_x) are 32-bit readable/writable registers that has a function to be compared with the TCNTC_x counter. Compare match between this register and the TCNTC_x counter is enabled by setting the TCNTC_x clear bit C_x (CLRC_x) in the timer control register (TCRC_x) to 1. The TCNTC_x counter is cleared to 0000 0000_H on compare match between the TCNTC_x counter and counter upper-limit setting compare register (CUCRC_x).

If this compare match occurs at the same time as counting up by the timer counter C_x (TCNTC_x), the timer counter C_x (TCNTC_x) is cleared to 0000 0001_H.

The OVFC_x bit in timer status register C (TSRC_x) is set to 1 and an overflow interrupt is output on compare match between the TCNTC_x counter and counter upper-limit setting compare register C_x (CUCRC_x).

Establish “CUCRC_x ≥ GRC_{xy}” by occurrence condition of a compare match of GRC_{xy} by Counter Upper-Limit Setting Compare Registers C_x(CUCRC_x) when GRC_{xy} is used as a comparing register at CLRC_x = 1.

When not establishing it, an expectancy wave-shaped isn't output from an output terminal TIOC.

Do not set CUCRC_x to 0000 0000_H. If 0000 0000_H is set, a compare match may occur at an unintended timing.

CAUTION

The upper limit value setting function can be enabled (CLRC_x = 1) in other than PWM mode (PWM_{x0} = 0).

30.7.2.16 RCR1Cx — Range Comparison Value Setting Register 1Cx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	RCR1Cx[4:0]				
Value after reset	0*1	0*1	0*1	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_C-1: Corresponding to subblocks C0 to CSB_C-1

Note 1. These bits are not used. Fix these bits to "0".

The range comparison value setting registers 1Cx (RCR1Cx) are 8-bit readable/writable registers.

This register can enable / disable range compare function and set range value. Regarding the range compare function, refer to **Section 30.7.3.6, Range Compare Function**.

RCR1Cx is initialized to 00_H after reset.

30.7.2.17 RCR2Cx — Range Comparison Value Setting Register 2Cx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	RCR2Cx[4:0]				
Value after reset	0*1	0*1	0*1	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_C-1: Corresponding to subblocks C0 to CSB_C-1

Note 1. These bits are not used. Fix these bits to "0".

The range comparison value setting registers 2Cx (RCR2Cx) are 8-bit readable/writable registers. This register can enable / disable range compare function and set range value. Regarding the range compare function, refer to **Section 30.7.3.6, Range Compare Function**.

RCR2Cx is initialized to 00_H after reset.

The following table shows ranges that can be set as range comparison values (power of 2 (excluded only when RCR1Cx and RCR2Cx are 0)).

RCR1Cx[4:0], RCR2Cx[4:0]	Selectable Range	Remarks
00000	None	Range comparison is invalid (Default)
00001	2^1-1	—
00010	2^2-1	—
00011	2^3-1	—
00100	2^4-1	—
00101	2^5-1	—
00110	2^6-1	—
00111	2^7-1	—
01000	2^8-1	—
01001	2^9-1	—
01010	$2^{10}-1$	—
01011	$2^{11}-1$	—
01100	$2^{12}-1$	—
01101	$2^{13}-1$	—
01110	$2^{14}-1$	—
01111	$2^{15}-1$	—
10000	$2^{16}-1$	—
10001	$2^{17}-1$	—
10010	$2^{18}-1$	—
10011	$2^{19}-1$	—
10100	$2^{20}-1$	—
10101	$2^{21}-1$	—
10110	$2^{22}-1$	—
10111	$2^{23}-1$	—
11000	$2^{24}-1$	—
11001	$2^{25}-1$	—
11010	$2^{26}-1$	—
11011	$2^{27}-1$	—
11100	$2^{28}-1$	—
11101	$2^{29}-1$	—
11110	$2^{30}-1$	—
11111	$2^{31}-1$	—

30.7.2.18 NCMCR1C — Noise Cancellation Mode Channel Register 1C

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	NCM1C 10	NCM1C 9	NCM1C 8	NCM1C 7	NCM1C 6	NCM1C 5	NCM1C 4	NCM1C 3	NCM1C 2	NCM1C 1	NCM1C 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.116 NCMCR1C Register Contents

Bit Position	Bit Name	Function
15 to SB _c	—	These bits are not used. Fix these bits to “0”.
SB _c -1 to 0	NCM1C _x	Subblock x all channels noise cancellation mode This bit specifies the operating mode of all the noise cancellers in subblock x. 0: Premature-transition cancellation mode 1: Minimum time-at-level cancellation mode (When NCMC = 0 and NCM2C _x = 0) 1: Level accumulation cancellation mode (When NCMC = 0 and NCM2C _x = 1)

Note: x = 0 to SB_c-1. Corresponding to subblocks C0 to CSB_c-1.

The noise cancellation mode channel register 1C is an 8-bit or a 16-bit readable/writable register used to select an operating mode of noise canceler in each channel. However, the setting of this register applies to all channels in the corresponding subblock and the setting cannot be made for the individual channels.

Premature-transition cancellation mode detects a change in input signal level, and then ignores changes in input signal level within the specified time period. Changes in input signal level within the specified time period after the first level change are regarded as noise in this mode.

Level accumulation cancellation mode accumulates input signal levels and regards that the input level has reached 0 or 1 when the accumulation result becomes 0 or the specified value.

Each period is specified by the noise cancellation register in each channel, and the noise cancellation counter measures the time.

Subblock x All Channels Noise Cancellation Mode (NCM1C_x)

This bit specifies the operating mode of all the noise cancellers in subblock x.

CAUTION

This register is only effective when the NCMC bit of the noise cancellation mode register (NCMR) in the common control unit is “0”. Furthermore, when the corresponding channel bit (NCM1C_x) in this register is set to “1”, the mode can be set to premature-transition cancellation mode, minimum time-at-level cancellation mode, or level accumulation cancellation mode according to the state of the corresponding channel bit in the noise cancellation mode channel register 2C (NCMCR2C).

Note: x indicates 0 to SB_c-1.

30.7.2.19 NCMCR2C — Noise Cancellation Mode Channel Register 2C

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	NCM2C ₁₀	NCM2C ₉	NCM2C ₈	NCM2C ₇	NCM2C ₆	NCM2C ₅	NCM2C ₄	NCM2C ₃	NCM2C ₂	NCM2C ₁	NCM2C ₀
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.117 NCMCR2C Register Contents

Bit Position	Bit Name	Function
15 to SB _C	—	These bits are not used. Fix these bits to “0”.
SB _C -1 to 0	NCM2C _x	Subblock x all channels noise cancellation mode This bit specifies the operating mode of all the noise cancellers in subblock x. 0: Minimum time-at-level cancellation mode (When NCMC = 0 and NCM1C _x = 1) 1: Level accumulation cancellation mode (When NCMC = 0 and NCM1C _x = 1)

Note: x = 0 to SB_C-1. Corresponding to subblocks C0 to CSB_C-1.

The noise cancellation mode channel register 2C is an 8-bit or a 16-bit readable/writable register used to select an operating mode of noise canceler in each channel. However, the setting of this register applies to all channels in the corresponding subblock and the setting cannot be made for the individual channels.

Level accumulation cancellation mode accumulates input signal levels and regards that the input level has reached 0 or 1 when the accumulation result becomes 0 or the specified value.

Each period is specified by the noise cancellation register in each channel, and the noise cancellation counter measures the time.

Subblock x All Channels Noise Cancellation Mode 2 (NCM2C_x)

This bit specifies the operating mode of all the noise cancellers in subblock x.

CAUTION

This register is only effective when the NCMC bit of the noise cancellation mode register (NCMR) in the common control unit is “0” and the corresponding channel bit (NCM1C_x) in noise cancellation mode channel register 1C (NCMCR1C) is “1”.

Note: x indicates 0 to SB_C-1.

30.7.3 Details of Operation

30.7.3.1 Input Capture Mode (Input Capture Function)

Timer general registers Cxy (GRCxy) and output compare registers Cxy (OCRCxy) perform input capture when an edge is input from the corresponding external pin (TIOCxy) or an event output (event 1, 2A, or 2B) from timer A is input by setting input capture in timer I/O control registers (TIORCx).

The timer counter (TCNTCx) starts up-counting by setting the timer start register (TSTRx). When an edge of the corresponding external pin of GRCxy and OCRCxy is input or an event output from timer A used as a capture timing signal of GRCx is input, the corresponding bit (IMFCxy/OCMFCxy) in the timer status register (TSRC) is set to 1 and the counter value is transferred to GRCxy and OCRCxy. After a changed edge of TIOCxy has been extracted and then two internal operating clock (PCLK) cycles have passed, the interrupt output changes. Input capture flags (IMFCxy/OCMFCxy bits in TSRCx) can be cleared by writing 1 to the corresponding clear bit in the timer status clear register (TSCRCx).

An edge to be input can be selected from rising edge, falling edge, and both edges according to the setting of the IOC bit in the TIORCx register.

Furthermore, an interrupt request can be output to the CPU at the input capture timing.

The value in timer counter Cx (TCNTCx) is cleared when its value matches the value of counter upper limit value setting compare register Cx (CUCRCx) while the CLRCx bit in timer control register Cx (TCRCx) is set to enabled. But, when the writing in value to CUCRC is smaller than TCNTCx, it'll be FFFF_FFFF_H.

Figure 30.38 shows an operation example of input capture in subblock C0. TIOC00 is captured at active high, and TIOC01 is captured at active low.

Input capture is performed even if TCNTCx is stopped (when the TCE bit in ATUENR or bit STRCx is cleared to 0). The counter value of the TCNTCx stopped is captured in GRCxy and OCRCxy.

TIORC.IOCxy				TCRCx	TCRC.	TCRC.	OCRCxy	GRCxy	CUCRCx	RCR1Cx	RCR2Cx	The upper limit value of TCNTCx
[3]	[2]	[1]	[0]	PWMx0	CLRCx	CLRCSELCx						
0	1	0	0	0 *(1 setting prohibited)	0/1		Cmp	x	0/1	o	x	[When TCRC.CLRCx = 0] FFFF_FFFF _H
		0	1				Cap(TIOC↓)	Cap(TIOC↑)				
		1	0				Cap(TIOC↑)	Cap(TIOC↓)				
		1	1				Cmp	Cap(TIOC↑↓)				
1	1	0	0				Cmp	x		o		[When TCRC.CLRCx = 1] CUCRCx or FFFF_FFFF _H
		0	1				Cap	(Event Output 1)				
		1	0				Cap	(Event Output 2A)				
		1	1				Cap	(Event Output 2B)				

Note: Cmp: It can be used as a compare match.

Cap: It can be used as a capture. () indicates a capture trigger.

o: Setting possible, x: Setting prohibited

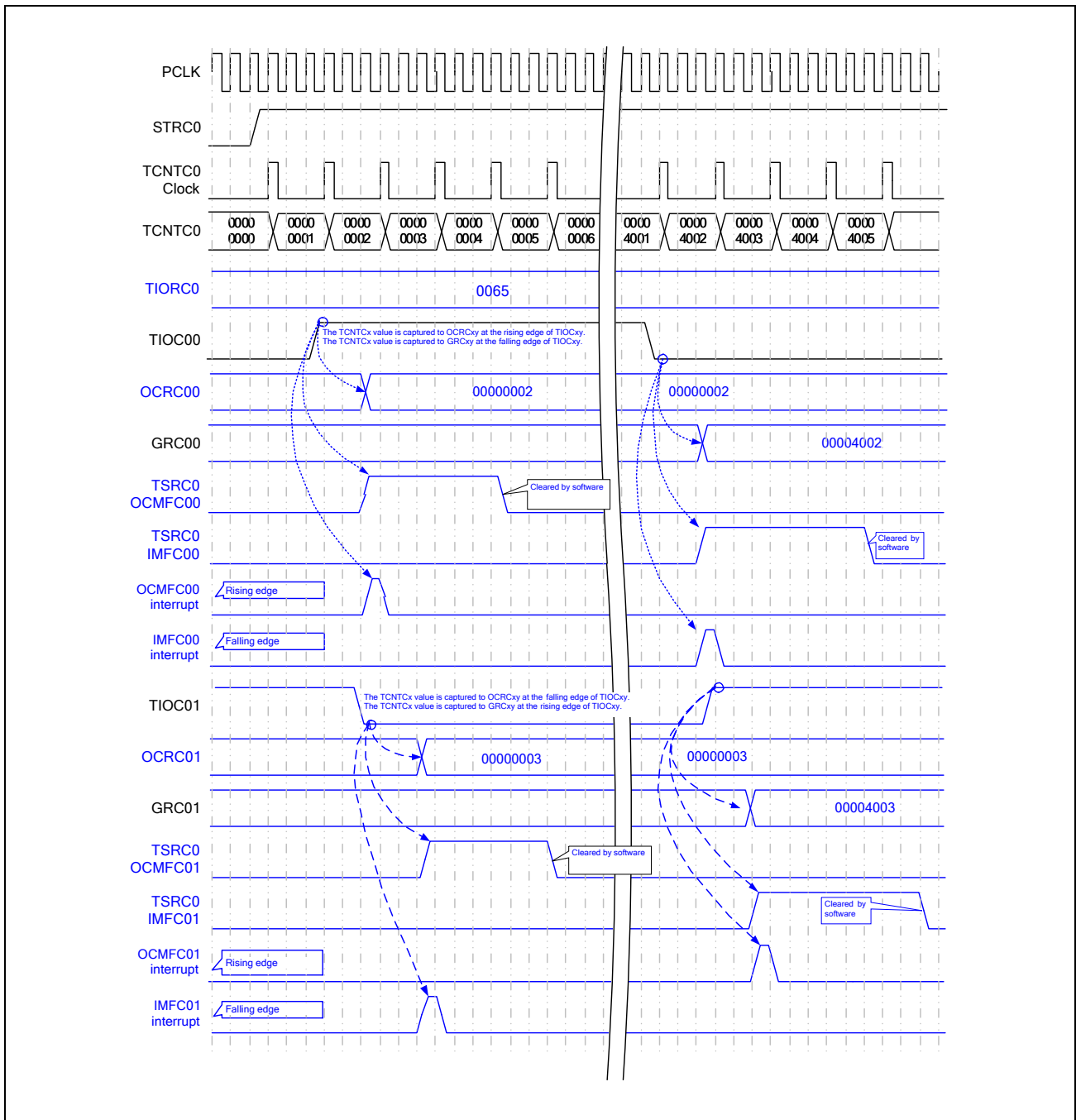


Figure 30.38 Operation Example of Input Capture

30.7.3.2 Compare Match Mode (Compare Match Function)

The following table can be set as a compare match mode. The PWM function explains the explanation at the time of on/off below.

TIORC.IOCxy				TCRCx.	TCRC.	TCRC.	OCRCxy	GRCxy	CUCRCx	RCR1Cx	RCR2Cx	The upper limit value of TCNTCx	
[3]	[2]	[1]	[0]	PWMx0	CLRCx	CLRCSELCx							
0	1	0	0	0	0/1		Cmp	x	o	o	x	[When TCRC.PWMx0 = 1] GRCx0	
				0	1	0							0/1
				1	0 *(1 setting prohibited)								
		1	0	0	0/1								
				1	0 *(1 setting prohibited)								
				1	1	0							0/1
								[When TCRC.CLRCx = 0] FFFF_FFFF _H		[When TCRC.CLRCx = 1] CUCRCx or FFFF_FFFF _H			

Note: Cmp: It can be used as a compare match.

o: Setting possible, x: Setting prohibited

General registers (GRCxy) of timer C output compare match from the corresponding external pin (TIOCxy) by setting compare match operation in timer I/O control registers (TIORCx). A comparing match by Output Compare Registers Cxy (OCRCxy) isn't precocious in relation to output control of the outside terminal (TIOCxy).

Timer counter (TCNTCx) is started for counting up by setting a bit in timer start register (TSTRC) to 1. Set the value in GRCxy before starting the counter. When the values in GRCxy and TCNTCx match, a bit corresponding to GRCxy in timer status register (TSRCx) is set and a waveform is output on external pin TIOCxy.

The compare match flag is set and the signal level on pin TIOCxy is changed on the first edge of the PCLK immediately after compare match between GRCxy and TCNTCx. Compare match flags (IMFCxy bits in TSRCx) can be cleared to 0 by writing 1 to the corresponding clear bit in the timer status clear register (TSCRCx).

A logical one, a logical zero, or a toggled output can be selected for the signal to be output on pin TIOCxy.

An interrupt request can be output to the CPU at occurrence of a compare match.

When PWM mode is disabled (TCRCx.PWMx0 = 0) and the CLRCx bit of timer control register Cx (TCRCx) is enabled, timer counter (TCNTCx) is cleared if its value matches the value of counter upper limit value setting compare register Cx (CUCRCx).

Figure 30.39 shows an operation example of compare match. In this example, a toggled output on GRC00, a logical one output on GRC01, and a logical zero output on GRC02 are externally output. A value of 004004 is set for GRC0y, which changes the TIOC0y output on the next PCLK after compare match with the TCNTC0 value.

Output compare registers Cxy (OCRCxy) is compared with timer counter (TCNTCx), and if they match, the OCMFCxy bit in timer status register (TSRCx) corresponding to OCRCxy is set.

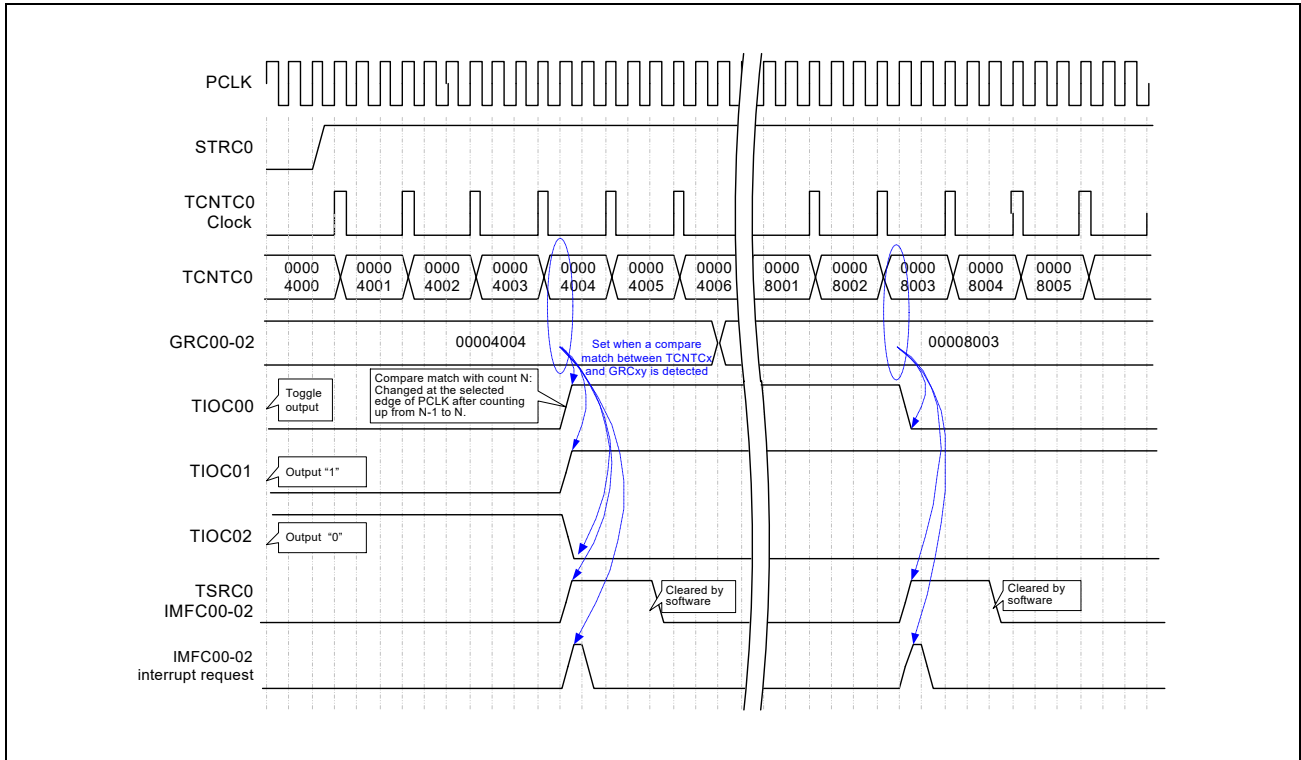


Figure 30.39 Operation Example of Compare Match

By setting a forced compare match bit (FMCM_{xy}) in timer control register (TCRC_x), compare match can be generated even if TCNTC_x has not matched GRC_{xy}. The compare match flag and the signal level on pin TIOC_{xy} are changed on the first edge of the PCLK after bit FCMFC_{xy} is set to 1.

Compare match is detected by GRC_{xy} when any of the following occurs.

- When the values in TCNTC_x and GRC_{xy} match (forced compare match is OFF)
- When the forced compare match bit (bit FCMC_{xy} in TCRC_x) is changed from 0 to 1
- When TCNTC_x sets the value besides 00_H as RCR2C_x, and it's in the range set as RCR2C_x at the time of writing in GRC_{xy}.

Compare match is detected by OCRC_{xy} when any of the following occurs.

- When the values in TCNTC_x and OCRC_{xy} match (forced compare match is OFF)
- When the forced compare match bit (bit FOCMC_{xy} in TCRC_x) is changed from 0 to 1
- When TCNTC_x sets the value besides 00_H as RCR1C_x, and it's in the range set as RCR1C_x at the time of writing in GRC_{xy}.

The compare match flag and the signal level on pin TIOC_{xy} is changed on the first edge of the PCLK after bit FCMC_{xy} is set to 1. Make sure that the compare match operation is selected by bits IOC_{xy}[2:0] in TIORC_x before starting operation. At the moment TCNTC_x and GRC_{xy} are set to the same value, or when the forced compare match bit is set to "1", compare match is not detected at the moment when compare match is enabled.

Compare match is detected regardless of the counter operating state. Even if the counter is stopped, compare match occurs when the condition is satisfied.

When the compare match status flag is cleared before GRC_{xy} and TCNTC_x are changed (such as before counting while the counter has been stopped), the compare match is not detected.

30.7.3.3 PWM Function

Setting bit PWMx in timer control register C (TCRCx) to 1 makes channels 1 to 3 in each subblock function as PWM timers with the same frequency. In PWM mode, GRCx0 as a cycle setting register and GRCx1 to GRCx3 as duty cycle setting registers are used. External pins TIOCx1 to TIOCx3 corresponding GRCx1 to GRCx3 are used to output PWM signals. To use them as PWM signal outputs, select the compare match operation by bits IOCxy in TIORCx as well as setting bit PWMx so that GRCn0 to GRCn3 function as compare match registers.

Timer counter (TCNTCx) is started by setting timer start register (TSTRC). When the value in TCNTCx reaches the value in the cycle setting register (GRCx0), compare match occurs and the bit in timer status register C (TSRCx) is set to 1. At this time, TCNTCx is cleared and a signal is output on external pins TIOCx1 to TIOCx3 according to bit IOCx0 in PWM mode. The output signal levels on pin TIOCx0 depend on bit IOCx0.

When the value in TCNTCx reaches the value in the duty cycle setting register (GRCx1 to GRCx3), the bit in timer status register C (TSRCx) is set to 1 and a signal is output on external pins TIOCx1 to TIOCx3 according to bits IOCx1 to IOCx3.

When the same value is set in the cycle and duty cycle setting registers, priority is given to bit IOCx0 corresponding to the cycle setting register.

Figure 30.40 shows an operation example of block C0 in PWM mode.

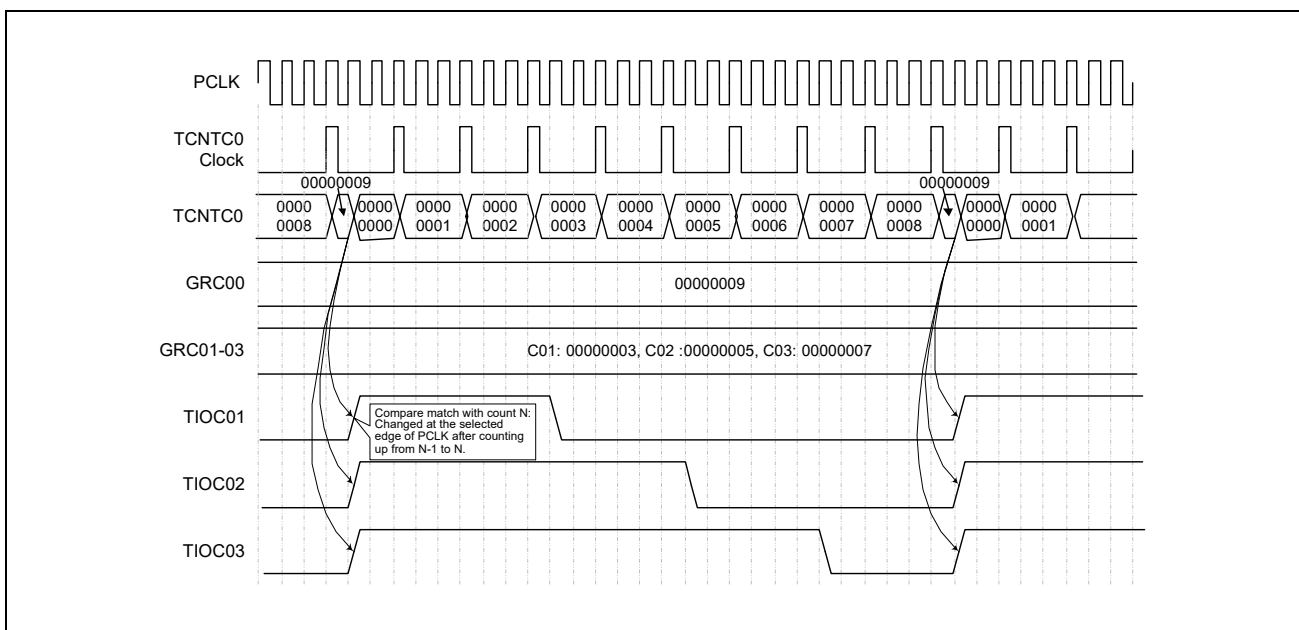


Figure 30.40 Operation Example of PWM Mode

The settings to obtain the above waveforms are IOC00 = 10 (for the output of 1 on TIOC00) and IOC01, 02, 03 = 01 or 11 (for the output of 0 on or inversion of TIOC01, 02, 03).

CAUTION

Operation will not be correct for PWM output with a duty cycle of 0%. Do not set GRCxy to 0, since the output will not be as expected in this case (in the case of frequency division by one, operation becomes the same as with a duty cycle of 100%, and in the case of division by two, pulses of one cycle of PCLK will be generated).

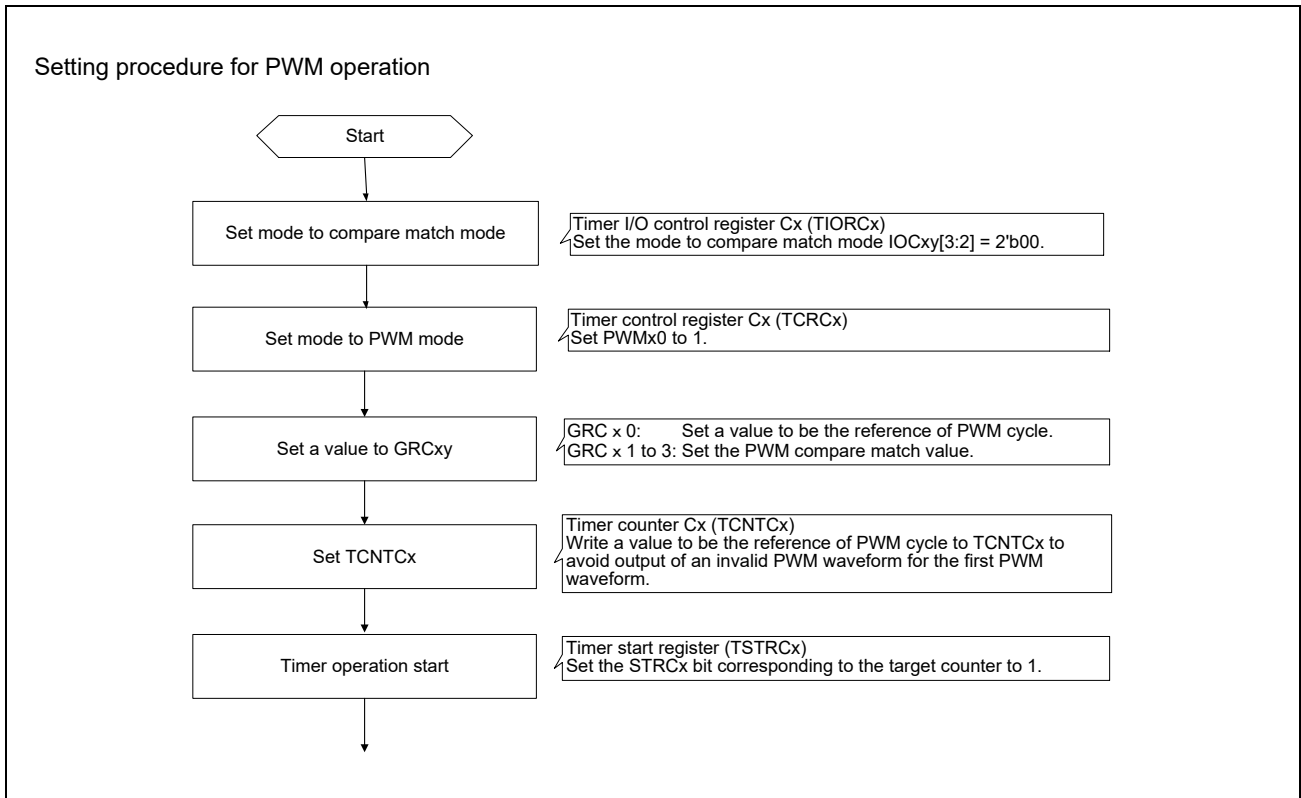


Figure 30.41 PWM Operation Setting Procedure

30.7.3.4 One-Shot Pulse Mode (One-Shot Pulse Function)

By setting the PWMx0 bits in timer control register Cx (TCRCx) to 0_B and setting the IOCxy[3:2] bits in timer I/O control register Cx (TIORXCx) to 10_B, one-shot pulses are output from the corresponding external pin (TIOCxy).

Setting the timer start register (TSTRC) starts up-count of the timer counter (TCNTCx).

In one-shot pulse mode, a one-shot pulse is output at a trigger of write access to the GRCxy register.

After the write access to the GRCxy register, a compare match interrupt is output on compare match between timer counter Cx (TCNTCx) and OCRCxy register, set the external pin (TIOCxy) to active level, a compare match interrupt is output on compare match between timer counter Cx (TCNTCx) and GRCxy register, and then reset the external pin (TIOCxy) to inactive level. When a single one-shot pulse output has been completed after an access to the GRCxy register, no interrupt occurs until the next write access to the GRCxy register is made and the external pin (TIOCxy) retains inactive level.

When one-shot pulse mode (IOCxy[3:2] = 10_B) is used, initialize the external pin (TIOCxy) by forced compare match Cxy (FCMCxy) after operating mode is set, and then start counting.

If one-shot pulse mode is activated without initializing the external pin (TIOCxy) to inactive level, output level becomes unstable.

To cancel one-shot pulse mode after a value is written to the GRCxy register, write 1 to forced compare match Cxy (FCMCxy) and forced output compare match Cxy (FOCMCxy) simultaneously.

TIORC.IOCxy				TCRCx	TCRC.	TCRC.	OCRCxy	GRCxy	CUCRCx	RCR1Cx	RCR2Cx	The upper limit value of TCNTCx
[3]	[2]	[1]	[0]	PWMx0	CLRCx	CLRCSELCx						
1	0	0	0	0 *(1 setting prohibited)	0/1		x	x	0/1	x	x	[When TCRC.CLRCx = 0] FFFF_FFFF _H
		0	1				Cmp	Cmp		o	o	[When TCRC.CLRCx = 1] CUCRCx or FFFF_FFFF _H
		1	0									
		1	1									

Note: Cmp: It can be used as a compare match Cap: It can be used as a capture

o: Setting possible, x: Setting prohibited

Figure 30.42 shows an example of one-shot pulse mode operation in block C0.

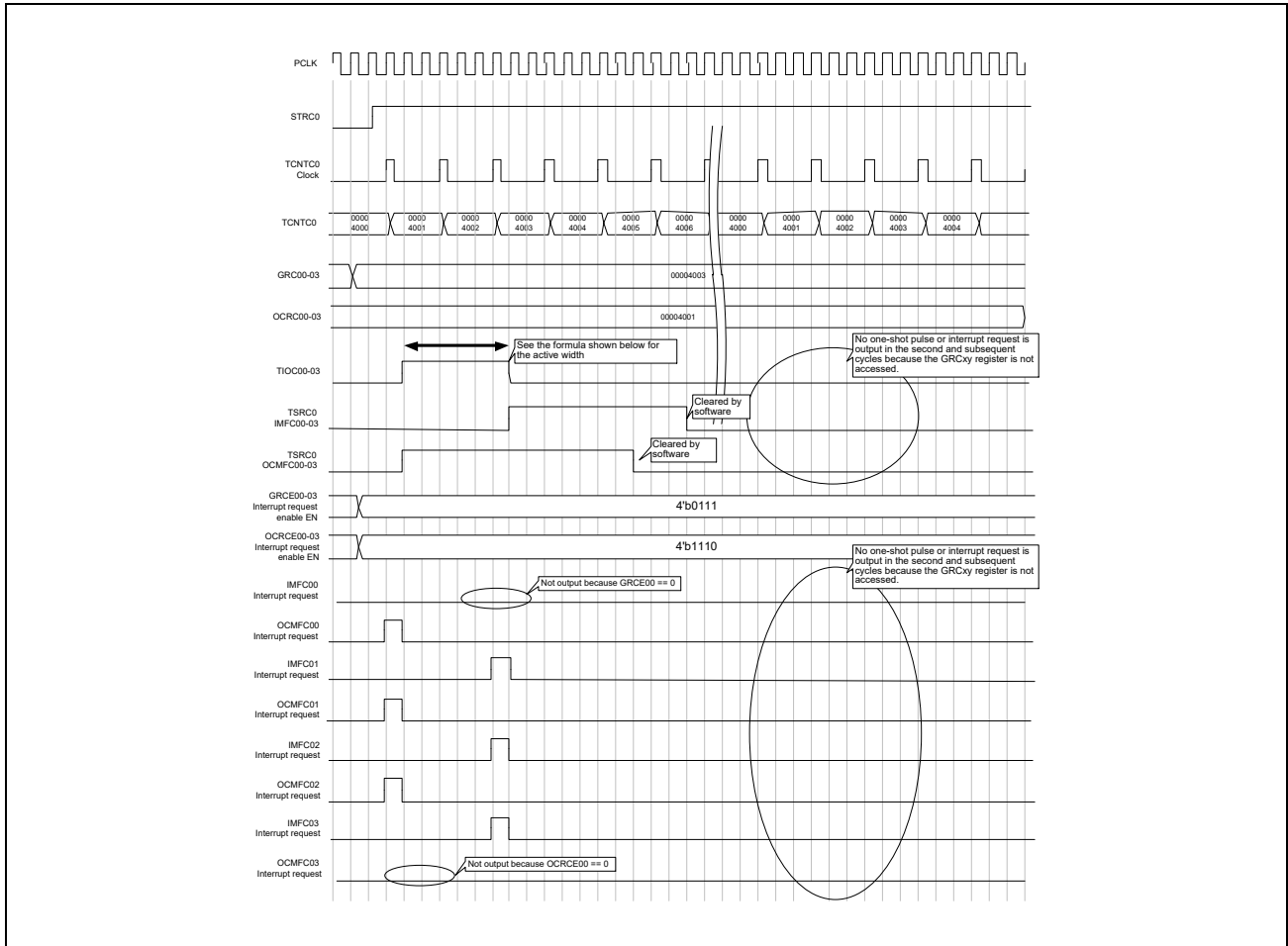


Figure 30.42 One-Shot Pulse Mode Operation

NOTE

The active width of one-shot pulse can be obtained from the following formula.

- $GRC_{xy} = OCR_{Cxy}$:
One-shot pulse (active) width = 0
- $GRC_{xy} > OCR_{Cxy}$:
One-shot pulse (active) width = $GRC_{xy}[31:0] - OCR_{Cxy}[31:0]$
- $GRC_{xy} < OCR_{Cxy}$:
 - Upper-limit setting function is off
One-shot pulse (active) width = $(FFFF\ FFFF_H - OCR_{Cxy}[31:0]) + (GRC_{xy}[31:0] + 1)$
 - Upper-limit setting function is on
One-shot pulse (active) width = $(CUCRC_x - OCR_{Cxy}[31:0]) + GRC_{xy}[31:0]$

To output a one-shot pulse in the second and subsequent cycles, be sure to confirm that the previously set one-shot pulse output has been completed, and then set GRCxy and OCRCxy in the second and subsequent cycles. If the GRCxy or OCRCxy register value is modified before the one-shot pulse output has been completed, the modified value is reflected to the ongoing one-shot-pulse output.

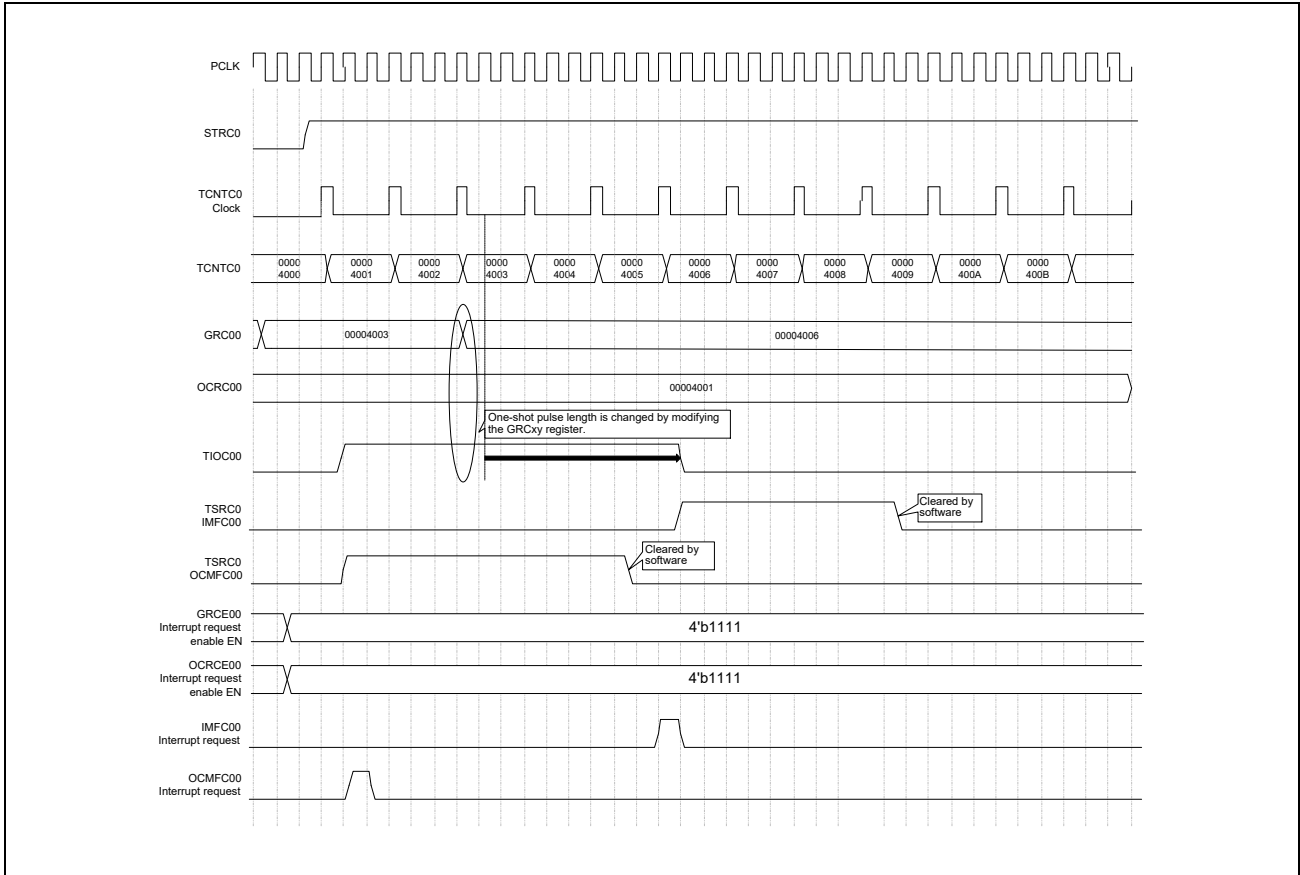


Figure 30.43 Operation when the Compare Register Is Modified during One-Shot Pulse Output

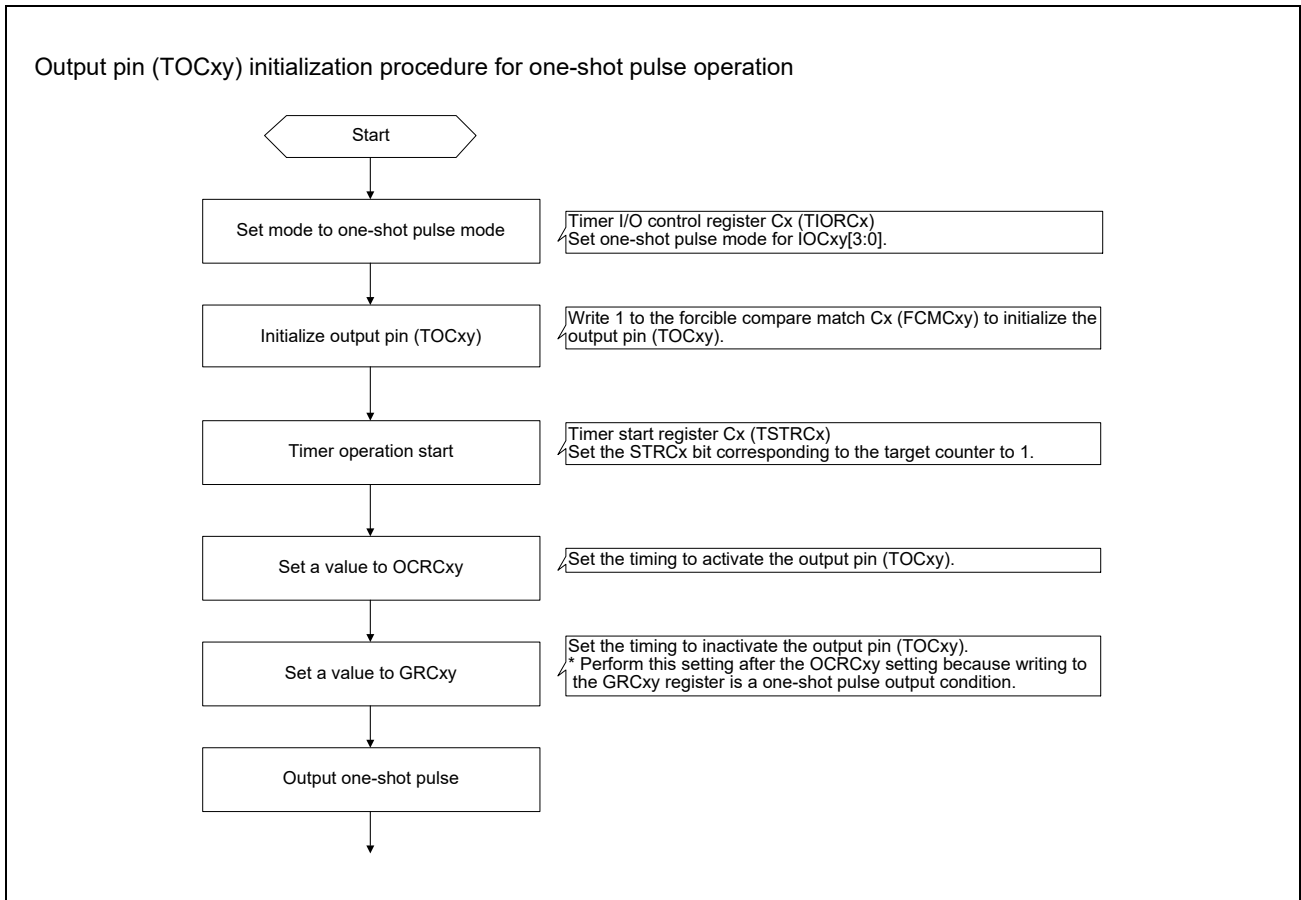


Figure 30.44 One-Shot Pulse Operation (Output Pin Initialization Procedure)

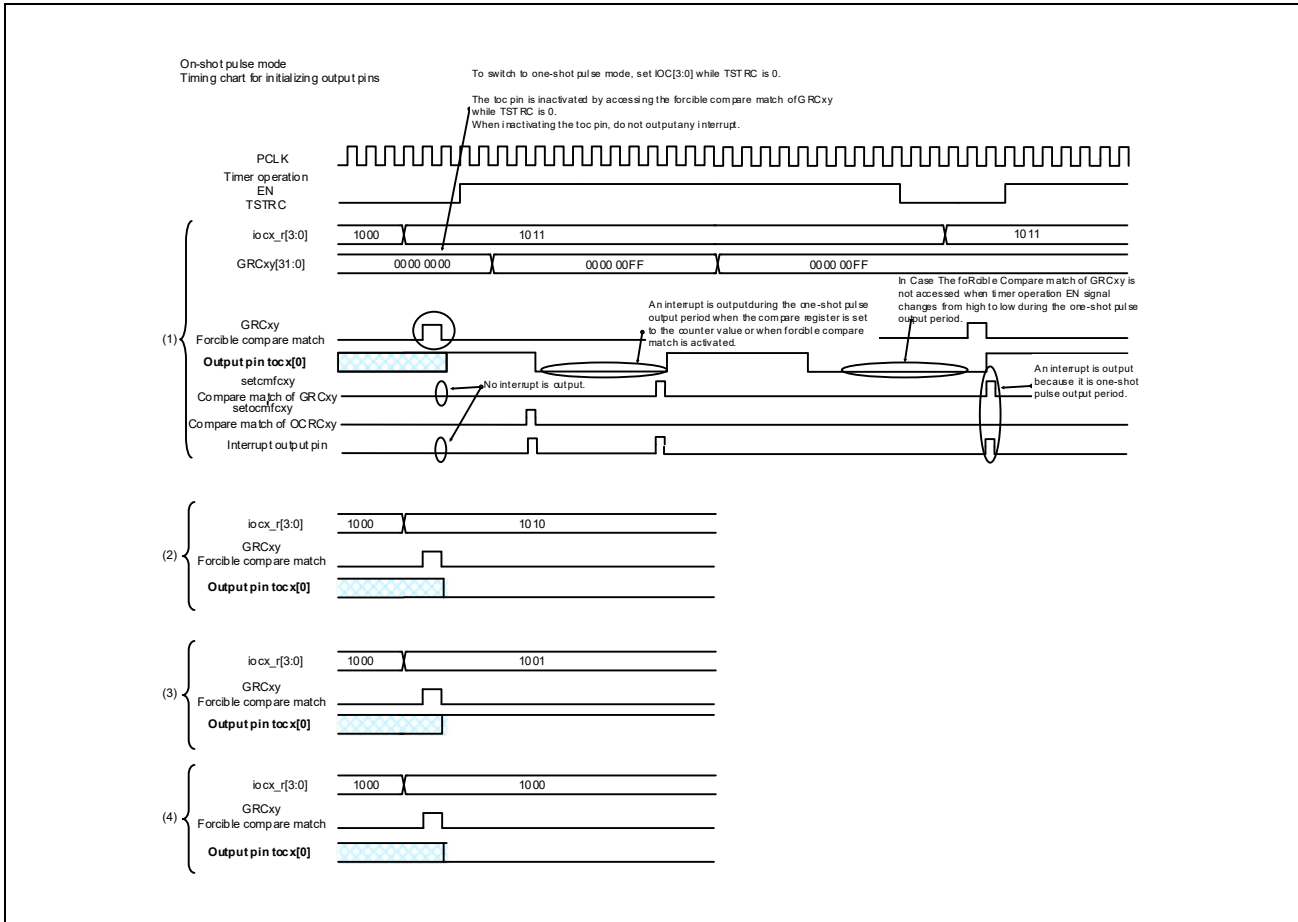


Figure 30.45 One-Shot Pulse Operation (Output Pin Initialization Timing Chart)

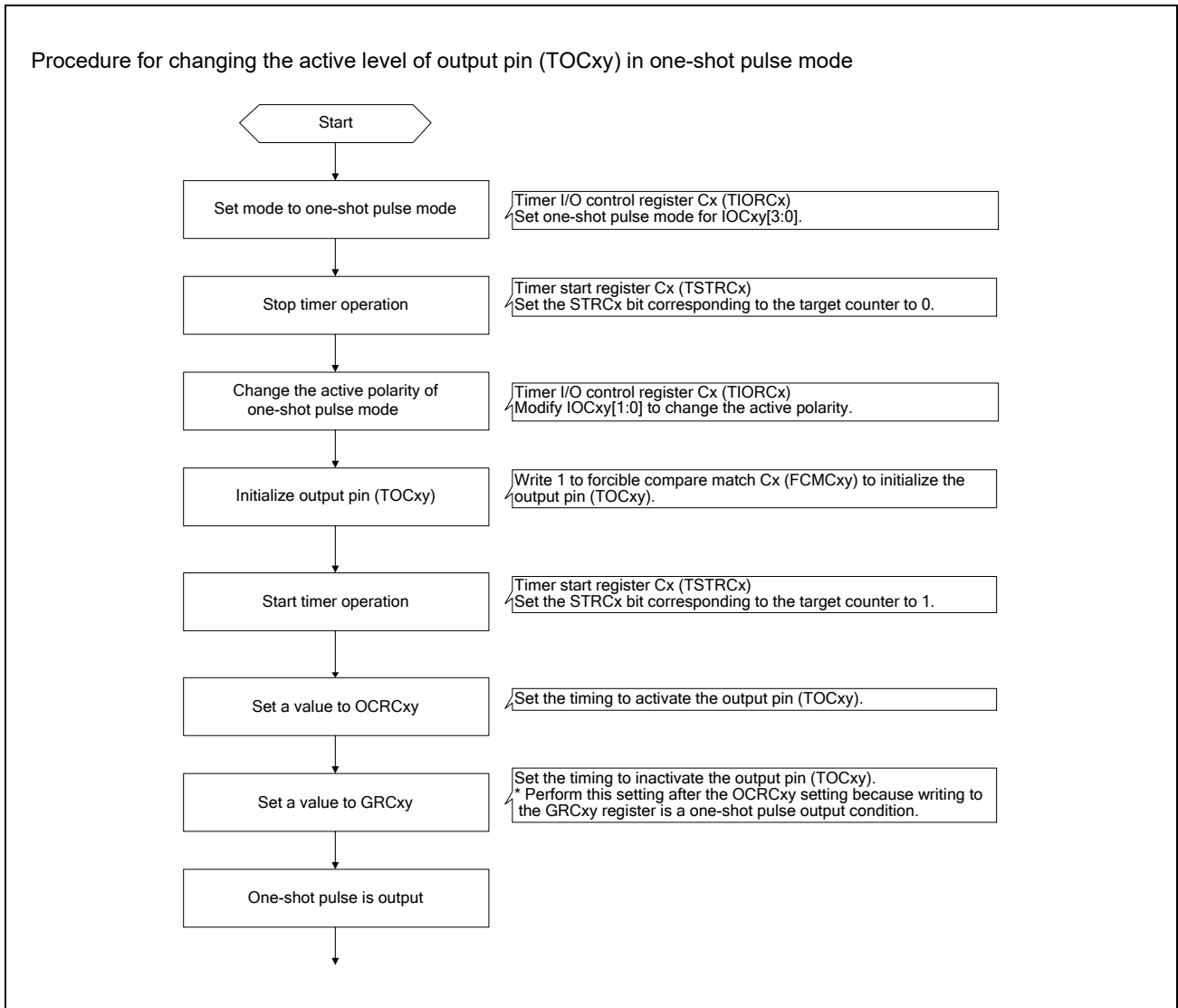


Figure 30.46 One-Shot Pulse Operation (Outline of Active Polarity Change)

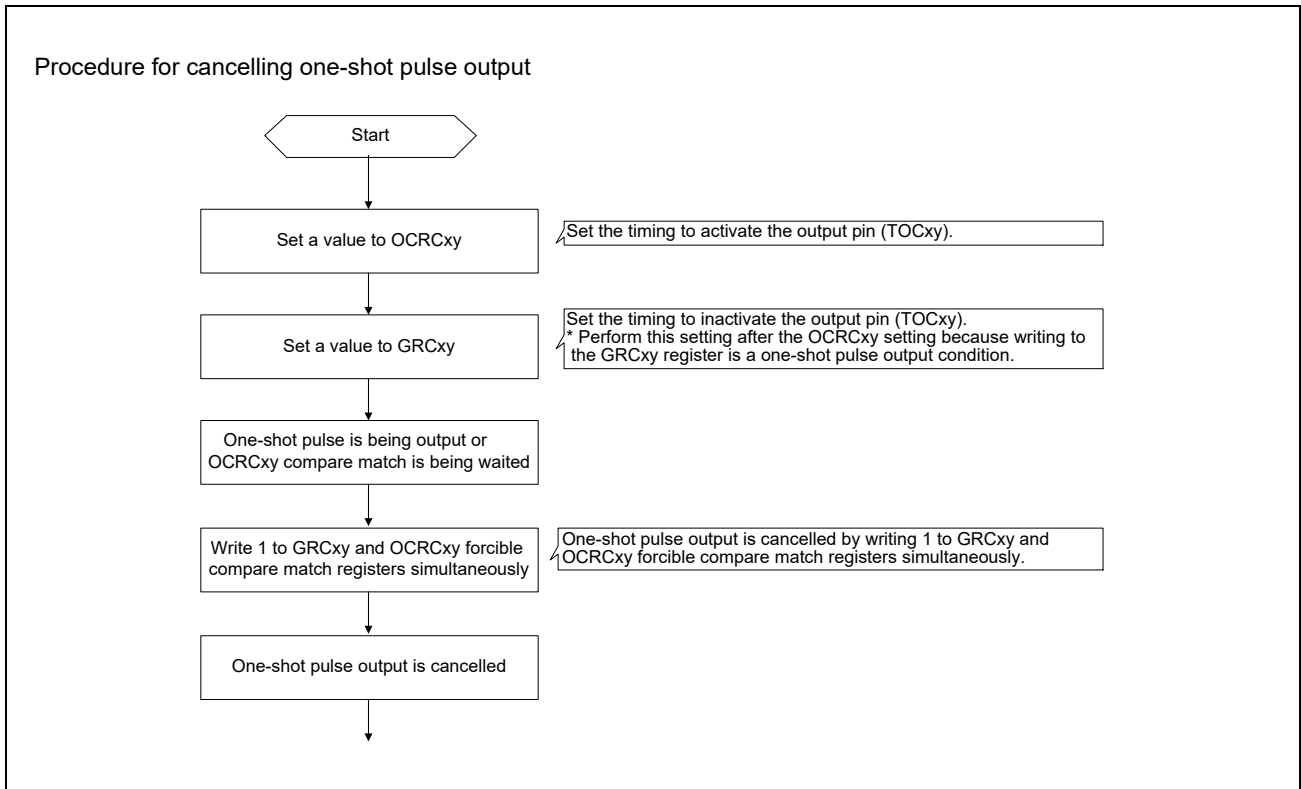


Figure 30.47 One-Shot Pulse Output Cancellation Procedure

30.7.3.5 Counter Upper-Limit Setting Function

The timer counter (TCNTCx) upper-limit value in each subblock can be changed by the counter upper-limit setting compare register Cxy (CUCRCx) to perform interval operation. To enable the counter upper-limit setting function, set the CLRCx bit in the timer control register C (TCRCx) to 1.

Figure 30.48 shows an example of operation with the counter upper-limit setting function of block C0 enabled.

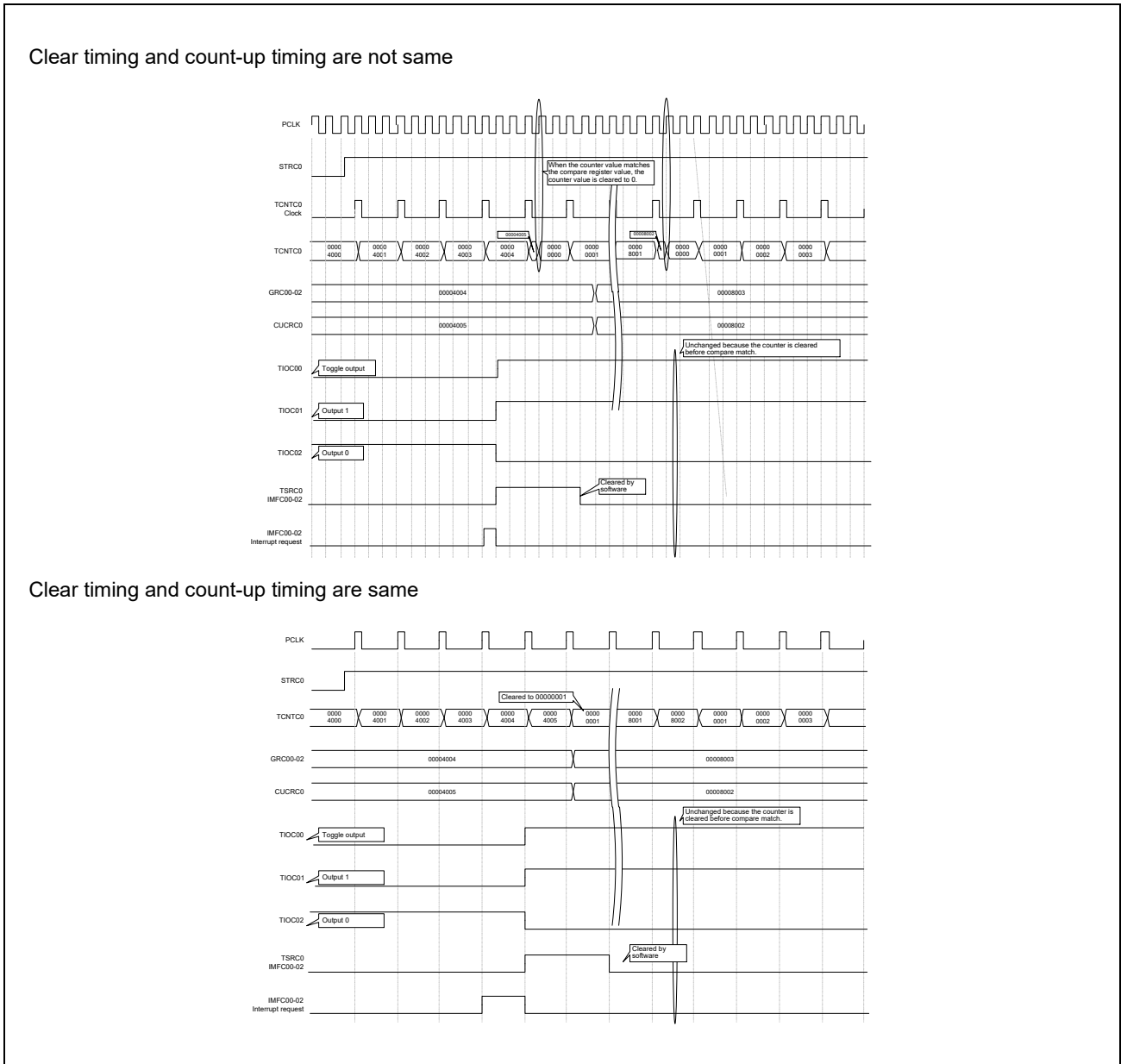


Figure 30.48 Example of Upper-Limit Setting Function Operation

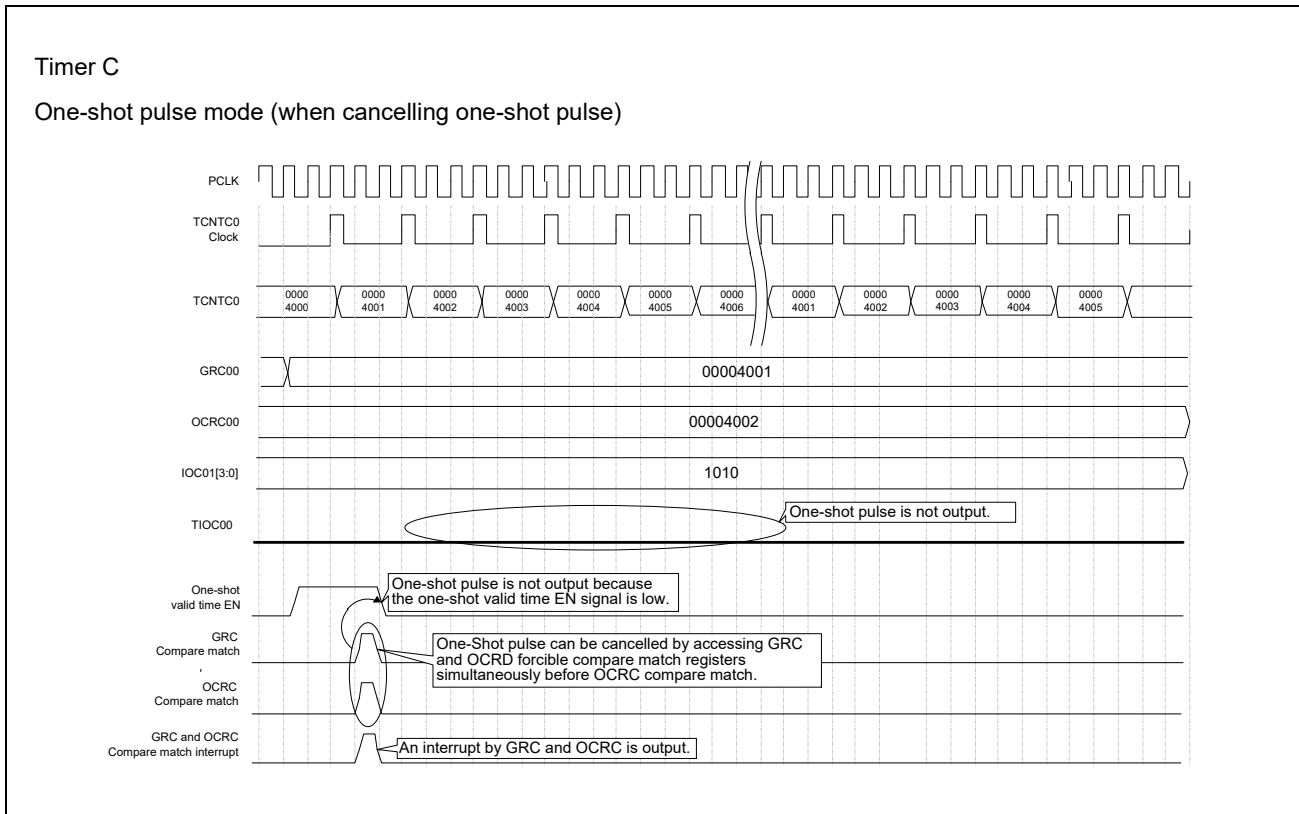


Figure 30.49 One-Shot Pulse Output Cancellation Timing Chart

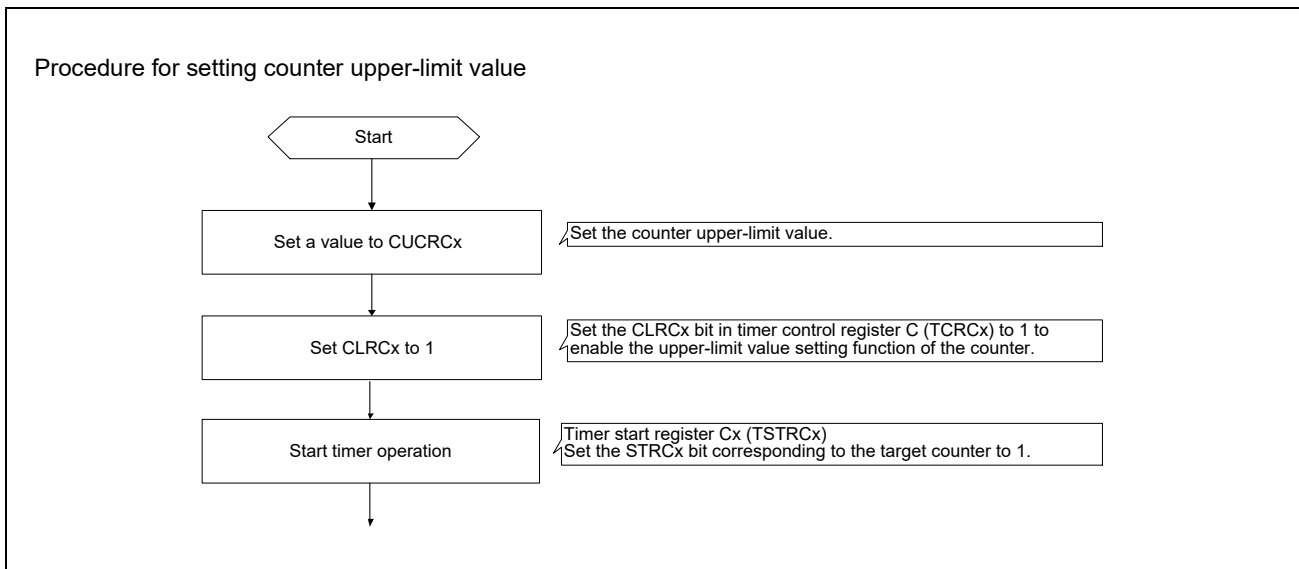


Figure 30.50 Counter Upper-Limit Setting Function Setting Procedure

30.7.3.6 Range Compare Function

(1) Range compare function

The range compare function generates a compare match when the counter value (TCNTCx) is within the set compare range when writing to “output compare register (OCRCxy) and timer general register Cxy (GRCxy)”. When using “PWM mode (PWMx0 = 1)” or “TIORCx.IOCxy setting OCRCxy, GRCxy as the capture register”, the range compare function is prohibited. The range compare function is enabled when setting other than 00H to the range compare value setting register (RCR1Cx or RCR2Cx).

The range compare value setting register 1Cx (RCR1Cx) and the range compare value setting register 2Cx (RCR2Cx) correspond to the output compare register (OCRCxy) and timer general register Cxy (GRCxy), respectively.

Range of OCRCxy, GRCxy Judge the range of compare match using the following values.

OCRCxy Range Compare Match

- Write value of OCRCxy
- Value of TCNTCx
- Range value set in RCR1Cx
- TCNTCx upper limit value (FFFFFFFH [TCRCx.CLRCx = 0] or CUCRCx setting value [TCRCx.CLRCx = 1])

GRCxy Range Compare Match

- Write value of GRCxy
- Value of TCNTCx
- Range value set in RCR2Cx
- TCNTCx upper limit value (FFFFFFFH [TCRCx.CLRCx = 0] or CUCRCx setting value [TCRCx.CLRCx = 1])

Figure 30.51, **Figure 30.52** show the operation diagram of OCRCxy range compare match.

Also, OCRCxy and GRCxy capture the counter value (TCNTCx) when a compare match is generated by the range compare function. **Figure 30.53** shows the capture operation when range compare match occurs.

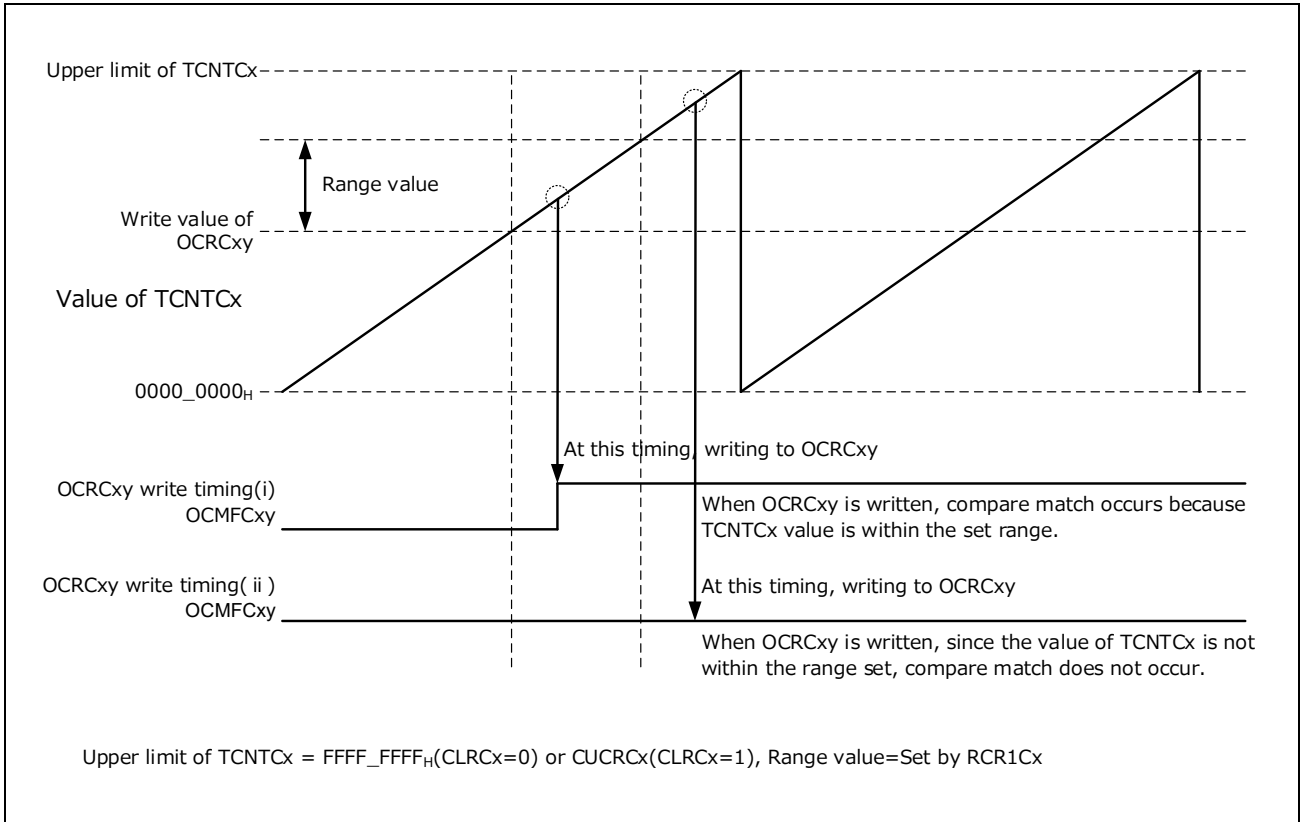


Figure 30.51 Operation of the Range Comparison Function(1)

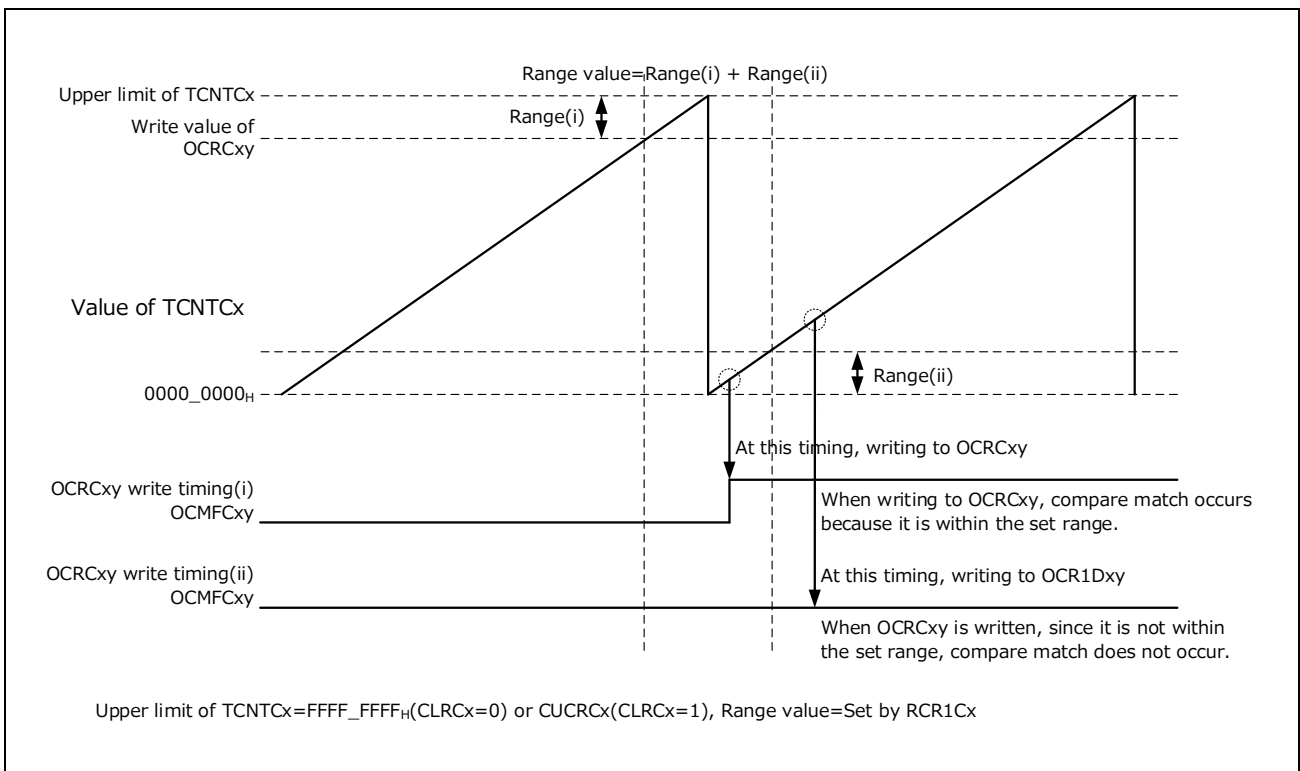


Figure 30.52 Operation of the Range Comparison Function(2)

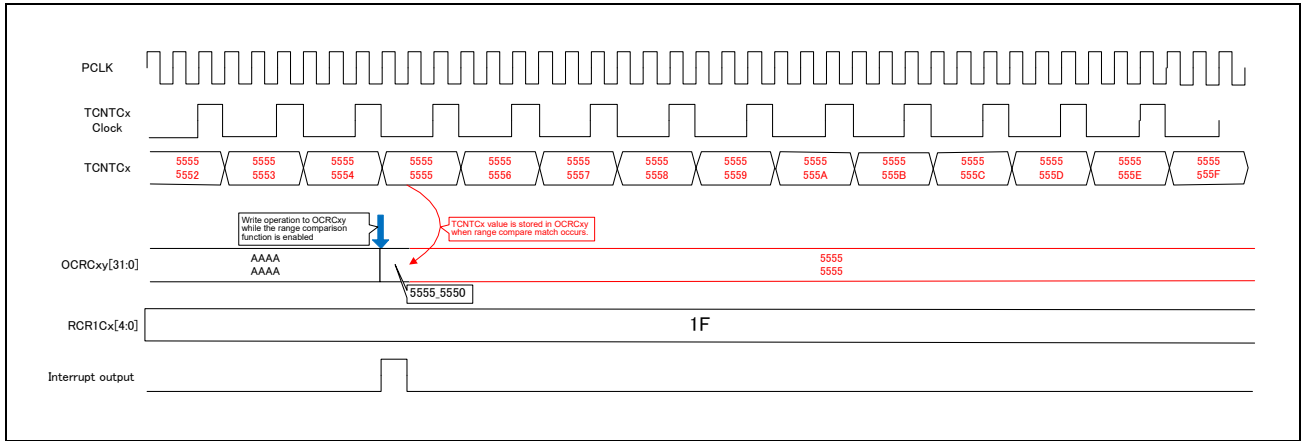


Figure 30.53 Range Compare Counter Value Capturing Operation

OCRCxy range compare match occurs when the condition of **Table 30.118** is satisfied when writing to OCRCxy by setting the range compare function to valid (except for 00_H in RCR1Cx).

Table 30.118 OCRCxy Range Compare Match Generation Condition

	Range condition	Range compare match generation condition in OCRCxy
Condition 1	When the set compare range is less than or equal to the TCNTCx upper limit value. OCRCxy + Range value set in RCR1Cx ≤ TCNTCx upper limit value.	OCRCxy ≤ TCNTCx value ≤ Range value set for OCRCxy + RCR1Cx
Condition 2	When the set compare range exceeds the TCNTCx upper limit value. TCNTCx upper limit < Range value set for OCRCxy + RCR1Cx	Range(i): OCRCxy ≤ TCNTCx value ≤ TCNTCx upper limit value or Range(ii): 00000000 ≤ TCNTCx value ≤ OCRCxy + Range value set for RCR 1Cx + TCNTCx upper limit value

Note 1. "Condition 1 When a set comparing area is below the TCNTCx upper limit value.", The range condition shown in the **Figure 30.51**.

Note 2. "Condition 2 When a set comparing area exceeds TCNTCx.", The range condition shown in the **Figure 30.52**.

Note 3. Range (i) and Range (ii) in the table indicate Range (i) and Range (ii) in **Figure 30.52**.

GRCxy range compare match occurs if the condition of **Table 30.119** is satisfied when writing to GRCxy by setting the range compare function to valid (except for H'00 in RCR2Cx).

Table 30.119 GRCxy Range Compare Match Generation Condition

	Range condition	Range compare match generation condition in GRCxy
Condition 1	When the set compare range is less than or equal to the TCNTCx upper limit value. Range value set for GRCxy + RCR2Cx ≤ TCNTCx upper limit value	GRCxy ≤ TCNTCx value ≤ Range value set for GRCxy + RCR2Cx
Condition 2	When the set compare range exceeds the TCNTCx upper limit value TCNTCx upper limit value < Range value set for GRCxy + RCR2Cx	Range(i): GRCxy ≤ TCNTCx value ≤ TCNTCx upper limit value or Range(ii): 00000000 _H ≤ TCNTCx value ≤ Range value set for GRCxy + RCR2Cx - TCNTCx upper limit value

Note 1. "Condition 1 When a set comparing area is below the TCNTCx upper limit value.", The range condition shown in the **Figure 30.51**.

Note 2. "Condition 2 When a set comparing area exceeds TCNTCx.", The range condition shown in the **Figure 30.52**.

Note 3. Range (i) and Range (ii) in the table indicate Range (i) and Range (ii) in **Figure 30.52**.

The table below shows the setting example when the range compare function is used and the counter value at which the range compare match occurs.

No.1, No.3 are examples of the condition “When the set compare range is less than or equal to the TCNTCx upper limit value”, and No.2, No.4 are examples of the condition “When the set compare range exceeds the TCNTCx upper limit value”.

Table 30.120 Range Compare Function Register Setting Example

No.	CLRCx setting value	CUCRCx setting value	The write value of OCRCxy or GRCxy	Set value (range value) of range compare value setting register (RCR1Cx, RCR1Cx)	When writing to the compare register (OCRCxy, GRCxy), the value of the counter where range compare match occurs
1	0	—	FFFF0000 _H	12 (range value: FFF _H)	FFFF0000 _H to FFFF0FFF _H *1
2	0	—	FFFF0000 _H	18 (range value: 3FFFF _H)	FFFF0000 _H to FFFFFFFF _H 00000000 _H to 00030000 _H *2
3	1	00000FFF _H	00000E00 _H	6 (range value: 3F _H)	00000E00 _H to 00000E3F _H *3
4	1	00000FFF _H	00000E00 _H	10 (range value: 3FF _H)	00000E00 _H to 00000FFF _H 00000000 _H to 00000200 _H *4

Note 1. FFFF0000_H to (FFFF0000_H + 2¹² -1)

Note 2. 00000000_H to (FFFF0000_H + 2¹⁸ -1 – FFFFFFFF_H)

Note 3. 00000E00_H to (00000E00_H + 2⁶ -1)

Note 4. 00000000_H to (00000E00_H + 2¹⁰ -1 – 00000FFF_H)

30.8 Timer D

30.8.1 Overview of Operation

The timer D block is a one-shot pulse output timer that consists of SB_D subblocks. Each subblock has four channels.

Timer D has the following functions.

- A compare match between the counter and output compare register is generated. Compare match A by compare match of TCNT1Dx (or TCNT2Dx) and OCR1Dx and compare match B by compare match of TCNT2Dx (or TCNT1Dx) and OCR2Dx are generated respectively.
- Down counter is started on compare match A or B of the output compare register or writing to the counter start bit in the down-counter start register. A one-shot pulse with an offset can be output.
- The output waveform can be forcibly shutoff regardless of the down-counter value by compare match B of the compare match register. If compare match A or counter start bit writing occurs concurrently with compare match B, output shutoff takes precedence.
- The input capture register enables to capture the TCNT2Dx (or TCNT1Dx) value using compare match A as a trigger and the TCNT1Dx (or TCNT2Dx) value using compare match B as a trigger.
- Pulse indicating that compare match A or B can be output.
- Interrupt requests can be output on compare matches A and B ($8 \times SB_D$ lines supported).
- Range compare function by the compare match register is enabled according to the range comparison value setting register.
- The counters (TCNT1Dx and TCNT2Dx) can be cleared upon compare match.
 - When the upper limit setting of TCNT1Dx is valid, the counters can be cleared upon compare match with CUCR1Dx.
 - When the upper limit setting of TCNT2Dx is valid, the counters can be cleared upon compare match with CUCR2Dx.
- Interrupt requests ($2 \times SB_D$ in total) can be output on counter overflow: SB_D outputs from TCNT1Dx and SB_D outputs from TCNT2Dx.
- Interrupt requests can be output on down-counter underflow ($4 \times SB_D$ lines supported).
- Offset base register can capture the counter value by a trigger signal from timer A and a PH notification trigger from DFE
- Clearing TCNT1Dx and TCNT2Dx is possible by a clear signal issued from timer B.
- The value configured in the output value register can be output from pins according to the output select register.

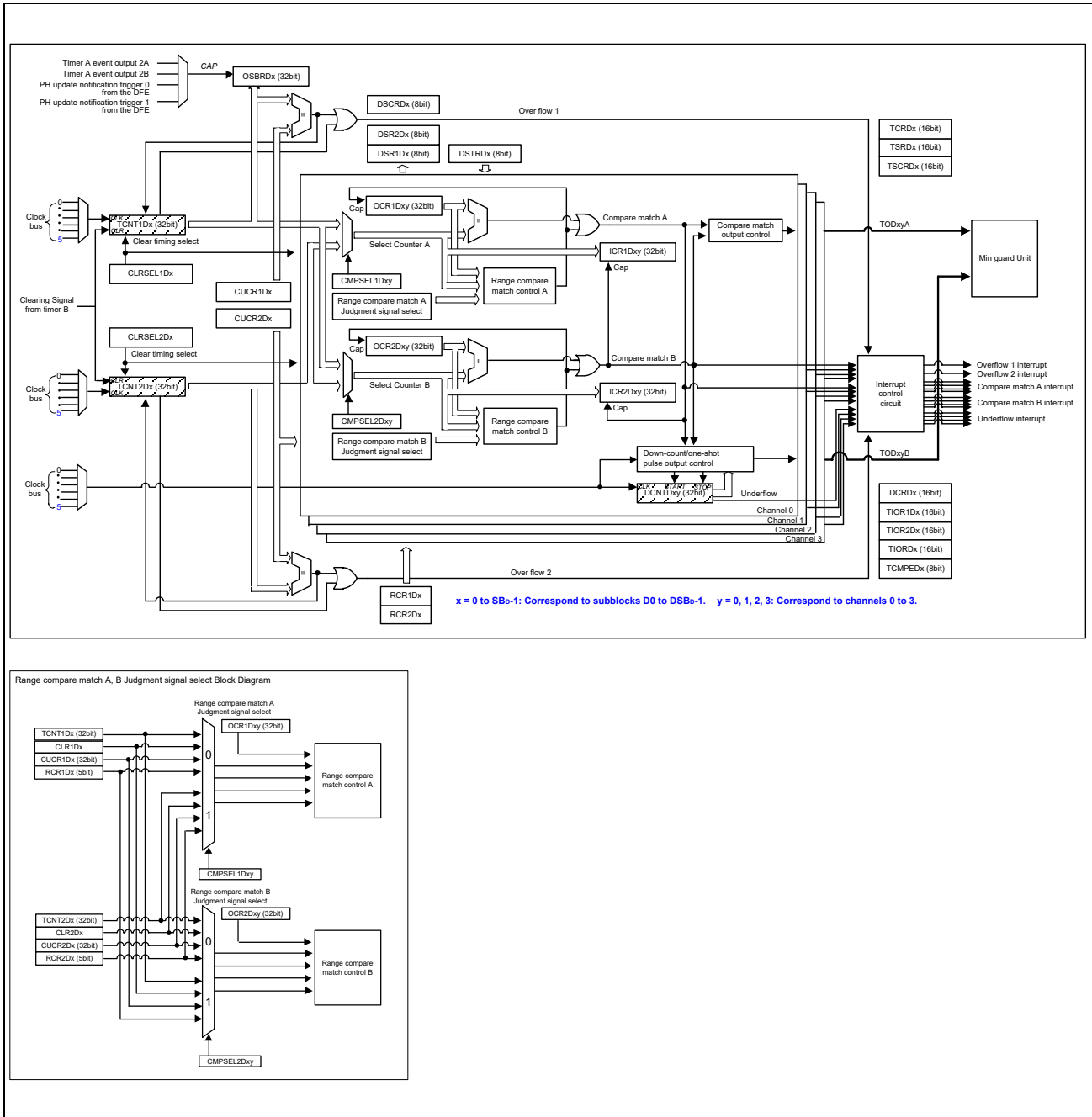


Figure 30.54 Block Diagram of Timer D

Figure 30.54 is a block diagram of timer D. Components of timer D subblock Dx include two range compare setting registers (RCR1Dx and RCR2Dx), two 32-bit timer counters (TCNT1Dx and TCNT2Dx), one offset base register (OSBRDx), one output value register (ORDx), one output select register (OSELRDx), SB_D compare match registers (OCR1Dxy and OCR2Dxy), two counter upper-limit setting registers (CUCR1Dx and CUCR2Dx), four timer down-counters Dxy (DCNTDxy), SB_D input capture registers (ICR1Dxy and ICR2Dxy), and a controller. Each channel includes two output pins; TODxyA for compare match output and TODxyB for one-shot pulse output. (Output pins are supported by subblocks D0 to DSB_{D-1}.)

TODxyA and TODxyB are output a level of 0 as a default.

When compare match A or B is detected, a pulse is output as an interrupt request.

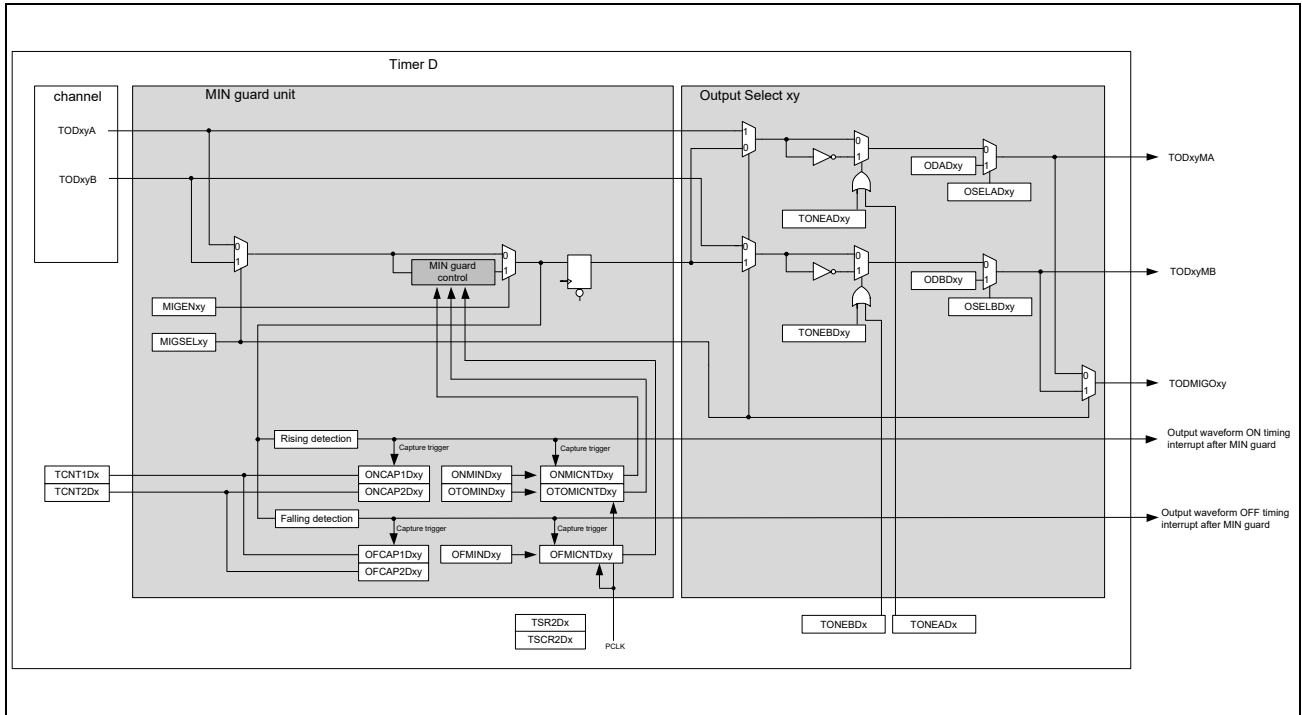


Figure 30.55 Timer D Channel xy MIN Guard Unit Connection Diagram

Figure 30.55 shows the block diagram of the timer D MIN guard unit.

The MIN guard unit is connected to each channel of timer D, and it is possible to add MIN guard function to TODxyA (compare match output) or TODxyB (one shot pulse output) and it can output as TODxyMA, TODxyMB and TODMIGOxy.

It is also possible to output an interrupt request at the ON timing and OFF timing of the output waveform after the MIN guard to the output that selected the MIN guard function, and capture the values of TCNT1Dx and TCNT2Dx at the timing of interrupt output.

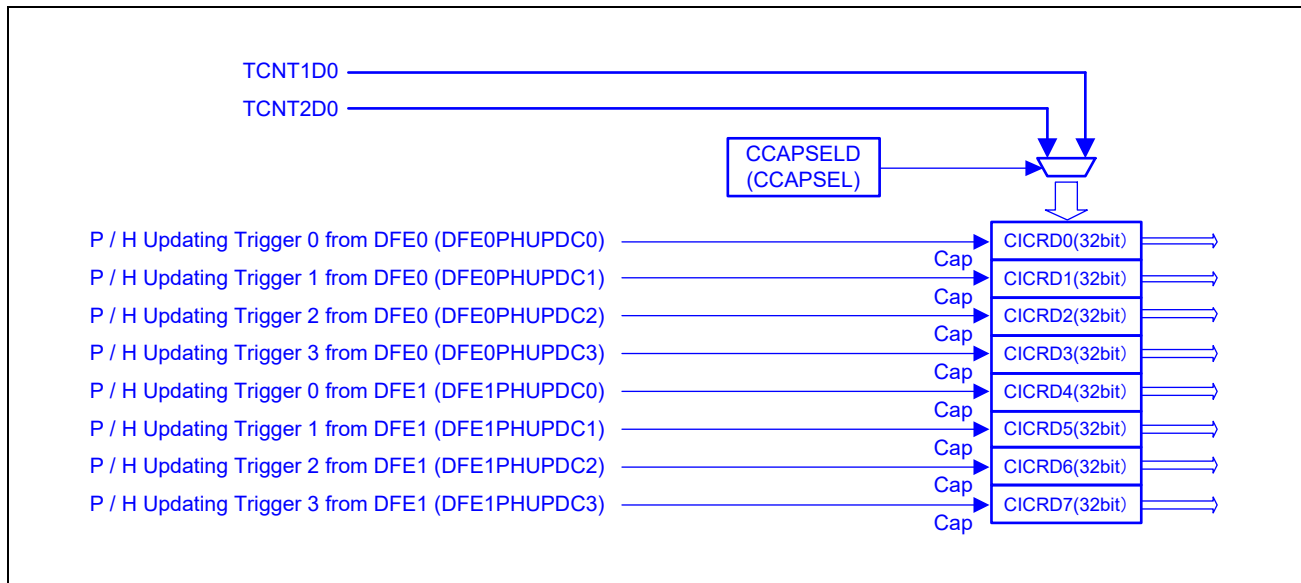


Figure 30.56 Common Input Capture Register

Figure 30.56 is a common input capture select register. A common input capture register (CICRD0-7) is prepared 8, and stocks the value of TCNT1D0 in subblock 0 or TCNT2D0 at the time of PH renewal notice trigger 0-7 occurrence from DFE.

Transmission of DFE to timer trigger

Timer D trigger of compare match A (TODxyA) and compare match B (TODxyB) can be used as a timer trigger source for DFE. **Table 30.121** shows pairs of DFE's timer trigger (compare A0/B0 to A19/B19 from the Timer) and compare match A / compare match B.

Table 30.121 Timer D to DFE Timer Trigger Pair Table (1/2)

Timer D (compare match A, B)	DFE (Timer Trigger)
Timer D0 compare match A (channel 0)	Timer trigger Compare (Compare A0)
Timer D0 compare match A (channel 1)	Timer trigger Compare (Compare A1)
Timer D0 compare match A (channel 2)	Timer trigger Compare (Compare A2)
Timer D0 compare match A (channel 3)	Timer trigger Compare (Compare A3)
Timer D1 compare match A (channel 0)	Timer trigger Compare (Compare A4)
Timer D1 compare match A (channel 1)	Timer trigger Compare (Compare A5)
Timer D1 compare match A (channel 2)	Timer trigger Compare (Compare A6)
Timer D1 compare match A (channel 3)	Timer trigger Compare (Compare A7)
Timer D2 compare match A (channel 0)	Timer trigger Compare (Compare A8)
Timer D2 compare match A (channel 1)	Timer trigger Compare (Compare A9)
Timer D2 compare match A (channel 2)	Timer trigger Compare (Compare A10)
Timer D2 compare match A (channel 3)	Timer trigger Compare (Compare A11)

Table 30.121 Timer D to DFE Timer Trigger Pair Table (2/2)

Timer D (compare match A, B)	DFE (Timer Trigger)
Timer D3 compare match A (channel 0)	Timer trigger Compare (Compare A12)
Timer D3 compare match A (channel 1)	Timer trigger Compare (Compare A13)
Timer D3 compare match A (channel 2)	Timer trigger Compare (Compare A14)
Timer D3 compare match A (channel 3)	Timer trigger Compare (Compare A15)
Timer D4 compare match A (channel 0)	Timer trigger Compare (Compare A16)
Timer D4 compare match A (channel 1)	Timer trigger Compare (Compare A17)
Timer D4 compare match A (channel 2)	Timer trigger Compare (Compare A18)
Timer D4 compare match A (channel 3)	Timer trigger Compare (Compare A19)
Timer D0 compare match B (channel 0)	Timer trigger Compare (Compare B0)
Timer D0 compare match B (channel 1)	Timer trigger Compare (Compare B1)
Timer D0 compare match B (channel 2)	Timer trigger Compare (Compare B2)
Timer D0 compare match B (channel 3)	Timer trigger Compare (Compare B3)
Timer D1 compare match B (channel 0)	Timer trigger Compare (Compare B4)
Timer D1 compare match B (channel 1)	Timer trigger Compare (Compare B5)
Timer D1 compare match B (channel 2)	Timer trigger Compare (Compare B6)
Timer D1 compare match B (channel 3)	Timer trigger Compare (Compare B7)
Timer D2 compare match B (channel 0)	Timer trigger Compare (Compare B8)
Timer D2 compare match B (channel 1)	Timer trigger Compare (Compare B9)
Timer D2 compare match B (channel 2)	Timer trigger Compare (Compare B10)
Timer D2 compare match B (channel 3)	Timer trigger Compare (Compare B11)
Timer D3 compare match B (channel 0)	Timer trigger Compare (Compare B12)
Timer D3 compare match B (channel 1)	Timer trigger Compare (Compare B13)
Timer D3 compare match B (channel 2)	Timer trigger Compare (Compare B14)
Timer D3 compare match B (channel 3)	Timer trigger Compare (Compare B15)
Timer D4 compare match B (channel 0)	Timer trigger Compare (Compare B16)
Timer D4 compare match B (channel 1)	Timer trigger Compare (Compare B17)
Timer D4 compare match B (channel 2)	Timer trigger Compare (Compare B18)
Timer D4 compare match B (channel 3)	Timer trigger Compare (Compare B19)

30.8.2 Registers Related to Timer D

30.8.2.1 TSTRD — Timer Start Register D

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	STRD8	STRD7	STRD6	STRD5	STRD4	STRD3	STRD2	STRD1	STRD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.122 TSTRD Register Contents

Bit Position	Bit Name	Function
15 to SB _D	—	These bits are not used. When these bits are read, “0” is always returned. When writing, always write “0”.
SB _D -1 to 0	STRD _x	Counter Dx Start 0: TCNT1D _x , TCNT2D _x , and DCNTD _{xy} are disabled 1: TCNT1D _x , TCNT2D _x , and DCNTD _{xy} are enabled Example: For b0 0: TCNT1D0, TCNT2D0, and DCNTD0y are disabled 1: TCNT1D0, TCNT2D0, and DCNTD0y are enabled

Note: A start bit is provided for each subblock D0 to DSB_D-1. (y = 0, 1, 2, 3: Corresponding to Dx0 to Dx3.)

Timer start register (TSTRD) is an 8-bit/16-bit readable/writable register. This register is set to enable or disable two timer counters (TCNT1D_x and TCNT2D_x) and down counters (DCNTD_{xy}) in SB_D subblocks Dx (x = 0 to SB_D-1). When the counter Dx start bit and the TDE bit in the ATU master enable register (ATUENR) are both set to 1, the counters are started.

TSTRD is initialized to 0000_H after reset.

Counter Dx Start Bit (STRD_x)

These bits enable and disable timer counters 1D_x and 2D_x (TCNT1D_x, TCNT2D_x) and down counters (DCNTD_{xy}).

When this bit is cleared to 0, TCNT1D_x, TCNT2D_x, and DCNTD_{xy} stop operation. The counter value is retained while the counter is stopped. When this bit is set to 1 again, the counter is restarted from the value. Note, however, that the counter is cleared if a clear upon match with the upper limit value setting register occurs while operation is stopped. When the count clear signal is input from timer B, the counter is cleared when restarting operation at C1CED_x in TCRD_x or the C2CED_x bit is effective.

Even if the counter D start bit is set to 1, counting does not start unless the timer D enable bit (TDE) is set to 1 by the ATU master enable register (ATUENR).

CAUTION

The prescalers run independently of the setting of the STRD_x bit and are not initialized at the start of TCNT1D_x and TCNT2D_x. Therefore, during the time between the activation and the start of actual count operation by the above counters, hardware-related uncertainty shorter than the period of selected count source (resolution) accompanies.

30.8.2.2 TCRDx — Timer Control Registers Dx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	OBRED _x	C2CED _x	C1CED _x	CLR2D _x	CKSEL2D _x [2:0]			CLR1D _x	CKSEL1D _x [2:0]			—	DCSELD _x [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.123 TCRDx Register Contents

Bit Position	Bit Name	Function
15	—	This bit is not used. When this bit is read, "0" is always returned. When writing, always write "0".
14	OBRED _x	Timer Offset Base Register Enable Enables and disables input capture to the offset base register.
13	C2CED _x	Counter 2 Clear Enable Enables and disables a timer counter 2 clearing request from timer B.
12	C1CED _x	Counter 1 Clear Enable Enables and disables a timer counter 1 clearing request from timer B.
11	CLR2D _x	TCNT2D _x Clear Setting Enables and disables clearing the timer counter at CUCR2D _x and TCNT2D _x are identical.
10 to 8	CKSEL2D _x [2:0]	TCNT2D _x Clock Select These bits select the TCNT2D _x counting-up clock from clock-bus lines 0 to 5.
7	CLR1D _x	TCNT1D _x Clear Setting Enables and disables clearing the timer counter at CUCR1D _x and TCNT1D _x are identical.
6 to 4	CKSEL1D _x [2:0]	TCNT1D _x Clock Select These bits select the TCNT1D _x counting-up clock from clock-bus lines 0 to 5.
3	—	This bit is not used. When this bit is read, "0" is always returned. When writing, always write "0".
2 to 0	DCSELD _x [2:0]	DCNTD _{xy} Clock Select These bits select the DCNTD _{xy} counting-down clock from clock-bus lines 0 to 5.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Timer control registers Dx (TCRDx) are 8-bit or 16-bit readable/writable registers that select the counter clocks in subblock Dx for 32-bit timer counter 1 (TCNT1Dx), 32-bit timer counter 2 (TCNT2Dx), and 32-bit timer down counter (DCNTDxy) from clock-bus lines 0 to 5. These registers also enable and disable capture of the timer offset base register, counter clearing requests from timer B for TCNT1Dx and TCNT2Dx, and clearing timer counters 1D (TCNT1Dx) and 2D (TCNT2Dx).

Offset Base Register Enable (OBREDx)

This register enables and disables input capture to offset base register (OSBRDx). When this bit is set to 1, the value in TCNT1Dx is captured by OSBRDx in the PCLK cycles following cycles in which the input signal selected by the TICTSELDx register is asserted. When the pulse width of the event 2A or 2B signal from timer A exceeds one cycle of the PCLK, the counter value is captured every clock cycle.

OBREDx	Function	
0	Input capture by OSBRDx is enabled	(Default)
1	Input capture by OSBRDx is disabled	

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Counter 2 Clear Enable (C2CEDx)

Enables and disables clearing the counter value in TCNT2Dx by timer B.

When an edge of the counter clearing signal output from timer B is detected while this bit is set to 1, TCNT2Dx is cleared (0000 0000_H) in the following timing.

When the rising edge of the clearing signal is detected in the cycle in which TCNT2Dx is counted up, the counter is cleared on the counting up timing. When the rising edge of the clearing signal is detected in other than the TCNT2Dx counting-up cycle, the counter is cleared on the first counting up timing after edge detection. In other words, the request for clearing remains in place from the rising edge of the counter-clearing signal from timer B until completion of the first clock cycle to drive counting up. Even if C2CEDx is changed to 0 during this period of retention, and even if timer D is disabled and the clock is stopped, the request for clearing remains in place.

The counter clearing signal is ignored with this bit set to the initial value.

C2CEDx	Function	
0	TCNT2Dx clearing signal from timer B is disabled	(Default)
1	TCNT2Dx clearing signal from timer B is enabled	

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1.

Counter 1 Clear Enable (C1CEDx)

Enables and disables clearing the value in TCNT1Dx by timer B.

When an edge of the counter clearing signal output from timer B is detected while this bit is set to 1, TCNT1Dx is cleared (0000 0000_H) in the following timing.

When the rising edge of the clearing signal is detected in the cycle in which TCNT1Dx is counted up, the counter is cleared on the counting up timing. When the rising edge of the clearing signal is detected in other than the TCNT1Dx counting-up cycle, the counter is cleared on the first counting up timing after edge detection. In other words, the clear request is held from the rising of the counter clear signal from timer B to the first count-up clock. The clear request is held during this period, even if C1CEDx is set to "0" or if timer D is disabled and the clock is stopped.

The counter clearing signal is ignored with this bit set to the initial value.

C1CEDx	Function	
0	TCNT1Dx clearing signal from timer B is disabled	(Default)
1	TCNT1Dx clearing signal from timer B is enabled	

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

TCNT2Dx Clear 2Dx (CLR2Dx)

Setting the TCNT2Dx clear 2Dx bit to 1 clears (0000 0000_H) the TCNT2Dx value at the first clock after a compare match between timer counter 2Dx (TCNT2D) and counter upper-limit setting compare register 2Dx (CUCR2Dx). During count operation, a compare match occurs and continues counting even after it is cleared.

CLR2Dx	Function
0	CUCR2Dx compare match is not used to clear the TCNT2Dx value. (Default)
1	CUCR2Dx compare match is used to clear the TCNT2Dx value.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

TCNT2Dx Clock Select (CKSEL2Dx[2:0])

These bits select the TCNT2Dx counting-up clock.

CKSEL2Dx			Function
[2]	[1]	[0]	
0	0	0	Incrementation of TCNT2Dx is driven by clock-bus line 0. (Default)
0	0	1	Incrementation of TCNT2Dx is driven by clock-bus line 1.
0	1	0	Incrementation of TCNT2Dx is driven by clock-bus line 2.
0	1	1	Incrementation of TCNT2Dx is driven by clock-bus line 3.
1	0	0	Incrementation of TCNT2Dx is driven by clock-bus line 4.
1	0	1	Incrementation of TCNT2Dx is driven by clock-bus line 5.
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

TCNT1Dx Clear 1Dx (CLR1Dx)

The TCNT1Dx value is cleared to 0000 0000_H by setting the TCNT1Dx clear 1Dx bit to 1 at the first clock after a compare match between timer counter 1Dx (TCNT1Dx) and counter upper-limit setting compare register 1Dx (CUCR1Dx). During count operation, a compare match occurs and continues counting even after it is cleared.

CLR1Dx	Function
0	CUCR1Dx compare match is not used to clear the TCNT1Dx value. (Default)
1	CUCR1Dx compare match is used to clear the TCNT1Dx value.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

TCNT1Dx Clock Select (CKSEL1Dx[2:0])

These bits select the TCNT1Dx counting-up clock.

CKSEL1Dx			Function
[2]	[1]	[0]	
0	0	0	Incrementation of TCNT1Dx is driven by clock-bus line 0. (Default)
0	0	1	Incrementation of TCNT1Dx is driven by clock-bus line 1.
0	1	0	Incrementation of TCNT1Dx is driven by clock-bus line 2.
0	1	1	Incrementation of TCNT1Dx is driven by clock-bus line 3.
1	0	0	Incrementation of TCNT1Dx is driven by clock-bus line 4.
1	0	1	Incrementation of TCNT1Dx is driven by clock-bus line 5.
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

DCNTDxy Clock Select (DCSEL Dx[2:0])

These bits select the clock signal to drive counting down by the 32-bit down counter (DCNTDxy) of the Dx sub-block.

DCSEL Dx			Function
[2]	[1]	[0]	
0	0	0	Decrementation of DCNTDxy is driven by clock-bus line 0. (Default)
0	0	1	Decrementation of DCNTDxy is driven by clock-bus line 1.
0	1	0	Decrementation of DCNTDxy is driven by clock-bus line 2.
0	1	1	Decrementation of DCNTDxy is driven by clock-bus line 3.
1	0	0	Decrementation of DCNTDxy is driven by clock-bus line 4.
1	0	1	Decrementation of DCNTDxy is driven by clock-bus line 5.
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Note 1: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Note 2: y = 0, 1, 2, 3: Corresponding to channels 0 to 3 in subblock Dx. Channels 0 to 3 of a subblock share a common counter clock.

Output of one-shot pulse (TODxyB) is synchronized with the clock selected in these bits.

30.8.2.3 TCCRLRDx — Timer Compare Control Register Dx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CLRSE L2Dx	CLRSE L1Dx	CMPSE L2Dx3	CMPSE L2Dx2	CMPSE L2Dx1	CMPSE L2Dx0	CMPSE L1Dx3	CMPSE L1Dx2	CMPSE L1Dx1	CMPSE L1Dx0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.124 TCCRLRDx Register Contents

Bit Position	Bit Name	Function
15 to 10	—	These bits are not used. When these bits are read, “0” is always returned. When writing, always write “0”.
9	CLRSEL2Dx	TCNT2Dx clear timing select bit Select the timing for clearing TCNT2Dx. 0: Clear in sync with PCLK (equivalent to ATU-IV) 1: Clear in sync with the count clock
8	CLRSEL1Dx	TCNT1Dx clear timing select bit Select the timing for clearing TCNT1Dx. 0: Clear in sync with PCLK (equivalent to ATU-IV) 1: Clear in sync with the count clock
7 to 4	CMPSEL2Dxy	Compare match B select bit Select the timer counter to be compared with OCR2Dxy. (Select counter B) 0: Compare with TCNT2Dx 1: Compare with TCNT1Dx
3 to 0	CMPSEL1Dxy	Compare match A select bit Select the timer counter to be compared with OCR1Dxy. (Select counter A) 0: Compare with TCNT1Dx 1: Compare with TCNT2Dx

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The timer compare control register Dx (TCCRLRDx) is an 8-bit/16-bit readable/writable register.

This register specifies the timing for clearing the timer counters 1Dx/2Dx (TCNT1Dx and TCNT2Dx) and the timer counters (TCNT1Dx and TCNT2Dx) to be compared with the output compare registers 1Dxy/2Dxy (OCR1Dxy and OCR2Dxy).

TCCRLRDx is initialized to 0000_H after reset.

TCNT2Dx Clear Timing Select Bit (CLRSEL2Dx)

Selects the timing for clearing the timer counter 2Dx (TCNT2Dx) after occurrence of compare match between the timer counter 2Dx (TCNT2Dx) and the counter upper limit value setting compare register (CUCR2Dx).

CLRSEL2Dx	Function	
0	Clear in sync with PCLK (equivalent to ATU-IV)	(Default)
1	Clear in sync with the count clock	

Note: x = 0 to SB_D-1. Corresponding to subblocks D0 to DSB_D-1.

If CLRSEL2Dx = 0, the cleared value of the timer counter 2Dx (TCNT2Dx) after occurrence of compare match between TCNT2Dx and the counter upper limit value setting compare register (CUCR2Dx) may differ. TCNT2Dx is cleared to 0 when the division ratio of the count clock is 2 or greater, while it is cleared to 1 when the division ratio is 1. If CLRSEL2Dx = 1, TCNT2Dx is cleared to 0 when spite of the division ratio of the count clock is 1.

TCNT1Dx Clear Timing Select Bit (CLRSEL1Dx)

Selects the timing for clearing the timer counter 1Dx (TCNT1Dx) after occurrence of compare match between the timer counter 1Dx (TCNT1Dx) and the counter upper limit value setting compare register (CUCR1Dx).

CLRSEL1Dx	Function	
0	Clear in sync with PCLK (equivalent to ATU-IV)	(Default)
1	Clear in sync with the count clock	

Note: x = 0 to SB_D-1. Corresponding to subblocks D0 to DSB_D-1.

If CLRSEL1Dx = 0, the cleared value of the timer counter 1Dx (TCNT1Dx) after occurrence of compare match between TCNT1Dx and the counter upper limit value setting compare register (CUCR1Dx) may differ. TCNT1Dx is cleared to 0 when the division ratio of the count clock is 2 or greater, while it is cleared to 1 when the division ratio is 1. If CLRSEL1Dx = 1, TCNT1Dx is cleared to 0 when spite of the division ratio of the count clock is 1.

Compare Match B Select Bit (CMPSEL2Dxy)

Selects the timer counter (TCNT1Dx or TCNT2Dx) to be compared with the output compare register 2Dxy (OCR2Dxy).

CMPSEL2Dxy	Function	
0	Compare with TCNT2Dx	(Default)
1	Compare with TCNT1Dx	

Note: x = 0 to SB_D-1. Corresponding to subblocks D0 to DSB_D-1.

Compare Match A Select Bit (CMPSEL1Dxy)

Selects the timer counter (TCNT1Dx or TCNT2Dx) to be compared with the output compare register 1Dx (OCR1Dx).

CMPSEL1Dxy	Function	
0	Compare with TCNT1Dx	(Default)
1	Compare with TCNT2Dx	

Note: x = 0 to SB_D-1. Corresponding to subblocks D0 to DSB_D-1.

30.8.2.4 TIOR1Dx — Timer I/O Control Registers 1Dx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSSDx3[1:0]		OSSDx2[1:0]		OSSDx1[1:0]		OSSDx0[1:0]		IOADx3[1:0]		IOADx2[1:0]		IOADx1[1:0]		IOADx0[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.125 TIOR1Dx Register Contents

Bit Position	Bit Name	Function
15 to 8	OSSDx3-Dx0	Compare match output source select These bits select the output level on the compare match output pin (TODxyA). The output is controlled by compare match A, compare match B, or both matches.
7 to 0	IOADx3-IOADx0	I/O control A These bits select the function of the output compare register (OCR1Dxy) on compare match A.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

The timer I/O control registers 1Dx (TIOR1Dx) are 8-bit/16-bit readable/writable registers.

TIOR1Dx selects the source of compare match output (TODxyA), enable and disable compare match of OCRDxy, and set the output level on pin TODxA on compare match A.

TIOR1Dx is initialized to 0000_H after reset.

Compare Match Output Source Select (OSSDxy[1:0])

OSSDxy		Function
[1]	[0]	
0	0	No TODxyA pin output (Default)
0	1	Output level on TODxyA depends on the I/O control bit A on compare match A
1	0	Output level on TODxyA depends on the I/O control bit B on compare match B
1	1	Output level on TODxyA depends on the I/O control bit A or B on compare match A respectively

Note: x = 0 to SB_D-1, y = 0, 1, 2, 3

These bits select the output level on the compare match output pin (TODxyA). The output is controlled by compare match A, compare match B, or both matches.

Either of compare matches A and B is used as a start/stop trigger of timer down counter D (DCNTDxy), and the other one is used as a trigger of compare match output. In addition, both of compare matches A and B are used to control the output, with one as a trigger of an assertion of the output and the other as a trigger of a negation of the output, enabling a one-shot pulse to be output from pin TODxyA.

When both matches are used as a trigger of the output and they occur at the same time, priority is given to compare match B and the output level depends on the IOBxy bit in TIOR2Dx.

A level of 0 is output on TODxyA as a default. While these bits are set to “00”, the output level on TODxyA is not changed even if compare match A or B occurs.

I/O Control A (IOADxy[1:0])

IOADxy		Function
[1]	[0]	
0	0	Compare match A is disabled (Default)
0	1	Output level on compare match A is 0
1	0	Output level on compare match A is 1
1	1	Output level on compare match A is toggled

Note: x = 0 to SB_D-1, y = 0, 1, 2, 3

These bits select the function of the output compare register (OCR1Dxy). When these bits are set to 00B, compare match between OCR1Dxy and timer counter (TCNT1Dx or TCNT2Dx) is not performed. Otherwise, the compare match is performed. An interrupt request is issued on compare match.

When a compare match occurs, if the compare match A is selected as the output source by the compare match output source select bit (OSSDxy) or by the timer compare match control register (TCMPED) and timer I/O control register (TIORDx), the compare match output value set by IOADxy is output to the TODxyA pin (in subblocks D0 to DSB_D-1). See the tables below for correspondence between the I/O control bit A (IOADxy), timer compare match control register (TCMPED), and timer I/O control register (TIORDx).

<<TCMPED and TIORDx when TIOR1Dx is written>>

Written to:		Applied to:		Operation
TIOR1Dx.IOADxy[1:0]		TCMPED.CMPE1Dx	TIORDx.IO1Dxy[1:0]	
00		0	Value held	Compare match is disabled
01		1	01	Output level on compare match is 0
10		1	10	Output level on compare match is 1
11		1	11	Output level on compare match is toggled

<<TIOR1Dx when TCMPED or TIORDx is written>>

Written to:		Applied to:		Operation
TCMPED.CMPE1Dx	TIORDx.IO1Dxy[1:0]	TIOR1Dx.IOADxy[1:0]		
0	Xx *1	00		Compare match is disabled
1	00 *2	11		Output level on compare match is toggled
1	01	01		Output level on compare match is 0
1	10	10		Output level on compare match is 1
1	11	11		Output level on compare match is toggled

Note 1. The written value is held. "11" is returned when read.

Note 2. If "00" is attempted to be written, the writing in value is "00", but the read value is "11".

30.8.2.5 TIOR2Dx — Timer I/O Control Register 2Dx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	IOBDx3[2:0]			—	IOBDx2[2:0]			—	IOBDx1[2:0]			—	IOBDx0[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.126 TIOR2Dx Register Contents

Bit Position	Bit Name	Function
15, 11 7, 3	—	These bits are not used. When these bits are read, “0” is always returned. When writing, always write “0”.
14-12, 10-8 6-4, 2-0	IOBDx3-IOBDx0	I/O control B Sets permission/prohibition of the output compare register (OCR2Dxy), and the compare match B output value.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The timer I/O control register 2Dx (TIOR2Dx) is an 8-bit/16-bit readable/writable register.

This register can function as capture or output compare registers (OCR2Dxy). TIOR2Dx also enables and disables compare match and set the output level on pin TODxA on compare match.

TIOR2Dx is initialized to 0000_H after reset.

I/O Control B (IOBDxy[2:0])

IOBDxy			Function	
[2]	[1]	[0]		
0	0	0	Selects compare match output of OCR2Dxy.	Compare match B is disabled (Default)
0	0	1		Output level on compare match B is 0
0	1	0		Output level on compare match B is 1
0	1	1		Output level on compare match B is toggled
1	0	0	Setting prohibited	Setting prohibited
1	0	1		Setting prohibited
1	1	0		Setting prohibited
1	1	1		Setting prohibited

Note: x = 0 to SB_D-1, y = 0, 1, 2, 3

These bits enable and disable of compare match of the output compare register 2Dxy (OCR2Dxy) and select the TODxA pin output value on the compare match B.

When TIOR2Dx is used as the compare match register (IOBDxy[2] = 0) and the IOBDxy[1:0] bits are set to 00B, compare match between GRDxy and timer counter (TCNT1Dx or TCNT2Dx) is not performed. Otherwise, the compare match is performed. At this time, an interrupt request is issued to CPU by compare match.

When a compare match occurs, if the compare match B is selected as the output source by the compare match output source select bit (OSSDxy), the compare match output value set by IOBDxy or by the timer compare match control register (TCMPED) and the timer I/O control register (TIORDx) is output to the TODxyA pin. See the tables below for correspondence between the I/O control bit B (IOBDxy), timer compare match control register (TCMPED), and timer I/O control register (TIORDx).

<<TCMPED and TIORDx when TIOR2Dx is written>>

Written to:	Applied to:		Operation
	TCMPED.CMPE2Dx	TIORDx.IO2Dxy[1:0]	
TIOR2Dx.IOBDxy[2:0]			
000	0	Value held	Compare match is disabled
001	1	01	Output level on compare match is 0
010	1	10	Output level on compare match is 1
011	1	11	Output level on compare match is toggled
1xx *1	0	Value held	Compare match is disabled

Note 1. "1xx" Indicates 100, 101, 110, 111.

<<TIOR2Dx when TCMPED or TIORDx is written>>

Written to:		Applied to:		Operation
TCMPED.CMPE2Dx	TIORDx.IO2Dxy[1:0]	TIOR2Dx.IOBDxy[2:0]		
0	xx *1	000		Compare match is disabled
1	00 *2	011		Output level on compare match is toggled
1	01	001		Output level on compare match is 0
1	10	010		Output level on compare match is 1
1	11	011		Output level on compare match is toggled

Note 1. The written value is held. If "00" is attempted to be written, "11" is returned when read.

Note 2. If "00" is attempted to be written, the writing in value is "00", but the read value is "11".

30.8.2.6 TCMPEd_x — Timer Compare Match Control Register D_x

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CMPE2D _x 3	CMPE2D _x 2	CMPE2D _x 1	CMPE2D _x 0	CMPE1D _x 3	CMPE1D _x 2	CMPE1D _x 1	CMPE1D _x 0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.127 TCMPEd_x Register Contents

Bit Position	Bit Name	Function
7 to 4	CMPE2D _x 3-D _x 0	Select counter B compare match control bit Controls compare match with Select counter B 0: Disable compare match 1: Enable compare match
3 to 0	CMPE1D _x 3-D _x 0	Select counter A compare match control bit Controls compare match with Select counter A 0: Disable compare match 1: Enable compare match

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The timer compare match control register D_x (TCMPED_x) is an 8-bit readable/writable register.

This register controls the compare match with TCNT1D_x and TCNT2D_x. The output compare register selected by TCCRLRD_x.CMPSEL1D_{xy} and TCCRLRD_x.CMPSEL2D_{xy} will be compared. The setting in this register is reflected to the TIOR1D_x and TIOR2D_x registers. The setting in the TIOR1D_x and TIOR2D_x registers is reflected to this register. For details about how the setting is reflected, see the description about the I/O control bit A (IOAD_{xy}[1:0]) of the timer I/O control register 1D_x (TIOR1D_x) and the I/O control bit B (IOBD_{xy}[2:0]) of the timer I/O control register 2D_x (TIOR2D_x).

TCMPED_x is initialized to 00_H after reset.

30.8.2.7 TIORDx — Timer I/O Control Register Dx

Value after reset: FFFF_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IO2Dx3[1:0]		IO2Dx2[1:0]		IO2Dx1[1:0]		IO2Dx0[1:0]		IO1Dx3[1:0]		IO1Dx2[1:0]		IO1Dx1[1:0]		IO1Dx0[1:0]	
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.128 TIORDx Register Contents

Bit Position	Bit Name	Function
15 to 8	IO2Dx3 to Dx0	TCNT2D compare match output value Controls the TCNT2Dx compare match output 00: Setting prohibited 01: Output 0 10: Output 1 11: Toggle output
7 to 0	IO1Dx3-Dx0	TCNT1D compare match output value Controls the TCNT1Dx compare match output 00: Setting prohibited 01: Output 0 10: Output 1 11: Toggle output

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The timer I/O control register Dx (TIORDx) is an 8-bit/16-bit readable/writable register.

This register control the TCNT1Dx and TCNT2Dx compare match output values. The setting of this register is valid when compare match is enabled by the TCMPEd_x register. When compare match is enabled by the TCMPEd_x register, the setting in this register is reflected to the TIOR1Dx and TIOR2Dx registers. The setting in the TIOR1Dx and TIOR2Dx registers is reflected to this register. For details about how the setting is reflected, see the description about the I/O control bit A (IOADxy[1:0]) of the timer I/O control register 1Dx (TIOR1Dx) and the I/O control bit B (IOBDxy[2:0]) of the timer I/O control register 2Dx (TIOR2Dx).

TIORDx is initialized to FFFF_H after reset.

30.8.2.8 OSELRDx — Output Select Register Dx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	OSELBDx3	OSELBDx2	OSELBDx1	OSELBDx0	OSELADx3	OSELADx2	OSELADx1	OSELADx0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.129 OSELRDx Register Contents

Bit Position	Bit Name	Function
7 to 4	OSELBDxy	Output selection B 0: TODxyB provides normal output. 1: TODxyB provides the value of the output value register.
3 to 0	OSELADxy	Output selection A 0: TODxyA provides normal output. 1: TODxyA provides the value of the output value register.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The output selection register Dx (OSELRDx) is an 8-bit readable/writable register.

It determines whether to output the value of output value register from the pins (TODxyA and TODxyB).

OSELRDx is initialized to 00_H after reset.

Output Selection Bits A (B) xy (OSEL A (B) Dxy)

Writing 1 to this bit outputs the value of the output value register to TODxyA (or TODxyB) pin. Setting of the timer I/O control register D (TIOR1Dx and TIOR2Dx), setting of the timer output control register D (TOCRDx), or one-shot pulse output by the timer down-counter Dxy has no effect on the output value.

When this bit is set to 0, the TODxyA pin outputs according to the settings of the timer I/O control register D (TIOR1Dx and TIOR2Dx), and settings of the one-shot pulse output by the timer output control register D (TOCRDx).

A glitch may be generated if a change in normal output on TODxyA (or TODxyB) coincides with the setting of this bit. Do not write a new value to the bit if the counter value is in the vicinity of that which leads to a match in comparison.

30.8.2.9 ODRDx — Output Value Register Dx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	ODBDx3	ODBDx2	ODBDx1	ODBDx0	ODADx3	ODADx2	ODADx1	ODADx0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.130 ODRDx Register Contents

Bit Position	Bit Name	Function
7 to 4	ODBDxy	Output value B Sets the value to be output to TODxyB.
3 to 0	ODADxy	Output value A Sets the value to be output to TODxyA.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The output value register Dx (ODRDx) is an 8-bit readable/writable register.

When a bit of the output selection register Dx (OSELRDx) is set to 1, the corresponding ODRDx value is issued from respective pins (TODxyA and TODxyB).

ODRDx is initialized to 00_H after reset.

30.8.2.10 DSTRDx — Down Counter Start Register Dx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DSTDx3	DSTDx2	DSTDx1	DSTDx0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.131 DSTRDx Register Contents

Bit Position	Bit Name	Function
7 to 4	—	These bits are not used. When these bits are read, “0” is always returned. When writing, always write “0”.
3 to 0	DSTDxy*1	Down Counter Start 0: No operation 1: Timer down counters (DCNTDx3-Dx0) are started

Note 1. Value 0 cannot be written. The value is not retained when 1 is written. These bits are always read as 0.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The down counter starting register Dx (DSTRDx) is an 8-bit writable register.

This register has a start bit for the down counter. Writing “1” to a bit starts down counting of the corresponding timer.

DSTRDx is initialized to 00_H after reset.

Down Counter Start Dxy (DSTDxy)

Setting these bits to 1 makes timer down counter Dxy (DCNTDxy) start. The setting in these bits is always valid and regardless of the start trigger setting in the down counter control register. When compare match B and writing 1 to these bits occurs at the same time if the down counter is set so that it is stopped on compare match B (compare match A or B), compare match B (compare match A or B) takes priority and the down counter is not started.

When DCNTDxy = 0000 0000_H, writing 1 to these bits has no effect. The DSFDxy bit in the down counter status register 1Dx (DSR1Dx) is retained to “1” until the next down counter clock is input. See the description about DSR1Dx described later.

30.8.2.11 DSR1Dx — Down Counter Status Registers 1Dx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	DWFDx3	DWFDx2	DWFDx1	DWFDx0	DSFDx3	DSFDx2	DSFDx1	DSFDx0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 30.132 DSR1Dx Register Contents

Bit Position	Bit Name	Function
7 to 4	DWFDxy	Down count wait flag 0: Down count compare match start wait status has not occurred. 1: Down count compare match start wait status has occurred.
3 to 0	DSFDxy	Down counter status flag 0: Down counter is disabled 1: Down counter is enabled

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The down count status register 1Dx (DSR1Dx) is an 8-bit read-only register. This register has a flag that indicates the status of the timer down counter (DCNTDxy).

DSR1Dx is initialized to 00_H after reset.

Down Count Wait Flag Dxy (DWFDxy)

These bits are flags that indicate whether timer down count Dxy (DCNTDxy) is in the count start wait status. This flag is valid only when compare match A and B is set for the count start trigger by TRGSELDxy bits in the down counter control register (DCRDx) at DCRDx.TRGSELDxy[3:0] = 0110_B. This flag is not set to 1 in other settings.

When compare match A and B is set as a count start trigger and DWFDxy is 0, if compare match A or compare match B occurs, DWFDxy is set to 1. When a compare match (compare match B if DWFDxy is set to 1 on compare match A) of a source different from setting DWFDxy to 1 occurs, it is recognized as a count start trigger and down-counting starts. At this time, the DWFDxy value is cleared to 0.

Even when a compare match of the same source as setting DWFDxy to 1 occurs, the DWFDxy value remains 1 and is not recognized as a count start trigger.

If compare match A and compare match B occur simultaneously, it is immediately recognized as a count start trigger. Therefore, this flag does not enter the down count wait status and DWFDxy is not set to 1.

This flag is read-only flag and cannot be set to 1 or cleared to 0 by the software.

- Setting (to 1) condition

When compare match A or compare match B has occurred while TRGSELDxy in the down counter control register Dx (DCRDx) is 0110_B

- Clearing (to 0) condition

- When DWFDxy is set to 1 on compare match A and compare match B has occurred while DWFDxy is 1
- When DWFDxy is set to 1 on compare match B and compare match A has occurred while DWFDxy is 1
- Writing 1 to DWFDxy in the down count status clear register (DSCRdx)

Down counter status flag Dxy (DSFDxy)

These bits indicate enabling/disabling of timer down counter Dxy (DCNTDxy). When these bits are read as 1, the counter operation is enabled. If the TDE bit of ATUENR is 1 and the STRDx bit of TSTRD is 1, the counter is running. When these bits are read as 0, the counter operation is disabled, so counting cannot be in progress.

This flag is a read-only flag; it cannot be set to 1 or cleared to 0 by software.

When the set/clear condition competed, 0 clear condition is given priority, and DSFDxy will be 0.

- Setting (to 1) condition
 - When writing 1 to the down count start bit in the down count starting register (DSTRDx)
 - When the condition set as the down counter start trigger is satisfied (compare-match A, compare-match B, compare-match A or B, compare-match A and B)
- Clearing (to 0) condition
 - When the down counter is stopped by underflow
 - When the condition to stop the down counter (compare match B, compare-match A or B)

These flags are set regardless of the settings of the TDE bit in ATUENR and the STRDx bit in TSTRD. Accordingly, if the TDE bit and the STRDn bit are not set to enable counting, the down counter is not actually running even if these bits indicate that counting is enabled. Clearing proceeds in synchronization with the down-count clock.

CAUTION

When the value of down counter DCNTDxy is 0000 0000_H, a trigger to start counting down by the counter being produced by writing 1 to DSTDxy, the bit that starts counting down by the counter, or by a match in compare-match A or B occurring, counting down will not start, although the value of the above counting down status flag (DSFDxy) becomes "1" over the period where the clock input to the down counter is "0". The value of DSFDxy returns to "0" when the clock input to the down counter becomes "1". If a start trigger is issued through the down-counting start bit or by compare-match A, B over the period where the clock input to the down counter is "1", the value of DSFDxy will remain "0".

30.8.2.12 DSR2Dx — Down Count Status Register 2Dx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	DWRFDx3[1:0]		DWRFDx2[1:0]		DWRFDx1[1:0]		DWRFDx0[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.133 DSR2Dx Register Contents

Bit Position	Bit Name	Function
7 to 0	DWRFDxy[1:0]	00: No compare match 01: Down-counting starts on compare match A. 10: Down-counting starts on compare match B. 11: Down-counting starts on simultaneous compare match A and compare match B.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Down count status registers 2Dx (DSR2Dx) are 8-bit read-only registers.

The DWRFDxy register is a flag that indicates the condition for starting count operation of the timer down counter (DCNTDxy).

The status is updated when the timer down counter (DCNTDxy) starts counting.

When the down count start condition is “A and B”, the down count start (compare match occurring later) condition is retained in the status register. When down-counting is started by the down count start register Dx (DSTRDx), the status register keeps retaining the previous value.

DSR2Dx is initialized to 00_H after reset.

- The condition of the renewal

When a start factor of the down count set as DCRDx.TRGSELDxy (besides setting prohibited) occurred, it's renewed. Further, the condition of DCRDx.TRGSELDxy that DWRFDxy = 11_B is set is only when comparing match A,B occurred at the same time when establishing one of 0110_B, 1010_B and 1011_B.

When writing 1 in DWRFCDxy of a down count status clear register (DSCRDx), it'll be 00_B.

- The condition of the state maintenance

When a down count has been begun by DSTRDx, the state is maintained.

The state is maintained in DSR1Dx.DSFDxy = 1 (A down count is moving.)

When DCRDx.TRGSELDxy is set as setting prohibited, the previous value is maintained.

30.8.2.13 DSCR_{Dx} — Down Count Status Clear Register Dx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	DWFC _{Dx3}	DWFC _{Dx2}	DWFC _{Dx1}	DWFC _{Dx0}	DWRFC _{Dx3}	DWRFC _{Dx2}	DWRFC _{Dx1}	DWRFC _{Dx0}
Value after reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.134 DSCR_{Dx} Register Contents

Bit Position	Bit Name	Function
7 to 4	DWFC _{Dxy}	Down count wait flag clear enable 0: Disabled (Default) 1: DWFC _{Dxy} in the down count status register 1 (DSR1D _x) is cleared to 0.
3 to 0	DWRFC _{Dxy}	Down count wait Record flag clear enable 0: Disabled (Default) 1: DWRFC _{Dxy} in the down count status register 2 (DSR2D _x) is cleared to 0.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The down count status clear register (DSCR_{Dx}) is an 8-bit readable/writable register that sets clearing of the flag for the down count status registers (DSR1D_x and DSR2D_x).

DSCR_{Dx} is initialized to 00_H after reset.

Down Count Wait Flag Clear Enable Dxy (DWFC_{Dxy})

These bits set clearing the flag that is set by occurrence of down count wait of the down count status register (DSR1D_x).

Setting each of these bits to 1 clears the down count wait flag (DWFC_{Dxy}) in the down count status register (DSR1D_x). These bits are always read as 0.

When DWFC_{Dxy} in down counter status registers 1D_x (DSR1D_x) sets “1” as this bit in case of 1, the down count WAIT status is cleared.

When “1” is set as this bit before compare match A occurs, and compare match B occurs, it’s cleared that compare match A occurred.

It’ll be similar movement until compare match B occurs, and compare match A occurs.

Down Count Wait Record Flag Clear Enable Dxy (DWRFC_{Dxy})

These bits set clearing of the flag that is set by occurrence of down count wait of the down count status register (DSR2D_x).

Setting each of these bits to 1 clears the down count wait record flag (DWRFC_{Dxy}) in the down count status register (DSR2D_x). These bits are always read as 0.

30.8.2.14 DCRDx — Down Counter Control Registers Dx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	TRGSELDx3				TRGSELDx2				TRGSELDx1				TRGSELDx0				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.135 DCRDx Register Contents

Bit Position	Bit Name	Function
15-12, 11-8	TRGSELDx3-Dx0	Down Counter Start/Stop Trigger Select
7-4, 3-0		Compare match A and compare match B can be set as the start or stop condition for counting of the timer down counter (DCNTDxy).

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Down counter control registers (DCRDx) are 8-bit/16-bit readable/writable registers.

DCRDx starts the timer down counter (DCNTDxy). Besides the down count start by DSTRDx and the down count stop condition when DCNTDx counted a down, and it was 0000_0000_H, Compare match A, compare match B, compare match A or B, or compare match A and B can be set as a starting or stopping trigger of timer down counters (DCNTDxy).

DCRDx is initialized to 0000_H after reset.

To change the TRGSELDx bit, stop the counter. Otherwise operation cannot be guaranteed.

Down Counter Start/Stop Trigger Select Dxy (TRGSELDxy)

TRGSELDxy				Function		
[3]	[2]	[1]	[0]	Counter start trigger	Counter stop trigger	
0	0	0	0	No trigger	No trigger	(Default)
0	0	0	1	No trigger	Compare match B	
0	0	1	0	Compare match A	No trigger	
0	0	1	1	Compare match A	Compare match B	
0	1	0	0	Compare match B	No trigger	
0	1	0	1	Setting prohibited		
0	1	1	0	Compare match A and B	No trigger	
0	1	1	1	Setting prohibited		
1	0	0	0	No trigger	No trigger	
1	0	0	1	No trigger	Compare match A or B	
1	0	1	0	Compare match A or B	No trigger	
1	0	1	1	Compare match A or B	Compare match A or B	
1	1	0	0	Setting prohibited		
1	1	0	1	Setting prohibited		
1	1	1	0	Setting prohibited		
1	1	1	1	Setting prohibited		

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3. “No trigger” of a count start shows a down count start by DSTRDx, and DCNTDxy counts a down, and “No trigger” of a count stop shows the case which became 0000_0000_H. These factors are relevant to everything’s condition besides the setting prohibited.

Besides the down count start by DSTRDx and the down count stop condition when DCNTDx counted a down, and it was 0000_0000_H, the trigger source to start and stop the timer down counter (DCNTDxy) can be selected by these TRGSELDxy bits. Compare-match A, compare-match B, compare-match A or B, or compare-match A and B can be set as a count start trigger. When it's "No trigger", a down count is begun by DSTRDx. Compare match A and B starts down-counting when both compare match A and compare match B are satisfied. (Simultaneous occurrence of compare match A and compare match B is not necessary.)

Compare-match B, compare-match A or B, or compare-match A and B can be set as a count stop trigger. DCNTDxy counts a down in case of "No trigger", and when it was 0000_0000_H, a trigger of a count stop occurs.

If the source of down counter start trigger and the source of down counter stop trigger occur simultaneously, the down counting is stopped.

A signal from compare-match A or B acts as a stop-trigger and as a start-trigger for counting when the setting of TRGSELDxy is 1011_B. Specifically, a trigger generated while down counting is stopped acts as a start trigger and a trigger generated while down counting is in progress acts as a stop trigger. It can be judged in DSR1Dx.DSFDxy whether a down count is moving.

CAUTION

When an underflow and compare match A occur simultaneously while TRGSELDxy is 1010_B, an underflow interrupt is not generated and the underflow flag is not set to 1. When a count start trigger and a compare match A occur simultaneously when the timer downcounter Dxy (DCNTDxy) stops with 0000_0000_H while TRGSELDxy is 1010_B, an underflow interrupt is not generated and the underflow flag is not set to 1.

In this case, to execute software processing for underflow occurrence, write the underflow interrupt flag to 1 (EICn.EIRFn = 1) only when the down counter is stopped (DSR1Dx.DSFDxy = 0) and the underflow interrupt flag of the interrupt controller is 0 (EICn.EIRFn = 0).

30.8.2.15 TSRD_x — Timer Status Registers D_x

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	OVF2D _x	OVF1D _x	UDFD _{x3}	UDFD _{x2}	UDFD _{x1}	UDFD _{x0}	CMFAD _{x3}	CMFAD _{x2}	CMFAD _{x1}	CMFAD _{x0}	CMFBD _{x3}	CMFBD _{x2}	CMFBD _{x1}	CMFBD _{x0}
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.136 TSRD_x Register Contents

Bit Position	Bit Name	Function
15, 14	—	These bits are not used. When these bits are read, “0” is always returned. When writing, always write “0”.
13	OVF2D _x	Overflow flag 2D _x 0: TCNT2D _x has not overflowed 1: TCNT2D _x has overflowed
12	OVF1D _x	Overflow flag 1D _x 0: TCNT1D _x has not overflowed 1: TCNT1D _x has overflowed
11 to 8	UDFD _{x3} to UDFD _{x0}	Underflow flag D _{x3} to D _{x0} 0: DCNTD _{xy} has not underflowed 1: DCNTD _{xy} has underflowed
7 to 4	CMFAD _{x3} to CMFAD _{x0}	Compare match A flag D _{x3} to D _{x0} 0: Compare match A has not occurred 1: Compare match A has occurred
3 to 0	CMFBD _{x3} to CMFBD _{x0}	Compare match B flag D _{x3} to D _{x0} 0: Compare match B has not occurred 1: Compare match B has occurred

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

The timer status register (TSRD_x) is an 8-bit/16-bit read-only register. This register indicates overflow of two timer counters (TCNT2D_x and TCNT1D_x) of subblock D_x (x = 0 to SB_D-1), underflow of the timer down counters (DCNTD_{xy}) of channels 0 to 3, and compare match of the output compare registers (OCR1D_{xy} and OCR2D_{xy}). If an interrupt source takes effect when this flag is set, an interrupt request is made again. An interrupt request is made even if there is a conflict between clearing by the corresponding timer status clear register and setting by an interrupt source.

Overflow flags 2 and 1 as well as underflow flags and compare match flags A and B can be cleared by setting the corresponding bits of the timer status clear register (TSCRD_x).

TSRD_x is initialized to 0000_H after reset.

Overflow Flag 2Dx (OVF2Dx)

OVF2Dx	Function	
0	[Clearing condition] When 1 is written to the OVFC2Dx bit of the timer status clear register (TSCRDx).	(Default)
1	[Setting condition] When TCNT2Dx overflowed (from FFFF FFFF _H to 0000 0000 _H) When the counter is cleared with the upper-limit setting function enabled	

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Indicates whether or not timer counter 2Dx (TCNT2Dx) has overflowed. If the reading value of this flag is 1, it means that an overflow in TCNT2Dx has occurred. This flag cannot be set to 1.

The overflow flag is set when the count-up clock is input while TCNT2Dx is FFFF FFFF_H or at the first edge of the PCLK after the counter value matches the upper-limit setting register value with the upper-limit setting function enabled. Writing 0000 0000_H to TCNT2Dx or starting TCNT2Dx from an initial value of 0000 0000_H has no effect on this bit.

When writing to TCNT2Dx at the same time as incrementation while it is FFFF FFFF_H, this bit is set to 1. However, TCNT2Dx is updated with the written value, but not 0000 0000_H.

When an assertion of counter clearing signal from timer B and overflow due to count up occur, overflow does not occur. Simultaneous occurrence of counter clearing by timer B and counter clearing with the upper-limit setting function enabled is detected as an overflow.

Overflow Flag 1Dx (OVF1Dx)

OVF1Dx	Function	
0	[Clearing condition] When 1 is written to the OVFC1Dx bit of the timer status clear register (TSCRDx).	(Default)
1	[Setting condition] When TCNT1Dx overflowed (from FFFF FFFF _H to 0000 0000 _H) When the counter is cleared with the upper-limit setting function enabled	

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Indicates whether or not timer counter 1Dx (TCNT1Dx) has overflowed. If the reading value of this flag is 1, it means that an overflow in TCNT1Dx has occurred. This flag cannot be set to 1.

The overflow flag is set when the count-up clock is input while TCNT1Dx is FFFF FFFF_H or at the first edge of the PCLK after the counter value matches the upper-limit setting register value with the upper-limit setting function enabled. Writing 0000 0000_H to TCNT1Dx or starting TCNT1Dx from an initial value of 0000 0000_H has no effect on this bit.

When writing to TCNT1Dn at the same time as incrementation while it is FFFF FFFF_H, this bit is set to 1. However, TCNT1Dx is updated with the written value, but not 0000 0000_H.

When an assertion of counter clearing signal from timer B and an overflow by counting up occur simultaneously, the overflow is not detected. Simultaneous occurrence of counter clearing by timer B and counter clearing with the upper-limit setting function enabled is detected as an overflow.

Underflow Flag Dxy (UDFDxy)

UDFDxy	Function
0	[Clearing condition] When 1 is written to the UDFCDxy bit of the timer status clear register (TSCRDX). (Default)
1	[Setting condition] When DCNTDxy underflowed (decremented while DCNTDxy = 0000 0000 _H)

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

This flag is a status flag that indicates underflow of subblock Dx and down counter Dxy of channel y. If the reading value of this flag is 1, it means that DCNTDxy had an underflow. This flag cannot be set to 1.

This bit is set to 1 when DCNTDxy is to be decremented while it is 0000 0000_H. DCNTDxy holds 0000 0000_H on underflow. Writing FFFF FFFF_H to DCNTDxy has no effect on these bits.

These bits are initialized to 0 after reset. Although DCNTDxy is initialized to 0000 0000_H, these flags do not indicate underflow because DCNTDxy has not started. For details on control of DCNTDxy, see descriptions of timer D down counters.

Compare Match A Flag Dxy (CMFADxy)

CMFADxy	Function
0	[Clearing condition] When 1 is written to the CMFCADxy bit of the timer status clear register (TSCRDX). (Default)
1	[Setting condition] The thing set as all except for 00 _B (comparing match A permission) does a set of TIOR1Dx.IOADxy in the following case by the precondition. <ul style="list-style-type: none"> • When the value of the select counter A (TCNT1Dx or TCNT2Dx) is match with OCR1Dxy. • All except for 00_H is set as the range comparing value setting register 1Dx (RCR1Dx), and when writing in OCR1Dxy, when selection counter A (TCNT1Dx or TCNT2Dx) is in the range set in RCR1Dx.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

CMFADxy is a status flag that indicates occurrence of a compare match in the output compare register (OCR1Dxy) of channel y in subblock Dx. If the reading value of this flag is 1, it means that a compare match occurs between select counter A (TCNT1Dx or TCNT2Dx) and OCR1Dxy or it's meant that it was in the area select counter A (TCNT1Dx or TCNT2Dx) set as the time of write in RCR1Dx or OCR1Dxy. This flag cannot be set to 1.

When operation of compare match between OCRDxy and select counter A (TCNT1Dx or TCNT2Dx) is enabled by the setting in timer I/O control register 1 (TIOR1Dx), compare match operation is performed regardless of the state of select counter A (TCNT1Dx or TCNT2Dx). Or when select counter A (TCNT1Dx or TCNT2Dx) is in the area set in RCR1Dx at the time of write in OCR1Dxy, a compare match A flag is set by the following PCLK cycle. These bits are set to 1 on the first edge of the PCLK after the values in select counter A (TCNT1Dx or TCNT2Dx) and OCR1Dxy match. Even if these compare match flags are cleared to 0 by software while TCNT1Dx = OCR1Dxy after the compare match is detected, these bits are not set to 1 again.

If select counter A (TCNT1Dx or TCNT2Dx) matches OCR1Dxy again before the status flag is cleared, the compare match A is detected and the status flag is rewritten with 1.

Compare match B flag Dxy (CMFBDxy)

CMFBDxy	Function	
0	[Clearing condition] When 1 is written to the CMFCBDxy bit of the timer status clear register (TSCRDX).	(Default)
1	[Setting condition] The thing set as all except for 00 _B (comparing match B permission) does a set of TIOR2Dx.IOADxy in the following case by the precondition. <ul style="list-style-type: none"> • When the value of the select counter B (TCNT1Dx or TCNT2Dx) is match with OCR1Dxy. • All except for 00H is set as the range comparing value setting register 2Dx (RCR2Dx), and when writing in OCR2Dxy, when selection counter B (TCNT1Dx or TCNT2Dx) is in the range set in RCR2Dx. 	

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

CMFBDxy is a status flag that indicates occurrence of a compare match in the output compare B register (OCBRDxy) of channel y in subblock Dx. If the reading value of this flag is 1, it means that a compare match occurs between select counter B (TCNT1Dx or TCNT2Dx) and OCR2Dxy or it's meant that it was in the area select counter B (TCNT1Dx or TCNT2Dx) set as the time of write in RCR2Dx or OCR2Dxy. This flag cannot be set to 1.

When operation of compare match between OCR2Dxy and select counter B (TCNT1Dx or TCNT2Dx) is enabled by the setting in timer I/O control register 2 (TIOR2Dx), compare match operation is performed regardless of the state of select counter B (TCNT1Dx or TCNT2Dx). Or when select counter B (TCNT1Dx or TCNT2Dx) is in the area set in RCR2Dx at the time of write in OCR2Dxy, a compare match B flag is set by the following PCLK cycle. These bits are set to 1 on the first edge of the PCLK after the values in select counter A (TCNT1Dx or TCNT2Dx) and OCR2Dxy match. Even if these compare match flags are cleared to 0 by software while TCNT2Dx = OCR2Dxy after the compare match B is detected, these bits are not set to 1 again.

If select counter B (TCNT1Dx or TCNT2Dx) matches OCR2Dxy again before the status flag is cleared, the compare match B is detected and the status flag is rewritten with 1.

30.8.2.16 TSCRDX — Timer Status Clear Register Dx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	OVFC2 Dx	OVFC1 Dx	UDFCD x3	UDFCD x2	UDFCD x1	UDFCD x0	CMFCA Dx3	CMFCA Dx2	CMFCA Dx1	CMFCA Dx0	CMFCB Dx3	CMFCB Dx2	CMFCB Dx1	CMFCB Dx0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.137 TSCRDX Register Contents

Bit Position	Bit Name	Function
15, 14	—	These bits are not used. When these bits are read, “0” is always returned. When writing, always write “0”.
13	OVFC2Dx	Overflow flag clear enable 2Dx 0: Disabled (Default) 1: Clears OVFC2Dx of timer status register D (TSRDx) to 0.
12	OVFC1Dx	Overflow flag clear enable 1Dx 0: Disabled (Default) 1: Clears OVFC1Dx of timer status register D (TSRDx) to 0.
11 to 8	UDFCDx3 to UDFCDx0	Underflow flag clear enable Dx3 to Dx0 0: Disabled (Default) 1: Clears UDFCDx3-UDFCDx0 of timer status register D (TSRDx) to 0.
7 to 4	CMFCA Dx3 to CMFCA Dx0	Compare match A flag clear enable Dx3 to Dx0 0: Disabled (Default) 1: Clears CMFCA Dx3-CMFCA Dx0 of timer status register D (TSRDx) to 0.
3 to 0	CMFCB Dx3 to CMFCB Dx0	Compare match B flag clear enable Dx3 to Dx0 0: Disabled (Default) 1: Clears CMFCB Dx3-CMFCB Dx0 of timer status register D (TSRDx) to 0.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

The timer status clear register (TSCRDX) is an 8-bit/16-bit writable register. This register specifies the flag clearing at the time of overflow of two 32-bit up-counters (TCNT2Dx and TCNT1Dx), compare match and the output compare register (OCR1Dxy), and compare match and the output compare register (OCR2Dxy).

TSCRDX is initialized to 0000_H after reset.

Overflow Flag Clear Enable 2Dx (OVFC2Dx)

This bit specifies flag clearing at the time of overflow of the timer counter 2Dx (TCNT2Dx).

Setting this bit enables clearing of the overflow flag 2 (OVF2Dx) of the timer status register (TSRDx). This bit is always read as 0.

OVFC2Dx	Function	
0	Disabled	(Default)
1	Clears OVF2Dx to 0.	

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Overflow Flag Clear Enable 1Dx (OVFC1Dx)

This bit specifies flag clearing at the time of overflow of the timer counter 1Dx (TCNT1Dx).

Setting this bit enables clearing of overflow flag 1 (OVF1Dx) of the timer status register (TSRDx). This bit is always read as 0.

OVFC1Dx	Function	
0	Disabled	(Default)
1	Clears OVF1Dx to 0.	

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Underflow Flag Clear Enable Dxy (UDFCDxy)

This bit specifies flag clearing at the time of underflow of the timer down counter Dxy (DCNTDxy).

Setting this bit enables clearing of the underflow flag (UDFDxy) of the timer status register (TSRDx). This bit is always read as 0.

UDFCDxy	Function	
0	Disabled	(Default)
1	Clears UDFDxy to 0.	

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Compare Match A Flag Clear Enable Dxy (CMFCADxy)

This flag specifies flag clearing at the time of compare match of the output compare register (OCR1Dxy).

Setting this bit enables clearing of the compare match A flag (CMFADxy) of the timer status register (TSRDx). This bit is always read as 0.

CMFCADxy	Function	
0	Disabled	(Default)
1	Clears CMFADxy to 0.	

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Compare Match B Flag Clear Enable Dxy (CMFCBDxy)

This flag specifies flag clearing at the time of compare match of the output compare register (OCR2Dxy).

Setting this bit enables clearing of the compare match B flag (CMFBDxy) of the timer status register (TSRDx). This bit is always read as 0.

CMEBDxy	Function	
0	Disabled	(Default)
1	Clears CMFBDxy to 0.	

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

30.8.2.17 TOCRDx — Timer Output Control Registers Dx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TONEBDx	TONEADx
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.138 TOCRDx Register Contents

Bit Position	Bit Name	Function
7 to 2	—	These bits are not used. When these bits are read, “0” is always returned. When writing, always write “0”.
1	TONEBDx	Output inversion select TODxyB 0: Output level is not inverted 1: Output level is inverted
0	TONEADx	Output inversion select TODxyA 0: Output level is not inverted 1: Output level is inverted

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The timer output control register Dx (TOCRDx) is an 8-bit readable/writable register. This register specifies whether to invert signals of the output pins TODxyA and TODxyB of each subblock Dx.

This register is initialized to 00_H after reset.

Output Inversion Select Dx (TONEBDx, TONEADx)

Specifies whether to invert the output from TODxy and TODxyA.

TONEBDx	Function
0	Normal output from the output pin (TODxyB) (Default)
1	Inverted output from the output pin (TODxyB)

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

TONEADx	Function
0	Normal output from the output pin (TODxyA) (Default)
1	Inverted output from the output pin (TODxyA)

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Signals on pins TODxyA and TODxyB are inverted on the first edge of the PCLK after the output inversion select Dx bit is set to 1.

This function is not affected by the operating state of timer counters 1Dx and 2Dx (TCNT1Dx, TCNT2Dx).

Output levels on pins TODxyA and TODxyB are initialized to a level of 0 (when TONEADx = 0 and TONEBDx = 0).

A glitch may be generated if a change in normal output on TODxyA (or TODxyB) coincides with the setting of this bit. Do not write a new value to the bit if the counter value is in the vicinity of that which leads to a match in comparison.

30.8.2.18 TOCCRDx — Timer Output Channel Control Registers Dx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	TONEBDx3	TONEBDx2	TONEBDx1	TONEBDx0	TONEADx3	TONEADx2	TONEADx1	TONEADx0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.139 TOCCRDx Register Contents

Bit Position	Bit Name	Function
7 to 4	TONEBDxy	Channel output inversion select TODxyB 0: The signal on channel y is output from ODxyB as is (without inversion). 1: The signal on channel y is inverted to be output from ODxyB.
3 to 0	TONEADxy	Channel output inversion select TODxyA 0: The signal on channel y is output from ODxyA as is (without inversion). 1: The signal on channel y is inverted to be output from ODxyA.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The timer output channel control register Dx (TOCCRDx) is an 8-bit readable/writable register that specifies whether to invert the signal on each channel before outputting to the TODxyA and TODxyB output pins of each subblock Dx.

This register is initialized to 00_H after reset.

Channel Output Inversion Select Dx (TONEBDx, TONEADx)

Specifies whether to invert the signal on the channel before outputting to TODxyB and TODxyA.

TONEBDxy	Function
0	The signal on channel y is output to TODxyB as is. (Default)
1	The signal on channel y is inverted to be output to TODxyB.

TONEADxy	Function
0	The signal on channel y is output to TODxyA as is. (Default)
1	The signal on channel y is inverted to be output to TODxyA.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

When the channel output inversion select Dx bit is changed, it is reflected to the logical output of TODxyB and TODxyA in the next PCLK cycle. The logical output of TODxyB and TODxyA is controlled by the timer output control register Dx (TOCRDx) and timer output channel control register Dx (TOCCRDx). The operation status (running or stopping) of the timer counters 1Dx and 2Dx (TCNT1Dx and TCNT2Dx) does not affect the switching of output inversion.

Setting for subblock	Setting for channel	
TOCRDx (0: Normal, 1: inversion)	TOCCRDx (0: Normal, 1: Inversion)	Pulse output inversion
Normal output	Normal output	Normal output
	Inversion output	Inversion output
Inversion output	Normal output	
	Inversion output	

30.8.2.19 OSBRD_x — Timer Offset Base Registers D_x

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSBRD _x [31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSBRD _x [15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Timer offset base registers D_x (OSBRD_x) are 32-bit read-only registers that are used only for input capture. The timer counter 1D_x (TCNT1D_x) value is captured and stored by a trigger signal from timer A and PH update notification triggers 0 to 1 from DFE. The trigger signal from timer A is selectable from TIA00 to TIA0SB_A-1, and the external input signals TID0 and TID1 are selectable with the timer input capture trigger select register (TICTSEL_{Dx}). For details, see the description of the timer control registers 1A, 3A, and 4A (TCR1A, TCR3A, TCR4A) and the timer input capture trigger select register (TICTSEL_{Dx}).

This register is initialized to 0000 0000_H after reset.

30.8.2.20 TICTSELDx — Timer Input Capture Trigger Select Register Dx

Value after reset: 0xH

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TIDSELDx[1:0]	
Value after reset	0	0	0	0	0	0	0	*1
R/W	R	R	R	R	R	R	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Note 1. The value varies depending on the channel to maintain compatibility with the traditional products.

x = 0 to 2: 0

x = 3 to 9: 1

The timer input capture trigger select register Dx (TICTSELDx) is an 8-bit readable/writable register. This register is used to select event 2A or 2B from timer A and external input signals PH update notification trigger 0 and PH update notification trigger 1 from the DFE, and to capture the TCNT1Dx counter value in the OSBRDx register.

Table 30.140 TICTSELDx Register Contents

Bit Position	Bit Name	Function
7 to 2	—	These bits are not used. Fix these bits to "0".
1, 0	TIDSELDx[1:0]	00: Event 2A 01: Event 2B 10: PH update notification trigger 0 from the DFE 11: PH update notification trigger 1 from the DFE

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

30.8.2.21 TCNT1Dx — Timer Counter 1Dx

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TCNT1Dx[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNT1Dx[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Timer counters 1Dx (TCNT1Dx) are 32-bit readable/writable registers.

TCNT1Dx is driven by the clock selected in the CKSEL1Dx[2:0] bits in timer control register (TCRDx). These counters are started by setting the bit in timer start register (TSTRD) to 1.

When overflow interrupt request is permitted (TIER1Dx.OV1EDx = 1), an overflow interrupt request can be issued to CPU when the timer overflows.

The overflow flag OV1Dx of the timer status register (TSRDx) is set to 1.

When a clear request from timer B made TCRD.C1CEDx effective, and occurred, TCNT1Dx is cleared by 0000_0000_H. But, when competing with writing in to TCNT1Dx, writing in of TCNT1Dx is given priority.

This register is initialized to 0000 0000_H after reset.

- The upper limit value of timer counter 1Dx (TCNT1Dx)

TCRDx.CLR1Dx	The upper limit value of TCNT1Dx
0	FFFF_FFFF _H
1	CUCR1Dx When the writing in value to CUCR1Dx is smaller than TCNT1Dx, it'll be FFFF_FFFF _H .

- The clear value of TCNT1Dx after reaching the upper limit value

TCRDx.CLR1Dx	TCCR1Dx.CLRSEL1Dx	The count clock chosen in TCRDx.CKSEL1Dx	The clear value of TCNT1Dx
0	0/1	1	0000_0000 _H
1	0	0	0000_0000 _H
		1	0000_0001 _H
1	1	0	State maintenance (=CUCR1Dx value)
		1	0000_0000 _H

30.8.2.22 TCNT2Dx — Timer Counter 2Dx

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TCNT2Dx[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNT2Dx[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Timer counters 2Dx (TCNT2Dx) are 32-bit readable/writable registers.

TCNT2Dx is driven by the clock selected in the CKSEL2Dx[2:0] bits in timer control register Dx (TCRDx). These counters are started by setting the bit in timer start register (TSTRD) to 1.

When overflow interrupt request is permitted (TIER1Dx.OV2EDx = 1), an overflow interrupt request can be issued to CPU when the timer overflows.

The overflow flag OV2Dx of the timer status register (TSRDx) is set to 1.

When a clear request from timer B made TCRD.C2CEDx effective, and occurred, TCNT2Dx is cleared by 0000_0000_H. But, when competing with writing in to TCNT2Dx, writing in of TCNT2Dx is given priority.

This register is initialized to 0000 0000_H after reset.

- The upper limit value of timer counter 2Dx (TCNT2Dx)

TCRDx.CLR2Dx	The upper limit value of TCNT2Dx
0	FFFF_FFFF _H
1	CUCR2Dx When the writing in value to CUCR2Dx is smaller than TCNT2Dx, it'll be FFFF_FFFF _H .

- The clear value of TCNT2Dx after reaching the upper limit value

TCRDx.CLR2Dx	TCCR2Dx.CLRSEL2Dx	The count clock chosen in TCRDx.CKSEL2Dx	The clear value of TCNT2Dx
0	0/1	1	0000_0000 _H
1	0	0	0000_0000 _H
		1	0000_0001 _H
1	1	0	State maintenance (=CUCR2Dx value)
		1	0000_0000 _H

30.8.2.23 CUCR1D_x — Counter Upper-Limit Setting Compare Register 1D_x

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CUCR1D _x [31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CUCR1D _x [15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

The counter upper-limit setting compare register (CUCR1D_x) is a 32-bit readable/writable register that has a function for comparison with the TCNT1D_x counter. Compare match between this register and the TCNT1D_x counter is enabled by setting the TCNT1D_x clear setting bit 1D_x (CLR1D_x) in the timer control register (TCRD_x) to 1. The TCNT1D_x counter value is cleared to 0000 0000_H or 0000 0001_H on compare match between the TCNT1D_x counter and counter upper-limit setting compare register (CUCR1D_x). The clear timing differs depending on TCCRLRD_x.CLRSEL1D_x. When TCCRLRD_x.CLRSEL1D_x = 0, clearing occurs without synchronization with the count clock, and when TCCRLRD_x.CLRSEL1D_x = 1, it clears in synchronization with the count clock. For the clear value and clear timing of TCNT1D_x, refer to the description of TCNT1D_x.

The OVF1D_x bit in timer status register D (TSRD_x) is set to 1 and an overflow interrupt is output on compare match between the TCNT1D_x counter and counter upper-limit setting compare register (CUCR1D_x).

The output width of overflow interrupt is one clock width of PCLK.

Do not set CUCR1D_x to 0000 0000_H. If 0000 0000_H is set, a compare match may occur at an unintended timing.

30.8.2.24 CUCR2D_x — Counter Upper-Limit Setting Compare Register 2D_x

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CUCR2D _x [31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CUCR2D _x [15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

The counter upper-limit setting compare register (CUCR2D_x) is a 32-bit readable/writable register that has a function for comparison with the TCNT2D_x counter. Compare match between this register and the TCNT2D_x counter is enabled by setting the TCNT2D_x clear setting bit 2D_x (CLR2D_x) in the timer control register (TCRD_x) to 1. The TCNT2D_x counter value is cleared to 0000 0000_H or 0000 0001_H on compare match between the TCNT2D_x counter and counter upper-limit setting compare register (CUCR2D_x). The clear timing differs depending on TCCRLRD_x.CLRSEL2D_x. When TCCRLRD_x.CLRSEL2D_x = 0, clearing occurs without synchronization with the count clock, and when TCCRLRD_x.CLRSEL2D_x = 1, it clears in synchronization with the count clock. For the clear value and clear timing of TCNT2D_x, refer to the description of TCNT2D_x.

The OVF2D_x bit in timer status register D (TSRD_x) is set to 1 and an overflow interrupt is output on compare match between the TCNT2D_x counter and counter upper-limit setting compare register (CUCR2D_x).

The output width of overflow interrupt is one clock width of PCLK.

Do not set CUCR2D_x to 0000 0000_H. If 0000 0000_H is set, a compare match may occur at an unintended timing.

30.8.2.25 OCR1Dxy — Output Compare Registers 1Dxy

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OC1Dxy[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OC1Dxy[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The output compare registers (OCR1Dxy) are 32-bit readable/writable registers. OCR1Dxy is constantly compared with the select counter A (TCNT1Dx or TCNT2Dx). When compare match operation is selected by bit IOADxy in TIOR1Dx, the compare match A flag (CMFADxy) in TSRDx is set to 1 on the first edge of the PCLK after the values in select counter A (TCNT1Dx or TCNT2Dx) and OCR1Dxy match. When compare match A is selected by an output source select bit, a signal is output to pin TODxyA on compare match.

When compare match A is selected as a down counter starting trigger by the TRGSELDxy bit in DCRDxy, DCNTDxy is ready to be counted down on compare match A.

When the down counter (DCNTDxy) is ready, it is started in synchronization with the down counter clock. At this time, a one-shot pulse can be output on pin TODxyB. If compare match A and down-counter stop trigger are output at the same time, output is disabled without any pulse.

When TCNT1Dx overflows to change from FFFF FFFF_H to 0000 0000_H and OCRDxy is set to 0000 0000_H, compare match is detected. If the counter upper-limit setting function (see **Section 30.8.3.2 Counter Upper-Limit Setting Function**) is in use with clock-bus line 5 or 6, or with a division ratio of 1/1 selected for the prescaler, the timer counter counts between 1 and the value of CUCR1Dx and the counter value never reaches 0000 0000_H. Therefore, do not set these registers to 0000 0000_H in such cases.

An interrupt can be issued to CPU by a compare match A detection.

When the range compare function is enabled, the select counter A (TCNT1Dx or TCNT2Dx) value at the compare match timing is stored in OCR1Dxy. Regarding the range compare function, see **Section 30.8.3.1 Range Comparison Function**.

This register is initialized to FFFF FFFF_H after reset.

30.8.2.26 RCR1Dx — Range Comparison Value Setting Register 1Dx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	RC1Dx[4:0]				
Value after reset	0*1	0*1	0*1	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Note 1. These bits are not used. Fix these bits to "0".

The range comparison value setting register 1Dx (RCR1Dx) is an 8-bit readable/writable register. This register can enable/disable range compare function and set range value. Regarding the range compare function, see **Section 30.8.3.1 Range Comparison Function**.

RCR1Dx is initialized to 00_H after reset.

The following table shows ranges that can be set as range comparison values (power of 2 (excluded only when RCR1Dx is 0)).

RCR1Dx[4:0]	Selectable Range	Remarks
00000	None	Range comparison disable (Default)
00001	2^1-1	—
00010	2^2-1	—
00011	2^3-1	—
00100	2^4-1	—
00101	2^5-1	—
00110	2^6-1	—
00111	2^7-1	—
01000	2^8-1	—
01001	2^9-1	—
01010	$2^{10}-1$	—
01011	$2^{11}-1$	—
01100	$2^{12}-1$	—
01101	$2^{13}-1$	—
01110	$2^{14}-1$	—
01111	$2^{15}-1$	—
10000	$2^{16}-1$	—
10001	$2^{17}-1$	—
10010	$2^{18}-1$	—
10011	$2^{19}-1$	—
10100	$2^{20}-1$	—
10101	$2^{21}-1$	—
10110	$2^{22}-1$	—
10111	$2^{23}-1$	—
11000	$2^{24}-1$	—
11001	$2^{25}-1$	—
11010	$2^{26}-1$	—
11011	$2^{27}-1$	—
11100	$2^{28}-1$	—
11101	$2^{29}-1$	—
11110	$2^{30}-1$	—
11111	$2^{31}-1$	—

30.8.2.27 OCR2Dxy — Output Compare Registers 2Dxy

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OC2Dxy[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OC2Dxy[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The output compare registers (OCR2Dxy) are 32-bit readable/writable registers. OCR2Dxy is always compared with select counter B (TCNT1Dx or TCNT2Dx), and if the IOBDxy bit of the TIOR2Dx register permits compare match, a match of select counter B (TCNT1Dx or TCNT2Dx) and OCR2Dxy values is detected. If the OSSDxy bit specifies compare match B as the output source, the compare match is output to the TODxyA pin at the next PCLK cycle.

When compare match B is selected as a down counter starting trigger by the TRGSELDxy bit in DCRDx, DCNTDxy is ready to be counted down on compare match B.

When the down counter (DCNTDxy) is ready, it is started in synchronization with the down counter clock. At this time, a one-shot pulse can be output on pin TODxyB. If compare match A and down-counter stop trigger are output at the same time, output is disabled without any pulse.

If the TRGSELDxy bit of DCRDx specifies compare match B as the down count stop trigger, detection of compare match B disables down counting. At the next down-counting clock cycle, the down counter is cleared to 0 and TODxyB output (one-shot pulse) is negated.

When select counter B (TCNT1Dx or TCNT2Dx) overflows to change from FFFF FFFF_H to 0000 0000_H and OCR2Dxy is set to 0000 0000_H, compare match is detected.

If the counter upper-limit setting function (see **Section 30.8.3.2 Counter Upper-Limit Setting Function**) is in use with clock-bus line 5 or 6, or with a division ratio of 1/1 selected for the prescaler, the timer counter counts between 1 and the value of CUCR2Dx and the counter value never reaches 0000 0000_H. Therefore, do not set these registers to 0000 0000_H in such cases.

An interrupt can be issued to CPU by compare match B detection.

When the range compare function is enabled, the select counter B value at the compare match timing is stored in OCR2Dxy. Regarding the range compare function, see **Section 30.8.3.1 Range Comparison Function**.

This register is initialized to FFFF FFFF_H after reset.

30.8.2.28 RCR2Dx — Range Comparison Value Setting Register 2Dx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	RC2Dx[4:0]				
Value after reset	0*1	0*1	0*1	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Note 1. These bits are not used. Fix these bits to "0".

The range comparison value setting register 2Dx (RCR2Dx) is an 8-bit readable/writable register. This register can enable/disable range compare function and set range value. Regarding the range compare function, see **Section 30.8.3.1 Range Comparison Function**.

RCR2Dx is initialized to 00_H after reset.

The following table shows ranges that can be set as range comparison values (power of 2 (excluded only when RCR2Dx is 0)).

RCR2Dx[4:0]	Selectable Range	Remarks
00000	None	Range comparison is invalid (Default)
00001	2^1-1	—
00010	2^2-1	—
00011	2^3-1	—
00100	2^4-1	—
00101	2^5-1	—
00110	2^6-1	—
00111	2^7-1	—
01000	2^8-1	—
01001	2^9-1	—
01010	$2^{10}-1$	—
01011	$2^{11}-1$	—
01100	$2^{12}-1$	—
01101	$2^{13}-1$	—
01110	$2^{14}-1$	—
01111	$2^{15}-1$	—
10000	$2^{16}-1$	—
10001	$2^{17}-1$	—
10010	$2^{18}-1$	—
10011	$2^{19}-1$	—
10100	$2^{20}-1$	—
10101	$2^{21}-1$	—
10110	$2^{22}-1$	—
10111	$2^{23}-1$	—
11000	$2^{24}-1$	—
11001	$2^{25}-1$	—
11010	$2^{26}-1$	—
11011	$2^{27}-1$	—
11100	$2^{28}-1$	—
11101	$2^{29}-1$	—
11110	$2^{30}-1$	—
11111	$2^{31}-1$	—

30.8.2.29 ICR1Dxy — Input Capture Register 1Dxy

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IC1Dxy[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC1Dxy[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The input capture register 1Dxy (ICR1Dxy) is a 32-bit read-only register, and its initial value is 0000 0000_H. This register captures the counter value of select counter A (TCNT1Dx or TCNT2Dx) into ICR1Dxy triggered by compare match B.

ICR1Dxy is initialized to 0000 0000_H after reset.

30.8.2.30 ICR2Dxy — Input capture register 2Dxy

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IC2Dxy[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2Dxy[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The input capture register 2Dxy (ICR2Dxy) is a 32-bit read-only register, and its initial value is 0000 0000_H. This register captures the counter value of select counter B (TCNT1Dx or TCNT2Dx) in ICR2Dxy triggered by compare match A.

ICR2Dxy is initialized to 0000 0000_H after reset.

30.8.2.31 DCNTDxy — Timer Down Counters Dxy

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTDxy[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTDxy[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Timer down counters (DCNTDxy) are 32-bit readable/writable registers.

DCNTDxy is driven by the clock selected in the DCSELDx[2:0] bits in timer control register D (TCRDx).

While the down counter is enabled, it is decremented every input of the down counter clock. Counting can be stopped by an underflow of DCNTDxy, detection of compare match B, or detection of compare match A or B. When the decrementing is stopped by an underflow, a down counter underflow interrupt request and a DMA transfer request can be issued.

Decrementation is enabled; on the first edge of the PCLK (the same as compare match A) after the values in select counter A (TCNT1Dx or TCNT2Dx) and OCRDxy match; on the first edge of the PCLK (the same as compare match B) after the values in select counter B (TCNT1Dx or TCNT2Dx) and OCR2Dxy match; on the first edge of the PCLK after the DSTDxy bit is set to 1. Decrementation is enabled until DCNTDxy underflows or until the first edge of the PCLK (the same as compare match A) after the values in select counter A (TCNT1Dx or TCNT2Dx) and OCR2Dxy or the first edge of the PCLK (the same as compare match B) after the values in select counter B (TCNT1Dx or TCNT2Dx) and OCR2Dxy match. The down counter is decremented every input of the down counter clock while it is enabled.

Once DCNTDxy is enabled, it remains enabled until DCNTDxy underflow, compare match B, or compare match A or B (if selected as a count stop trigger) is detected. While it is enabled, another counter starting trigger or writing 1 to DSTDxy bit has no effect on the enabled state of the counter.

When compare match B or compare match A or B (if selected as a count stop trigger) is detected, the counter is stopped on the first edge of the down counter clock and then cleared to 0000 0000_H. If a down count start trigger (compare match A or writing 1 to DSTDxy) and a down count stop trigger (compare match B) are detected at the same time, the count stop trigger takes priority. The counter is not decremented and no signal on pin TODxyB is output.

Moreover, When no down counter clock is input during the time between enabled state due to compare match A or writing 1 to DSTDxy and counter stop state due to compare match B, the counter stops without performing down-count. The counter stops down-counting in the down-count stop state regardless of that the DCNTDxy value is 0000 0000_H.

A request to stop counting down is retained until the next input of the clock signal to drive counting. Accordingly, after a request to stop counting is issued, a trigger to start counting will not be accepted until the next input of the clock signal to drive counting.

If a value other than 0000 0000_H is set in DCNTDxy after decrementation is terminated by underflow, the counter is not decremented until the counter stopping source is activated. When the DCNTDxy value is 0000 0000_H and the down-

count clock is not input at the beginning of down-counting, only flags in the down count status register 2Dx (DSR2Dx) are updated without performing down-counting. A count down stop interrupt (due to DCNTDxy = 0000 0000_H) is generated at the next down-count clock. The down-count status flag Dxy (DSFDxy) in the down count status register 1Dx (DSR1Dx) retains 1 until an interrupt occurs.

This register is initialized to 0000 0000_H after reset.

30.8.2.32 TIER1Dx — Timer Interrupt Enable Register 1Dx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OV2EDx	OV1EDx
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.141 TIER1Dx Register Contents

Bit Position	Bit Name	Function
7 to 2	—	These bits are not used. When these bits are read, “0” is always returned. When writing, always write “0”.
1	OV2EDx	Overflow 2 interrupt enable Dx Specifies enable/disable of interrupt requests issued by overflow of TCNT2Dx. 0: Interrupt requests are disabled. 1: Interrupt requests are enabled.
0	OV1EDx	Overflow 1 interrupt enable Dx Specifies enable/disable of interrupt requests issued by overflow of TCNT1Dx. 0: Interrupt requests are disabled. 1: Interrupt requests are enabled.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

The timer interrupt enable register 1Dx (TIER1Dx) is an 8-bit readable/writable register. This register enables and disables interrupt requests triggered by overflow of TCNT2Dx and TCNT1Dx.

TIER1Dx is initialized to 00_H after reset.

30.8.2.33 TIER2Dx — Timer Interrupt Enable Register 2Dx

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	UNDED x3	UNDED x2	UNDED x1	UNDED x0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CMPBE Dx3	CMPBE Dx2	CMPBE Dx1	CMPBE Dx0	—	—	—	—	CMPAE Dx3	CMPAE Dx2	CMPAE Dx1	CMPAE Dx0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.142 TIER2Dx Register Contents

Bit Position	Bit Name	Function
31 to 20	—	These bits are not used. Fixe these bits to "0".
19 to 16	UNDEDxy	Underflow interrupt enable Dx Specifies enable/disable of interrupt requests issued by underflow of DCNTDxy. 0: Interrupt requests are disabled. 1: Interrupt requests are enabled.
15 to 12	—	These bits are not used. Fixe these bits to "0".
11 to 8	CMPBEDxy	Compare match B interrupt enable Dxy Specifies enable/disable of interrupt requests issued by compare match B. 0: Interrupt requests are disabled. 1: Interrupt requests are enabled.
7 to 4	—	These bits are not used. Fixe these bits to "0".
3 to 0	CMPAEDxy	Compare match A interrupt enable Dxy Specifies enable/disable of interrupt requests issued by compare match A. 0: Interrupt requests are disabled. 1: Interrupt requests are enabled.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The timer interrupt enable register 2Dx (TIER2Dx) is an 8-bit readable/writable register. UNDEDxy register controls enable/disable of interrupts triggered by underflow of DCNTDxy. CMPBEDxy, CMPAEDxy register enables and disables interrupt requests upon compare match A between the counter (TCNT1Dx or TCNT2Dx) set by the CMPSEL1Dxy bit in the timer compare control register Dx (TCCRLRDx) and OCR1Dxy, and compare match B between the counter (TCNT1Dx or TCNT2Dx) set by the CMPSEL1Dxy bit in the timer compare control register Dx (TCCRLRDx) and OCR2Dxy.

TIER2Dx is initialized to 0000 0000_H after reset.

30.8.2.34 TIER3Dx — Timer Interrupt Enable Register 3Dx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	FEIEDx 3	FEIEDx 2	FEIEDx 1	FEIEDx 0	—	—	—	—	REIEDx 3	REIEDx 2	REIEDx 1	REIEDx 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.143 TIER3Dx Register Contents

Bit Position	Bit Name	Function
15 to 12	—	These bits are not used. Fixe these bits to "0".
11 to 8	FEIEDxy	OFF timing of the output waveform after MIN guard interrupt enable Dxy Specifies enable/disable of interrupt requests issued by off timing of the output waveform after MIN guard interrupt enable Dxy. 0: Interrupt requests are disabled. 1: Interrupt requests are enabled.
7 to 4	—	These bits are not used. Fixe these bits to "0".
3 to 0	REIEDxy	ON timing of the output waveform after MIN guard interrupt enable Dxy Specifies enable/disable of interrupt requests issued by on timing of the output waveform after MIN guard interrupt enable Dxy. 0: Interrupt requests are disabled. 1: Interrupt requests are enabled.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.)

The timer interrupt enable register 3Dx (TIER3Dx) is an 8-bit/16-bit readable/writable register. This register controls enable/disable of interrupts triggered by on/off timing of the output waveform after MIN guard interrupt enable Dxy.

TIER3Dx is initialized to 0000_H after reset.

30.8.2.35 MIGCRDx — MIN Guard Control Register Dx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	MIGENDx3	MIGENDx2	MIGENDx1	MIGENDx0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.144 MIGCRDx Register Contents

Bit Position	Bit Name	Function
7 to 4	—	These bits are not used. Fixe these bits to “0”.
3 to 0	MIGENDxy	MIN Guard function Enable Dxy 0: MIN Guard is disabled. 1: MIN Guard is enabled.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

MIN Guard control register Dx (MIGCRDx) is an 8-bit readable/writable register. This register controls enable/disable of min guard function by one shot pulse output TODxyA/TODxyB.

MIGCRDx is initialized to 00_H after reset.

MIN Guard function Enable Dxy (MIGENDxy)

The MIN guard function enable in OFMINDxy, ONMINDxy, OTOMINDxy can be added to one shot pulsed output TODxyA or TODxyB by writing “1” in this bit.

When changing this bit to 0 from 1, a counter register for MIN guard (OFMICNTDxy, ONMICNTDxy, OTOMICNTDxy) is cleared in 00000000_H and the function of the MIN guard is suspended.(A generated one shot pulse is output directly.)

30.8.2.36 MIGSELDx — MIN Guard Pulse Select Register Dx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	MIGSELx3	MIGSELx2	MIGSELx1	MIGSELx0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

Table 30.145 MIGSELDx Register Contents

Bit Position	Bit Name	Function
7 to 4	—	These bits are not used. When these bits are read, “0” is always returned. When writing, always write “0”.
3 to 0	MIGSELxy	MIN Guard pulse selected. 0: MIN Guard function is added to TODxyA output. 1: MIN Guard function is added to TODxyB output.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.)

MIN Guard pulse select register Dx (MIGSELDx) is an 8-bit readable/writable register. This register select the one shot pulse output (TODxyA/TODxyB) to which the MIN guard function.

MIGSELDx is initialized to 00_H after reset.

MIN Guard pulse selected (MIGSELDxy)

This bits select the one shot pulse output (TODxyA/TODxyB) to which the MIN guard function.

MIN guard is added to TODxyA at 0 setting and min guard is added to TODxyB at 1 setting in OFMINDxy, ONMINDxy, OTOMINDxy.

30.8.2.37 OFMIND_{xy} —Pulse Output Off Minimum Width Setting Register D_{xy}

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OFMIND _{xy} [31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFMIND _{xy} [15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The pulse output off minimum width setting register D_{xy} (OFMIND_{xy}) is a 32-bit readable/writable register. When an enable the MIN guard function (MIGEND_{xy} = 1), the smallest width of the off period can be set as TOD_{xy}A or TOD_{xy}B. If a pulse whose OFF period is shorter than the width set by this register is generated, it is automatically corrected to satisfy the set minimum width.

30.8.2.38 ONMIND_{xy} —Pulse Output On Minimum Width Setting Register D_{xy}

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ONMIND _{xy} [31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ONMIND _{xy} [15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The pulse output on minimum width setting register D_{xy} (ONMIND_{xy}) is a 32-bit readable/writable register. When an enable the MIN guard function (MIGEND_{xy} = 1), the smallest width of the on period can be set as TOD_{xy}A or TOD_{xy}B. If a pulse whose ON period is shorter than the width set by this register is generated, it is automatically corrected to satisfy the set minimum width.

30.8.2.39 OTOMINDxy —Pulse Output On-to-On Minimum Width Setting Register Dxy

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OTOMINDxy[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTOMINDxy[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The pulse output on-to-on minimum width setting register Dxy (ONMINDxy) is a 32-bit readable/writable register that specifies the minimum width of the TODxyB (one-shot pulse output) between a change from OFF to ON and the next change from OFF to ON period in pulse type 2 mode. If a pulse whose ON-to-ON period is shorter than the width set by this register is generated, it is automatically corrected to satisfy the set minimum width.

30.8.2.40 OFMICNTDxy —Pulse Output Off Minimum Width Measurement Counter Dxy

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OFMICNTDxy[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFMICNTDxy[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The pulse output off minimum width measurement counter Dxy (OFMICNTDxy) is a 32-bit readable register that measures the pulse OFF period for correction to secure the minimum width of the pulse OFF period.

30.8.2.41 ONMICNTDxy —Pulse Output On Minimum Width Measurement Counter Dxy

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ONMICNTDxy[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ONMICNTDxy[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The pulse output on minimum width measurement counter Dxy (ONMICNTDxy) is a 32-bit readable register that measures the pulse ON period for correction to secure the minimum width of the pulse ON period.

30.8.2.42 OTOMICNTDxy —Pulse Output On-to-On Minimum Width Measurement Counter Dxy

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OTOMICNTDxy[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTOMICNTDxy[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The pulse output on-to-on minimum width measurement counter Dxy (OTOMICNTDxy) is a 32-bit readable register that measures the pulse ON-to-ON period for correction to secure the minimum width of the pulse ON-to-ON period.

30.8.2.43 ONCAP1Dxy — Input Capture Register ONCAP1Dxy

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ONCAP1Dxy[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ONCAP1Dxy[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The input capture register ONCAP1Dxy (ONCAP1Dxy) is a 32-bit read-only register, and its initial value is 0000 0000_H. This register captures the value of TCNT1Dx0 when TODxyB (pulse output) changes from OFF to ON in min guard function.

ONCAP1Dxy is initialized to 0000 0000_H after reset.

30.8.2.44 ONCAP2Dxy — Input Capture Register ONCAP2Dxy

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ONCAP2Dxy[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ONCAP2Dxy[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The input capture register ONCAP2Dxy (ONCAP2Dxy) is a 32-bit read-only register, and its initial value is 0000 0000_H. This register captures the value of TCNT2Dx0 when TODxyB (pulse output) changes from OFF to ON in min guard function.

ONCAP2Dxy is initialized to 0000 0000_H after reset.

30.8.2.45 OFCAP1Dxy — Input Capture Register OFCAP1Dxy

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OFCAP1Dxy[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFCAP1Dxy[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The input capture register OFCAP1Dxy (OFCAP1Dxy) is a 32-bit read-only register, and its initial value is 0000 0000_H. This register captures the value of TCNT1Dx0 when TODxyB (pulse output) changes from ON to OFF in min guard function.

OFCAP1Dxy is initialized to 0000 0000_H after reset.

30.8.2.46 OFCAP2Dxy — Input Capture Register OFCAP2Dxy

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OFCAP2Dxy[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFCAP2Dxy[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The input capture register OFCAP2Dxy (OFCAP2Dxy) is a 32-bit read-only register, and its initial value is 0000 0000_H. This register captures the value of TCNT2Dx0 when TODxyB (pulse output) changes from ON to OFF in min guard function.

OFCAP2Dxy is initialized to 0000 0000_H after reset.

30.8.2.47 TSR2Dx — Timer Status Register 2Dx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	FEFDx3	FEFDx2	FEFDx1	FEFDx0	—	—	—	—	REFDx3	REFDx2	REFDx1	REFDx0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.146 TSR2Dx Register Contents

Bit Position	Bit Name	Function
15 to 12	—	These bits are not used. When these bits are read, “0” is always returned. When writing, always write “0”.
11 to 8	FEFDx3 to FEFDx0	Output pulse after MIN guard OFF flag Dx3 to Dx0 0: Output pulse after MIN guard has not changed from ON to OFF 1: Output pulse after MIN guard has changed from ON to OFF
7 to 4	—	These bits are not used. When these bits are read, “0” is always returned. When writing, always write “0”.
3 to 0	REFDx3 to REFDx0	Output pulse after MIN guard ON flag Dx3 to Dx0 0: Output pulse after MIN guard has not changed from OFF to ON 1: Output pulse after MIN guard has changed from OFF to ON

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1

The timer status register 2 (TSR2Dx) is an 8-bit/16-bit read-only register. This register indicates the change of the output pulse in subblock Dx (x = 0 to SB_D-1) after MIN guard, from ON to OFF or OFF to ON.

If an interrupt source takes effect when this flag is set, an interrupt request is made again. An interrupt request is made even if there is a conflict between clearing by the corresponding timer status clear register and setting by an interrupt source.

The flags can be cleared by setting the corresponding bits of the timer status clear register 2 (TSCR2Dx).

TSR2Dx is initialized to 0000_H after reset.

Output pulse after MIN guard OFF flag Dxy (FEFDxy)

FEFDxy	Function
0	[Clearing condition] When 1 is written to the FEFDxy bit of the timer status clear register 2 (TSCR2Dx). (Default)
1	[Setting condition] When the output pulse after MIN guard has changed from ON to OFF

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1 y = 0, 1, 2, 3: Corresponding to channels 0 to 3

Output pulse after MIN guard ON flag Dxy (REFDxy)

REFDxy	Function
0	[Clearing condition] When 1 is written to the REFDxy bit of the timer status clear register 2 (TSCR2Dx). (Default)
1	[Setting condition] When the output pulse after MIN guard has changed from OFF to ON

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1 y = 0, 1, 2, 3: Corresponding to channels 0 to 3

30.8.2.48 TSCR2Dx — Timer Status Clear Register 2Dx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	FEFCD x3	FEFCD x2	FEFCD x1	FEFCD x0	—	—	—	—	REFCD x3	REFCD x2	REFCD x1	REFCD x0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W	R	R	R	R	W	W	W	W

Table 30.147 TSCR2Dx Register Contents

Bit Position	Bit Name	Function
15 to 12	—	These bits are not used. When these bits are read, “0” is always returned. When writing, always write “0”.
11 to 8	FEFCDx3 to FEFCDx0	Output pulse after MIN guard OFF flag clear enable Dx3 to Dx0 0: Disabled (Default) 1: Clears FEFCDx3-FEFCDx0 of timer status register 2D (TSR2Dx) to 0.
7 to 4	—	These bits are not used. When these bits are read, “0” is always returned. When writing, always write “0”.
3 to 0	REFCDx3 to REFCDx0	Output pulse after MIN guard ON flag clear enable Dx3 to Dx0 0: Disabled (Default) 1: Clears REFCDx3-REFCDx0 of timer status register 2D (TSR2Dx) to 0.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1.

The timer status clear register 2Dx (TSCR2Dx) is an 8-bit/16-bit readable/writable register. This register clears the flags that indicate the change of output pulse after MIN guard (from ON to OFF or vice versa).

TSCR2Dx is initialized to 0000_H after reset.

Output pulse after MIN guard OFF flag clear enable 2Dxy (FEFCDxy)

This bit clears the flags that indicate the change of output pulse after MIN guard from ON to OFF.

Setting this bit enables clearing of the output pulse after MIN guard OFF flag (FEFCDxy) of the timer status register 2Dx (TSR2Dx). When these bits are read, “0” is always returned.

FEFCDxy	Function
0	Disabled (Default)
1	Clears FEFCDxy to 0.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1 y = 0, 1, 2, 3: Corresponding to channels 0 to 3

Output pulse after MIN guard ON flag clear enable 2Dxy (REFCDxy)

This bit clears the flags that indicate the change of output pulse after MIN guard from OFF to ON.

Setting this bit enables clearing of the output pulse after MIN guard ON flag (REFCDxy) of the timer status register 2Dx (TSR2Dx). When these bits are read, “0” is always returned.

REFCDxy	Function
0	Disabled (Default)
1	Clears REFCDxy to 0.

Note: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1 y = 0, 1, 2, 3: Corresponding to channels 0 to 3

30.8.2.49 CCAPSELD — Common Input Capture Select Register D

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CCAPSEL
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 30.148 CCAPSELD Register Contents

Bit Position	Bit Name	Function
7 to 1	—	These bits are not used. When these bits are read, “0” is always returned. When writing, always write “0”.
0	CCAPSEL	Common input capture select. 0: The value of the TCNT1D0 is stocked in CICRD0-7 1: The value of the TCNT2D0 is stocked in CICRD0-7

Common input capture select register D (CCAPSELD) is an 8-bit readable/writable register.

When a PH renewal notice trigger occurred, Its select which to stock in Common input capture registerD0-7, Timer Counter 1D0 (TCNT1D0) or Timer Counter 2D0 (TCNT2D0).

CCAPSELD is initialized to 00_H after reset.

30.8.2.50 CICRD0-7 — Common Input Capture Register D0-7

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CICRD0-7[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CICRD0-7[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The common input capture register D0-7 (CICRD0-7) is a 32-bit read-only register, and its initial value is 0000 0000_H.

The value of the timer counter (If CCAPSEL = 0 then TCNT1D0, CCAPSEL = 1 then TCNT2D0) is stocked by PH renewal notice trigger 0~7 from DFE.

CICRD_x is assigned to “PH renewal notice trigger x”. (x = 0-7)

CICRD0-7 is initialized to 0000 0000_H after reset.

30.8.3 Details of Operation

Timer D consists of SB_D subblocks. Each block has two range compare setting registers (RCR1D_x and RCR2D_x), two 32-bit up-counters (TCNT1D_x and TCNT2D_x), one offset base register (OSBRD_x), one output value register (ODRD_x), one output select register (OSELRD_x), and four channels described below. Each channel consists of the output compare register (OCR1D_{xy}) to which select counter A (TCNT1D_x or TCNT2D_x) is compared, the output compare register (OCR2D_{xy}) to which select counter B (TCNT1D_x or TCNT2D_x) is compared, one down-counter D_{xy} (DCNTD_{xy}), two upper-limit setting registers (CUCR1D_x and CUCR2D_x) for two counters, and two input capture registers (ICR1D_{xy} and ICR2D_{xy}).

One-shot pulse can be output from timer D. By using compare match A, compare match B, compare match A or B, (or compare match A and B), 1 writing in to DSTRD_x.DSTD_{xy} as a start trigger of a down-counter, one-shot pulse with an offset can also be output.

Setting the TDE bit in ATUENR and bit STRD_x in timer start register (TSTRD) to 1 makes two up-counters (TCNT1D_x and TCNT2D_x) in subblock D_x start operation. Also, this becomes a request for counting down by the down counter to start.

The down-counter is started by any of five sources; setting the DSTD_{xy} bit in the down count start register (DSTRD) to 1, compare match A, compare match B, compare match A or B, and compare match A and B that are set by the TRGSEL_{Dxy} bit in the down counter control register (DCRD). When any of these triggers occurs, DCNTD_{xy} driven by the down counter clock is started. But, during off-Min guard and during OTO-Min guard or DCNTD_{xy} doesn't begin down count movement at 0000_0000_H.

The down counter is stopped on an underflow of the down counter and compare match B or A or B selected by bit TRGSEL_{xy}. When an underflow occurs (during down-counting when the counter value is 0000 0000_H), the value of the down counter (DCNTD_{xy}) is immediately cleared to 0000 0000_H and the down-counting is terminated. When a compare match B or compare match A or B occurs, the same operation is performed in synchronization with the next down-count clock.

For compare match A between TCNT1D_x (or TCNT2D_x) and OCR1D_{xy} and compare match B between TCNT2D_x (or TCNT1D_x) and OCR2D_{xy} are set to 1 on the first edge of the PCLK after the compare match. A compare match interrupt request is issued to CPU triggered by the compare match. According to the settings of range comparison value setting registers (RCR1D_x and RCR2D_x), enable the range compare function, compare match A can be generated if it is within the range set by select counter A when writing to OCR1D_{xy}, compare match B can be generated if it is within the range set by select counter B when writing to OCR2D_{xy}. Furthermore, the counter value of select counter B is captured into ICR2D_{xy} triggered by compare match A and the counter value of select counter A is captured into ICR1D_{xy} triggered by compare match B. The CMFAD_{xy} and CMFBD_{xy} bits in TSRD_x are set to 1 in synchronization with PCLK in the occurrence of compare match.

A signal on pin TOD_{xy}A is output when the output source selected in the OSSD_{xy} bit in timer I/O control register 1D (TIOR1D_x) is activated. For example, assume that compare match A is selected. A signal level set by the IOA_{xy} bit is output on pin TOD_{xy}A on the first edge of the PCLK after the compare match between select counter A and OCR1D_{xy}.

Output of the one-shot pulse is synchronized with the down counter clock in a way similar to the down counter operation. The five counter starting sources are synchronized with the PCLK, a signal on pin TOD_{xy}B is output on the first edge of the down counter clock after the source is activated. Underflow of the down counter, which is a counter stopping source, is synchronized with the down counter clock and compare match B and compare match A or B is synchronized with the PCLK. As to negation timing, the TOD_{xy}B signal is negated in synchronization with underflow and on the first edge of the down counter clock after the compare match B or A or B.

Output Pin	Assertion Timing	Negation Timing	Initial Value
TODxyB	On the first edge of the down counter clock after the counter starting source is activated	On the first edge of the down counter clock after compare match B or on DCNTDxy underflow	0 (inverted depending on TOCRDx)

Note 1: x = 0 to SB_D-1: Corresponding to subblocks D0 to DSB_D-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Note 2: If an assertion and a negation occur simultaneously, the negation takes priority.

The initial values output on pins TODxyA and TODxyB are 0. However, the output level can be inverted by setting timer output control register (TOCRDx). Setting bit TONEADx to 1 makes pin TODxyA in subblock Dx inverted and setting bit TONEBDx to 1 makes pin TODxyB in subblock Dx inverted. By setting the output select register Dx (OSELRDx), the value of the corresponding output value register Dx (ODRDx) can be output to pins (TODxyA and TODxyB).

To set the clocks for TCNT1Dx, TCNT2Dx, or DCNTDxy or values in registers such as DCNTDxy, OCR1Dxy, and OCR2Dxy while TCNT1Dx or TCNT2Dx is in operation, note that the value to be set may lead to malfunction. For example, while setting the compare match value, the counter value may exceed the value to be set.

Figure 30.57 shows an operation example of one-shot pulse output for channel 0 in subblock D0.

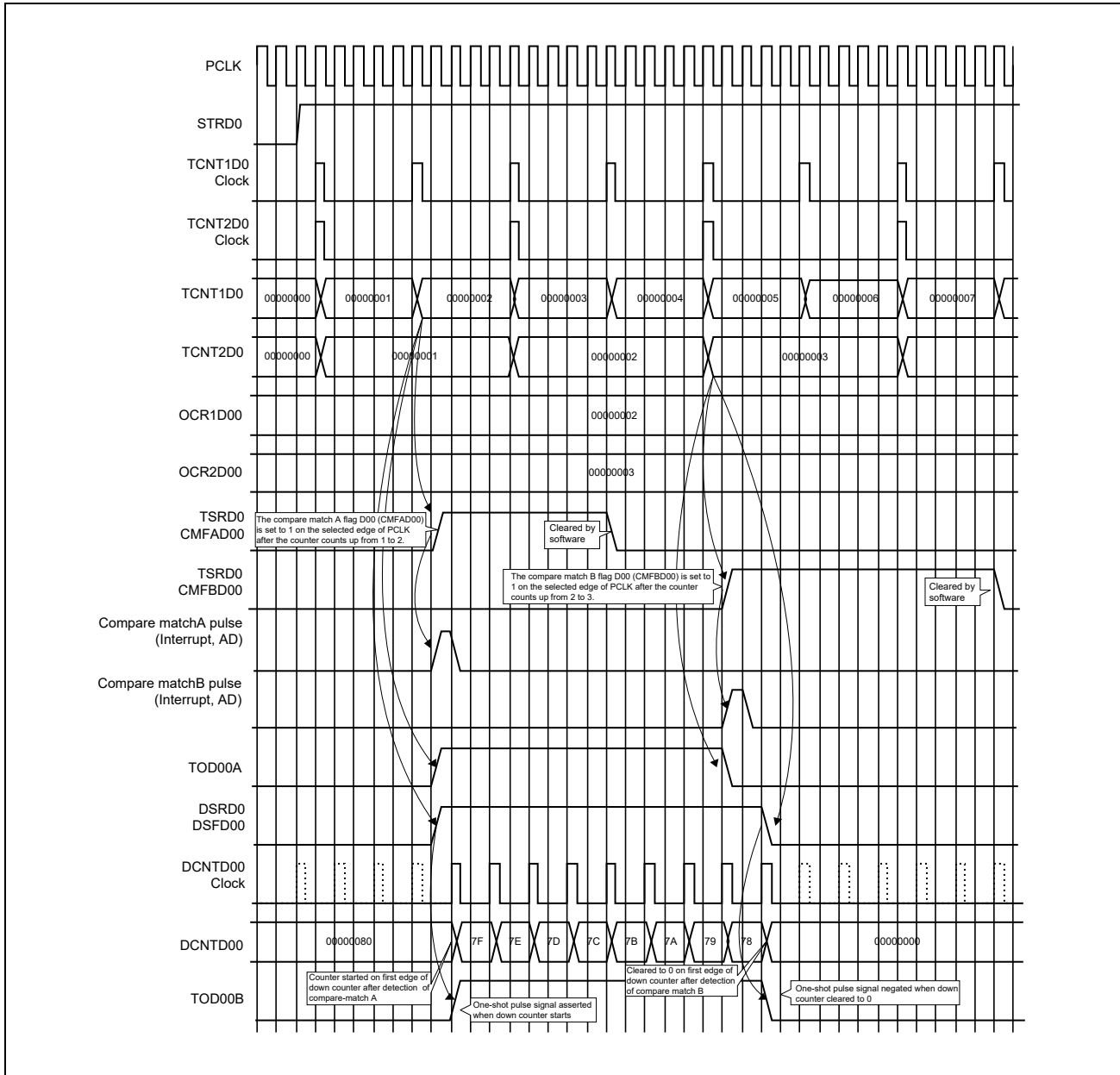


Figure 30.57 Operation Example of One-Shot Pulse Output (1) - Counting Started on Compare Match A and Stopped on Compare Match B -

Figure 30.57 shows operations of the down counter when the counter is started on compare match A and is stopped on compare match B. It also shows the assertion and negation of a one-shot pulse. Compare match A as the counter starting trigger and compare match B as the counter stopping trigger are set by the TRGSEL0 bits in DCRD0. Both matches are selected as the source of the output signal by the OSSD0 bits in TIOR1D0 and output levels are set by the IOAD0 and IOBD0 bits. A logical one for compare match A and a logical zero for compare match A are selected.

When the timer start register (TSTRD) is set to 1, TCNT1D0 and TCNT2D0 start counting up immediately after the next count clocks for TCNT1D0 and TCNT2D0, respectively. If TCNT1D0 coincides with the output compare register (OCR1D0), compare match A occurs at the next PCLK cycle. At this time, the counter value of TCNT2Dx is captured in ICR2Dxy. In addition, a compare match interrupt request is issued to CPU triggered by compare match A. The compare match A status flag (CMFAD0) is set to 1. At the same time, TOD00A outputs 1 and the down count status flag (DSFD0) is also set to 1, entering the down counter enabled state. This down count enabled state continues until

detection of compare match B or under flow of DCNTD00. Input of the down counter clock during this state decrements DCNTD00. TOD00B outputs 1 from the time of the first down counter clock input.

If TCNT2D0 coincides with the output compare register (OCR2D00), compare match B occurs at the next PCLK cycle. At this time, the counter value of TCNT1Dx is captured into ICR1Dxy. In addition, a compare match interrupt request is issued to CPU triggered by compare match B. The compare match B flag (CMFBD00) is set to 1. At this time, a level of 0 is output on pin TOD00A. The down counter is cleared and a one-shot pulse (TOD00B) is terminated.

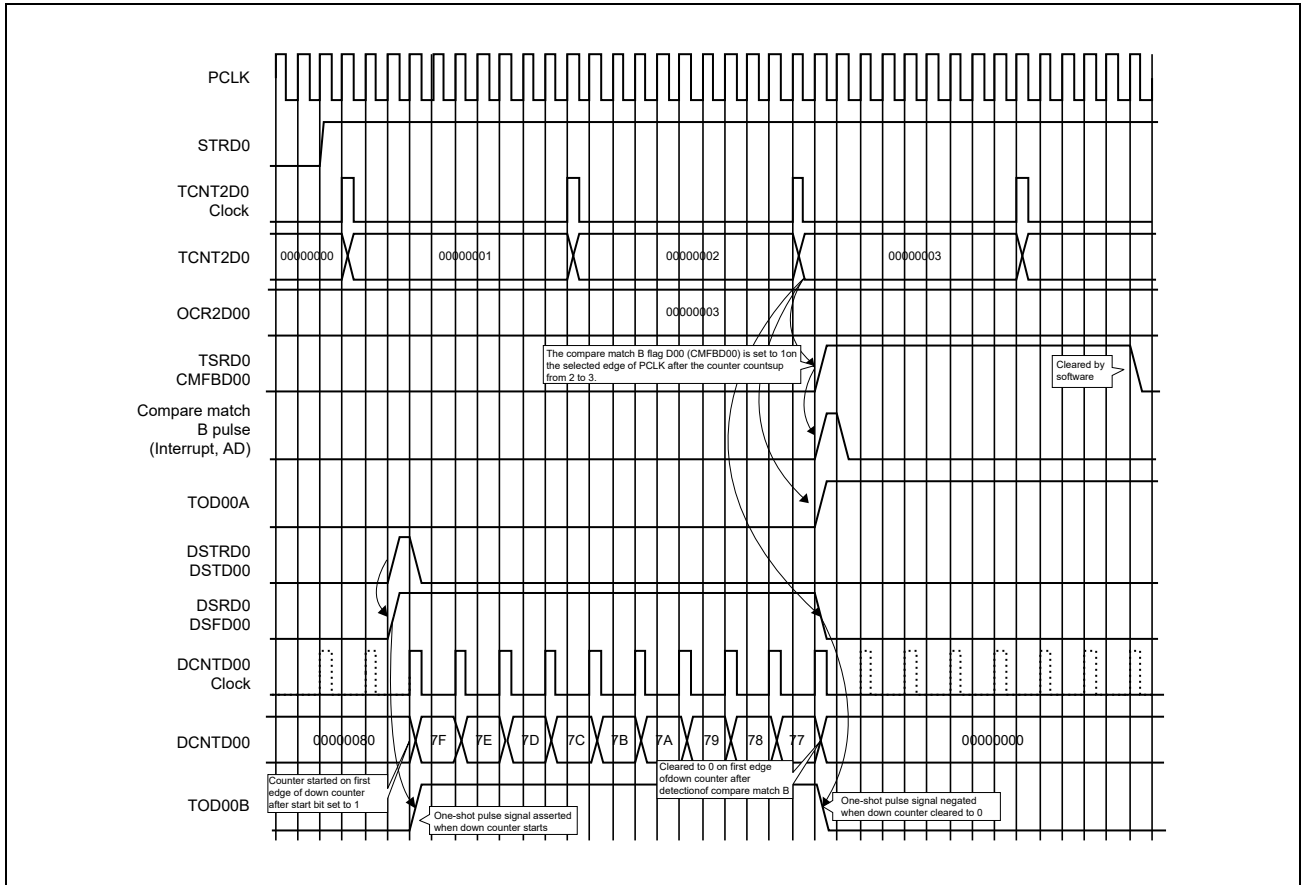


Figure 30.58 Operation Example of One-Shot Pulse Output (2) - Counting Started by Writing 1 to Counter Starting Bit and Stopped on Compare Match B -

Figure 30.58 shows an operation example when the down counter is started by writing 1 to the down counter starting bit. In this example, the counter starting trigger is not selected and compare match B is selected as the counter stopping trigger (TRGSELD00 in DCRD0). The source of the output signal is compare match B (OSSD00 in TIOR1D0) and logical one is output (IOBD00 in TIOR1D0).

Figure 30.59 shows an operation example of a one-shot pulse output for channel 0 in subblock D0.

Setting the DSTD00 bit in the down counter starting register (DSTRD0) sets the down counter status flag (DSFD00) to 1. This makes the down counter ready for counting down. DCNTD00 is started on the first edge of the down counter clock after DSFD00 is set to 1. At this time, a level of 1 is output on pin TOD00B.

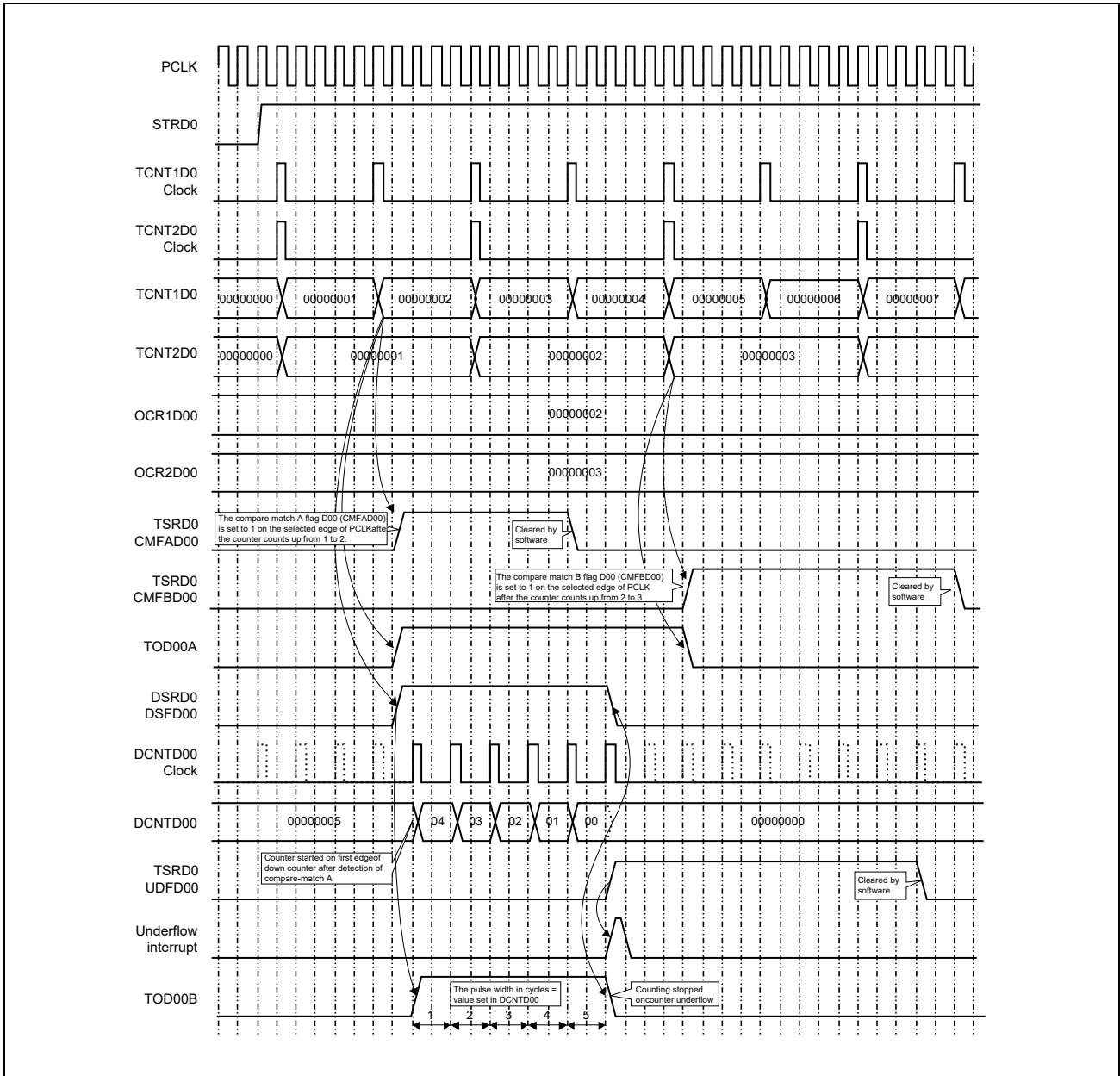


Figure 30.59 Operation Example of One-Shot Pulse Output (3) – Underflow Occurs -

Figure 30.59 shows an operation example when a one-shot pulse is terminated on underflow. In this example, compare match A as the counter starting trigger is selected and the counter stopping trigger is not selected (TRGSEL0 in DCRD0). Both compare matches A and B as the source of the output signal (OSSD0 in TIOR1D0) are selected. Output levels of 1 for compare match A (IOAD00) and 0 for compare match B (IOBD00) are selected. Underflow is detected and the underflow flag (UDFD00) in TSRD0 is set on the first edge of the down counter clock after the value in timer down counter Dxy (DCNTD00) is 00 0000_H. At the same time, the one-shot pulse output is terminated. The width of the pulse output on pin TOD00B is equal to the value set in DCNTD00 before counting down.

30.8.3.1 Range Comparison Function

(1) Range compare function

The range compare function is a function to generate a compare match when the counter value (TCNT1Dx, TCNT2Dx) is within the compare range set when outputting to the output compare register (OCR1Dxy, OCR2Dxy). The range compare function is valid when other than 00_H is set in the range compare value setting register (RCR1Dx or RCR2Dx). Range compare match A and B correspond to compare match A and B, respectively. Range Compare range of compare match A, B is judged by using the following values.

Range compare match A

- Write value of OCR1Dxy
- Value of select counter A
- Range value for select counter A
- Upper limit value of select counter A

Range compare match B

- Write value of OCR2Dxy
- Value of select counter B
- Range value for select counter B
- Upper limit value of select counter B

Figure 30.60 and **Figure 30.61** show the operation diagram of range compare match.

Also, OCR1Dxy and OCR2Dxy capture counter values (TCNT1Dx, TCNT2Dx) when a compare match is generated by the range compare function.

Figure 30.62 shows the capture operation when range compare match occurs.

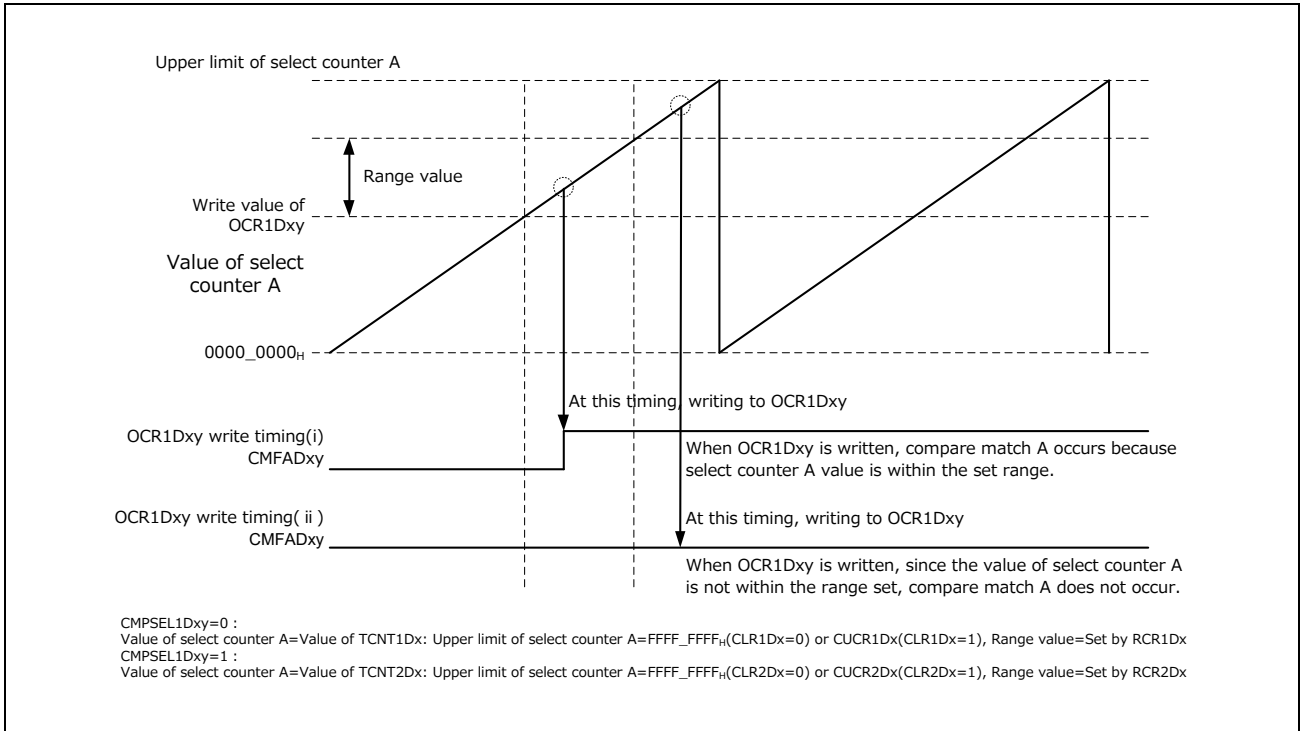


Figure 30.60 Operation of the Range Comparison Function(1)

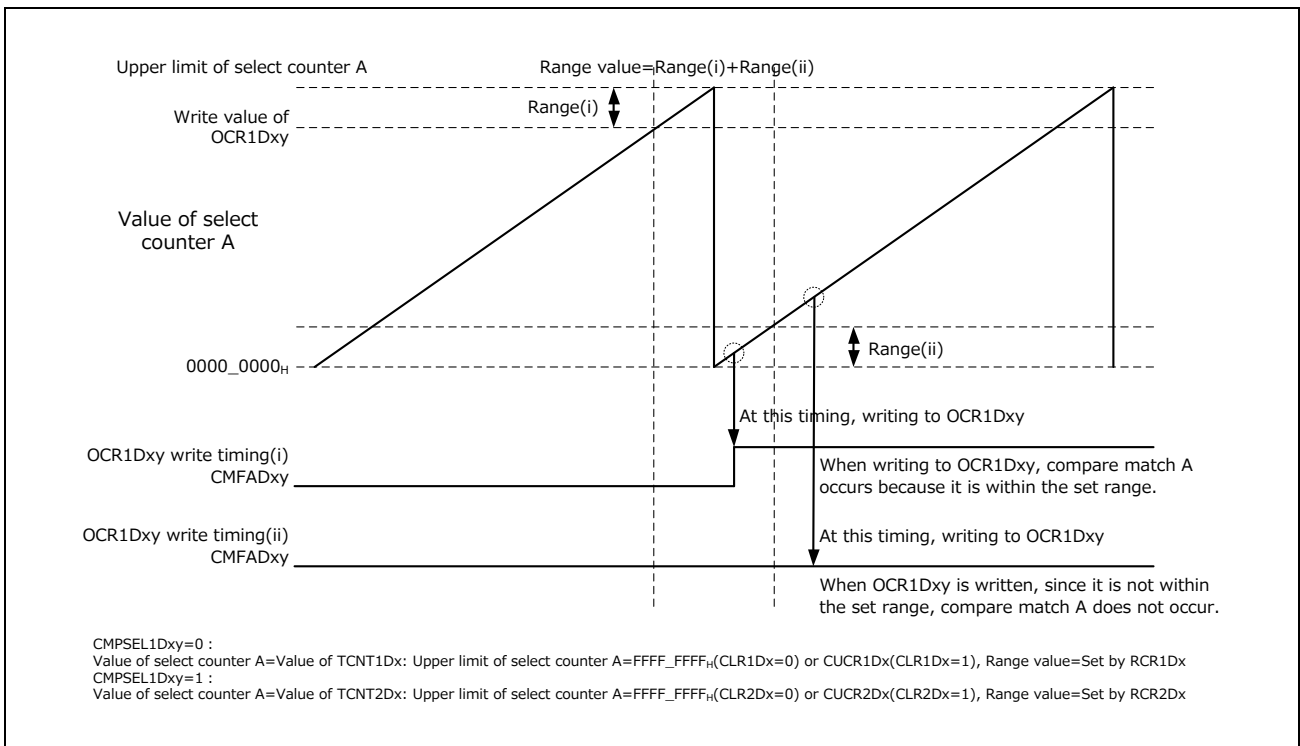


Figure 30.61 Operation of the Range Comparison Function(2)

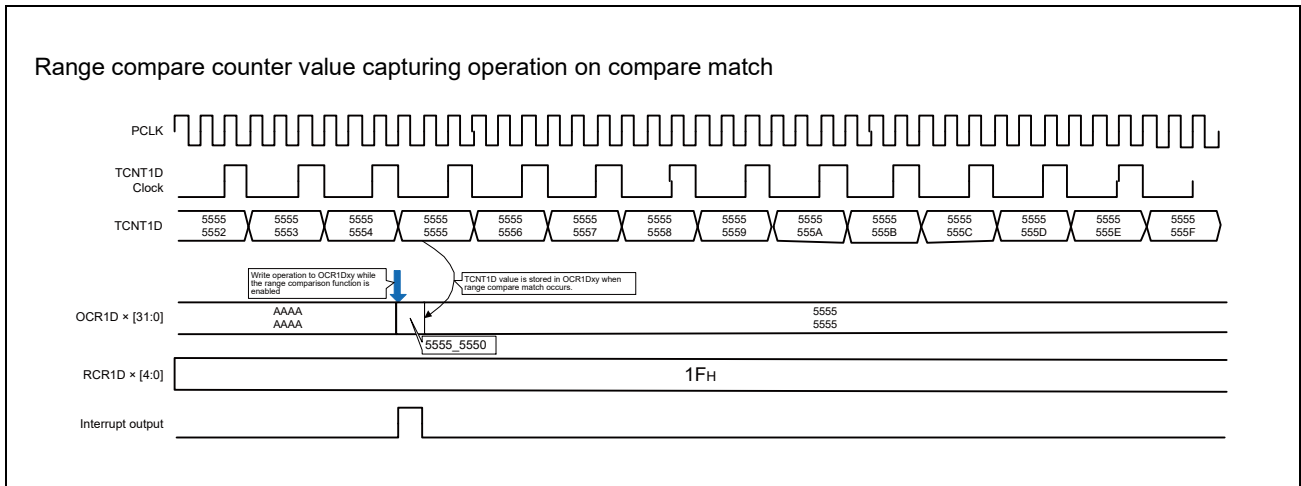


Figure 30.62 Range Compare Counter Value Capturing Operation

Range compare match A occurs if the condition of **Table 30.149** is satisfied when the range compare function is enabled (other than 00_H for the range value for select counter A) and writing to OCR1Dxy is performed.

Table 30.149 Range Compare Match A Generation Condition

	Range condition	Range compare match A generation condition
Condition 1	When the set compare range is lower than select counter A upper limit OCR1Dxy + Range value for select counter A ≤ Upper limit value of select counter A	OCR1Dxy ≤ Value of select counter A ≤ OCR1Dxy + Range value for select counter A
Condition 2	When the set compare range exceeds select counter A upper limit Upper limit value of select counter A < OCR1Dxy + Range value for select counter A	Range(i) : OCR1Dxy ≤ Value of select counter A ≤ Upper limit value of select counter A or Range(ii) : 00000000 ≤ Value of select counter A ≤ OCR1Dxy + Range value for select counter A – Upper limit value of select counter A

Note 1. “Condition 1 When the set compare range is lower than select counter A upper limit”, The range condition shown in the **Figure 30.60**.

Note 2. “Condition 2 When the set compare range exceeds select counter A”, The range condition shown in the **Figure 30.61**.

Note 3. Range (i) and Range (ii) in the table indicate Range (i) and Range (ii) in **Figure 30.61**.

Table 30.150 lists the registers used to determine the compare range of “Range compare match A”.

Table 30.150 Register Used in Range Judgment of Compare Match A

Configuration			Used registers			
CMPSEL1Dxy	CLR1Dx	CLR2Dx	Value of select counter A	Set value for select counter A	Upper limit value of select counter A	Output compare register
0	0	—	TCNT1Dx	RCR1Dx	FFFFFFF _H	OCR1Dxy
0	1	—			CUCR1Dx	
1	—	0	TCNT2Dx	RCR2Dx	FFFFFFF _H	
1	—	1			CUCR2Dx	

Note 1. “—” indicates that the setting has no effect.

Range compare match B occurs if the condition of **Table 30.151** is satisfied when the range compare function is enabled (other than 00_H for the range value for select counter B) and writing to OCR2Dxy is performed.

Table 30.151 Range Compare Match B Generation Condition

	Range condition	Range compare match B generation condition
Condition 1	When the set compare range is lower than select counter B upper limit OCR2Dxy + Range value for select counter B ≤ Upper limit value of select counter B	OCR2Dxy ≤ Value of select counter B ≤ OCR2Dxy + Range value for select counter B
Condition 2	When the set compare range exceeds select counter B upper limit Upper limit value of select counter B < OCR2Dxy + Range value for select counter B	Range(i) : OCR2Dxy ≤ Value of select counter B ≤ Upper limit value of select counter B or Range(ii) : 00000000 ≤ Value of select counter B ≤ OCR2Dxy + Range value for select counter B – Upper limit value of select counter B

Note 1. “Condition 1 When the set compare range is lower than select counter B upper limit”, The range condition shown in the **Figure 30.60**.

Note 2. “Condition 2 When the set compare range exceeds select counter B”, The range condition shown in the **Figure 30.61**.

Note 3. Range (i) and Range (ii) in the table indicate Range (i) and Range (ii) in **Figure 30.61**.

Table 30.152 lists the registers used to determine the compare range of “Range compare match B”.

Table 30.152 Register Used in Range Judgment of Compare Match B

Configuration			Used registers			
CMPSEL2Dxy	CLR2Dx	CLR1Dx	Value of select counter B	Set value for select counter B	Upper limit value of select counter B	Output compare register
0	0	—	TCNT2Dx	RCR2Dx	FFFFFFF _H	OCR2Dxy
0	1	—			CUCR2Dx	
1	—	0	TCNT1Dx	RCR1Dx	FFFFFFF _H	
1	—	1			CUCR1Dx	

Note 1. “—” indicates that the setting has no effect.

The table below shows the setting example when the range compare function is used and the counter value at which the range compare match occurs.

No.1, No.3 are examples of the condition “When the set compare range is less than or equal to the select counter A/B upper limit value”, and No.2, No.4 are examples of the condition “When the set compare range exceeds the select counter A/B upper limit value”.

Table 30.153 Range Compare Function Register Setting Example

No.	CLR1Dx or CLR2Dx bit value	Counter upper limit setting compare register (CUCR1Dx, CUCR2Dx)	The write value of the output compare register (OCR1Dxy, OCR2Dxy)	Set value (range value) of range compare value setting register (RCR1Dx, RCR2Dx)	When writing to the output compare register (OCR1Dxy, OCR2Dxy), the value of the counter where range compare match occurs
0	0	—	FFFF0000 _H	12 (range value: FFF _H)	FFFF0000 _H to FFFF0FFF _H *1
1	0	—	FFFF0000 _H	18 (range value: 3FFFF _H)	FFFF0000 _H to FFFFFFFF _H 00000000 _H to 00030000 _H *2
2	1	00000FFF _H	00000E00 _H	6 (range value: 3F _H)	00000E00 _H to 00000E3F _H *3
3	1	00000FFF _H	00000E00 _H	10 (range value: 3FF _H)	00000E00 _H to 00000FFF _H 00000000 _H to 00000200 _H *4

Note 1. FFFF0000_H to (FFFF0000_H + 2¹² -1)

Note 2. 00000000_H to (FFFF0000_H + 2¹⁸ -1 – FFFFFFFF_H)

Note 3. 00000E00_H to (00000E00_H + 2⁶ -1)

Note 4. 00000000_H to (00000E00_H + 2¹⁰ -1 – 00000FFF_H)

30.8.3.2 Counter Upper-Limit Setting Function

According to the setting for the counter upper-limit setting compare register 1 (CUCR1Dx), the upper-limit value of the timer counter (TCNT1Dx) in each subblock can be modified to change the one-shot pulse cycle. To enable the counter upper-limit setting function, the CLR1Dx bit in the timer control register D (TCRDx) must be set to 1.

According to the setting for the counter upper-limit setting compare register 2 (CUCR2Dx), the upper-limit value of the timer counter (TCNT2Dx) in each subblock can be modified to change the one-shot pulse cycle. To enable the counter upper-limit setting function, the CLR2Dx bit in the timer control register D (TCRDx) must be set to 1.

Figure 30.63 shows an example of operation when the counter upper-limit setting function in block D0 is enabled.

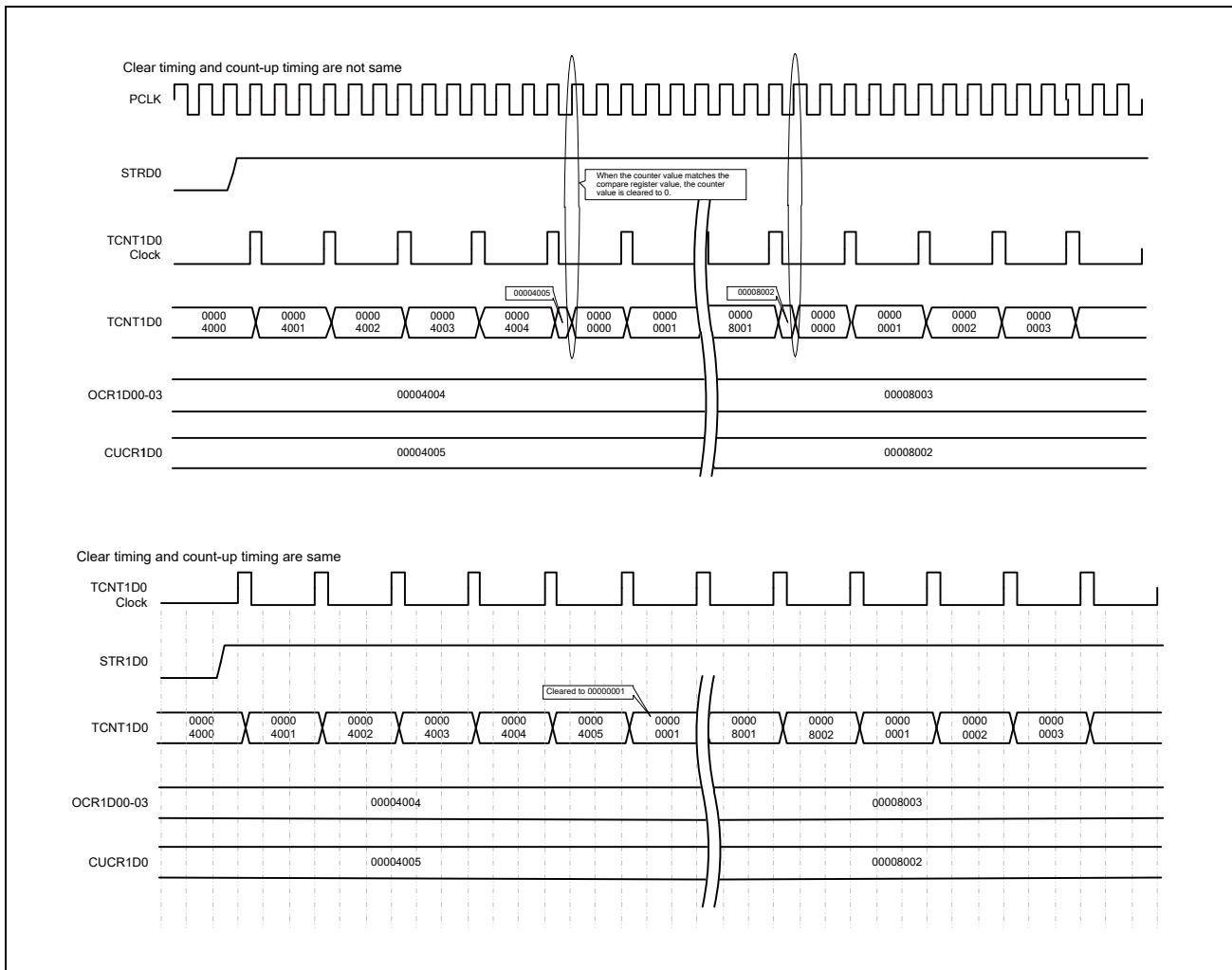


Figure 30.63 Operation of the Counter Upper-Limit Setting Function

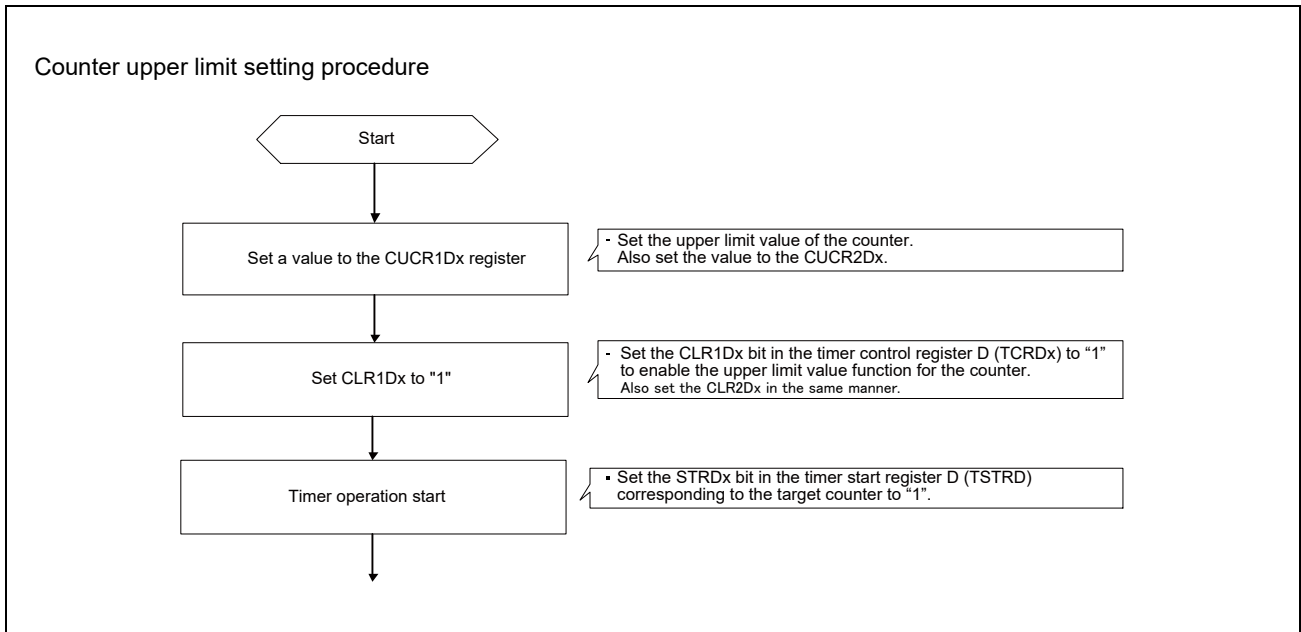


Figure 30.64 Counter Upper-Limit Function Setting Procedure

30.8.3.3 Capture Function

The input capture registers 1Dxy (ICR1Dxy) and 2Dxy (ICR2Dxy) capture the values counted by timer counters 1Dx (TCNT1Dx) and 2Dx (TCNT2Dx) in response to the compare match B and compare match A triggers, respectively. Capture proceeds on the rising edges of the pulses produced by matches.

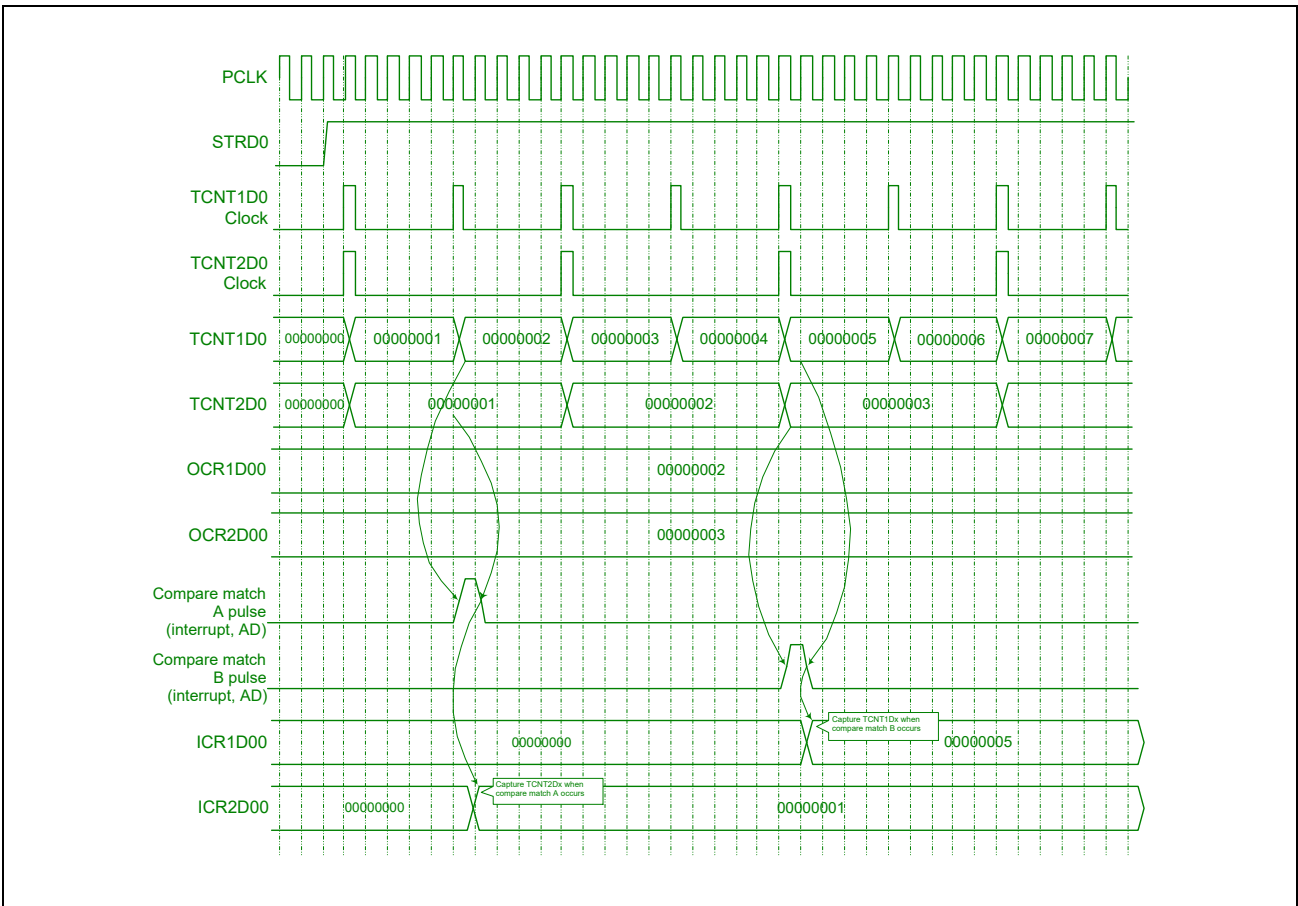


Figure 30.65 Capture Operation

30.8.3.4 MIN Guard Function

The MIN guard unit is connected to each channel of timer D, and it is possible to add MIN guard function to TODxyA (compare match output) or TODxyB (one shot pulse output).

Figure 30.66 shows the basic operation of the MIN guard function.

When the waveform selected from TODxyA, TODxyB changes from OFF to ON, ONMINDxy setting value is loaded to ONMICNTDxy and ON width of the selected waveform is guarded with ONMICNTDxy.

Similarly, when the waveform selected from TODxyA and TODxyB changes from ON to OFF, the setting value of OFMINDxy is loaded into OFMICNTDxy, and the setting value of OTOMINDxy is loaded into OTOMICNTDxy at the timing when it switches from OFF to ON, and the selected waveform Guard the OFF width and ON to ON width.

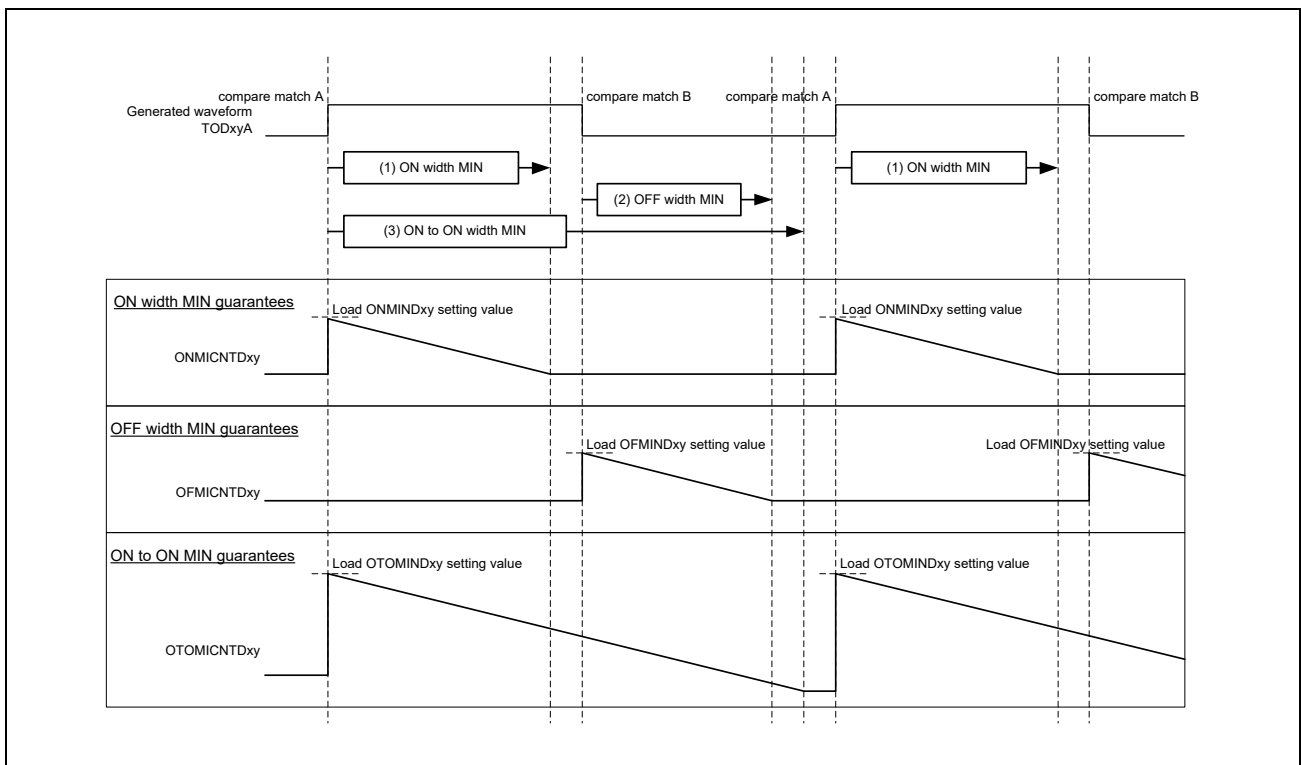


Figure 30.66 Basic Operation of the MIN Guard Function

Figure 30.67 shows the output waveform when MIN guard correction of ON width is performed for TODxyA, TODxyB.

At the timing from OFF to ON, ONMINDxy setting value is loaded into ONMICNTDxy and start down counting.

If the generated waveform (TODxyA or TODxyB) turns OFF before ONMICNTDxy becomes 0, the MIN guard correction is performed and the waveform after MIN guard (TODxyMA or TODxyMB) continues to be ON. After that, OFF waveform is output when ONMICNDxy becomes 0.

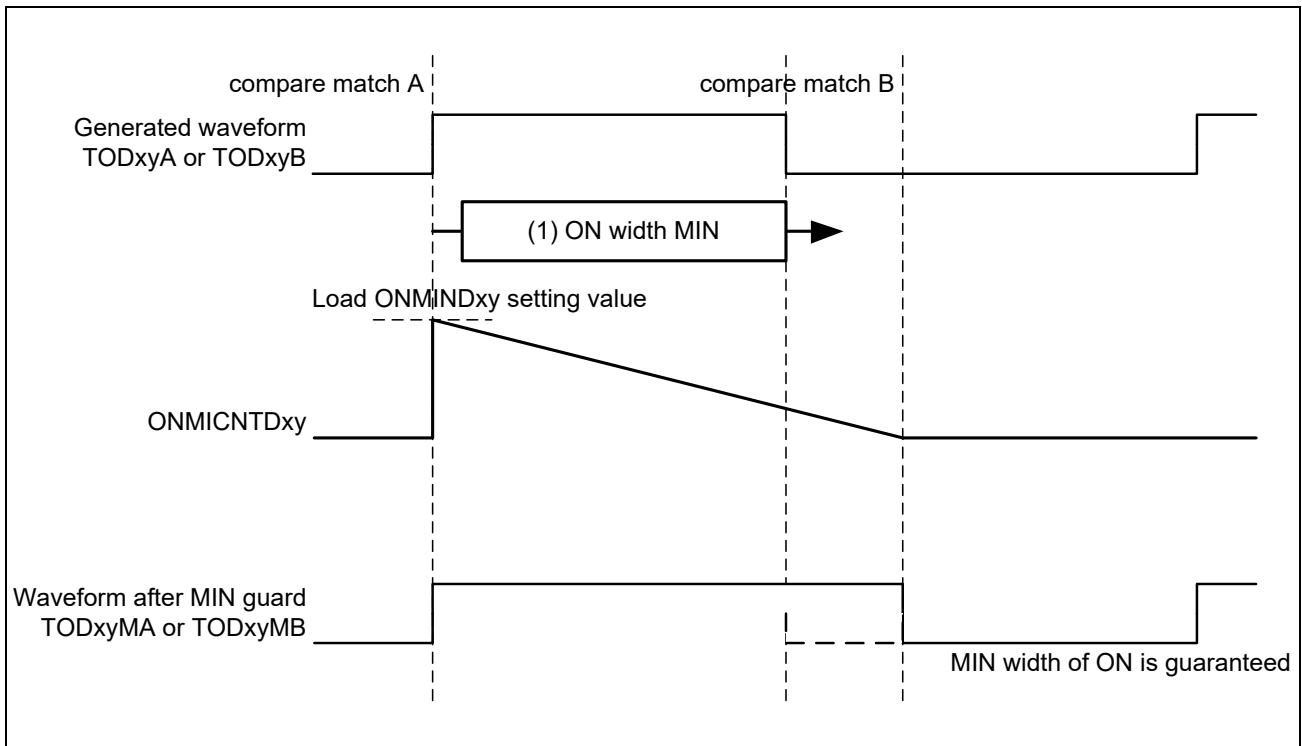


Figure 30.67 MIN Guard Function ON Width MIN Correction Operation Diagram

Figure 30.68 shows the output waveform when MIN guard correction of OFF width is performed for TODxyA. At the timing from ON to OFF, load OFMINDxy setting value into OFMICNTDxy and start down counting.

If the generated waveform (TODxyA) turns ON before OFMICNTDxy becomes 0, the MIN guard correction is performed, and the waveform after the MIN guard (TODxyMA) continues to be OFF.

After that, the ON waveform is output at the timing when OFMICNDxy becomes 0.

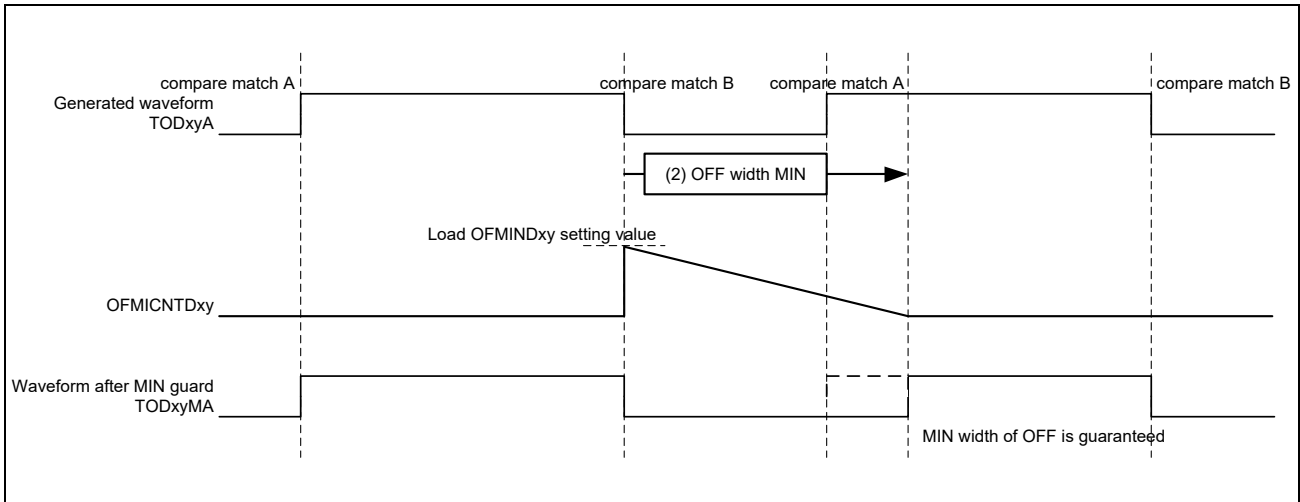


Figure 30.68 MIN Guard Function OFF Width MIN Correction Operation Diagram (TODxyA Correction)

Figure 30.69 shows the output waveform when MIN guard correction of OFF width is performed for TODxyB.

In the correction operation for TODxyB, the downcounter DCNTDxy starts down-counting at the OFF to ON timing of the OFF width MIN correction waveform and guarantees the Max value of the ON width of the output waveform.

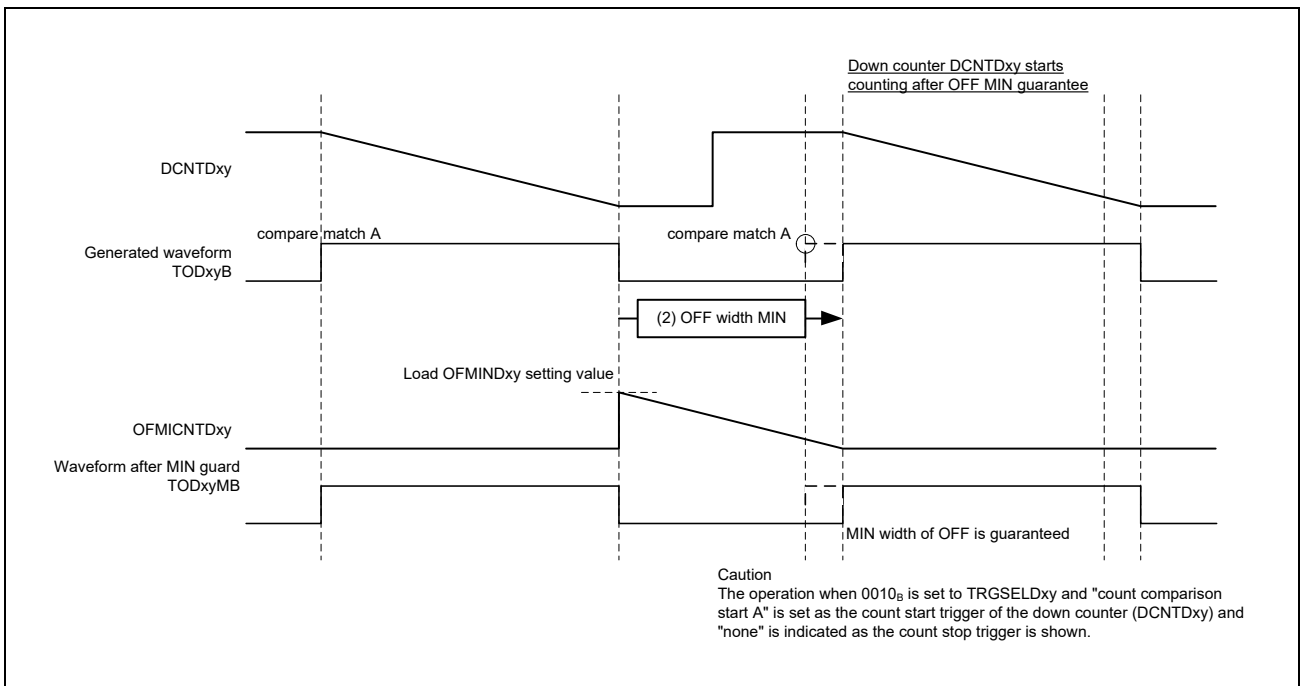


Figure 30.69 MIN Guard Function OFF Width MIN Correction Operation Diagram (TODxyB Correction)

Figure 30.70 shows the output waveform when MIN guard correction with ON to ON width is performed on TODxyA.

At the timing from OFF to ON, load the OTOMINDxy setting value into OTOMICNTDxy and start down counting.

If the generated waveform (TODxyA) turns ON before OTOMICNTDxy becomes 0, the MIN guard correction is performed, and the waveform after MIN guard (TODxyMA) continues to be OFF. After that, the ON waveform is output at the timing when OTOMICNDxy becomes 0.

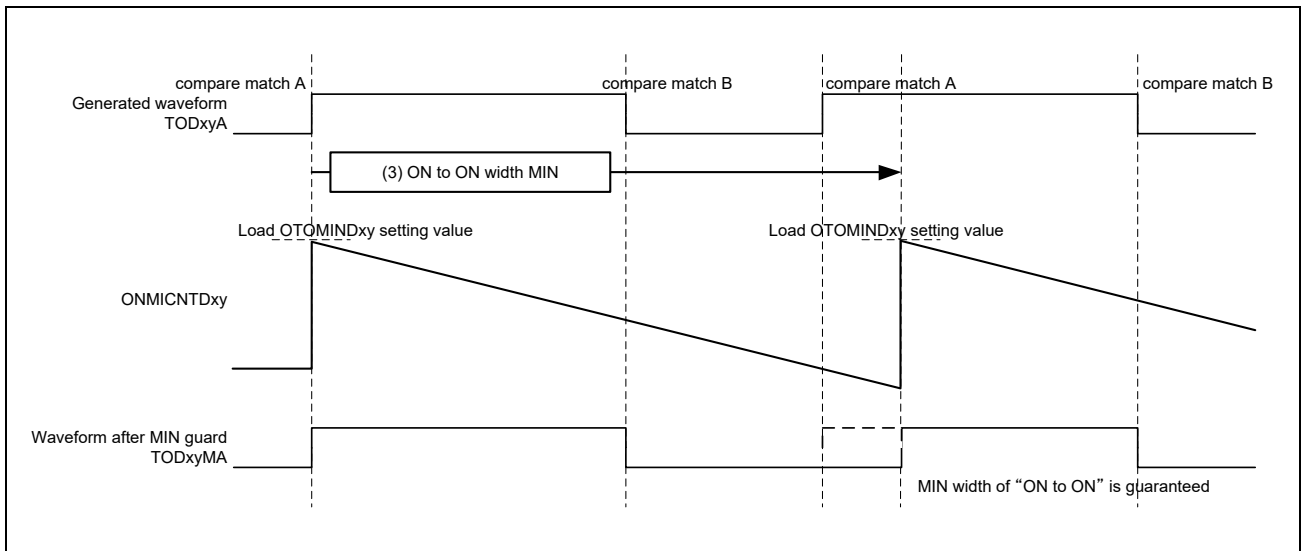


Figure 30.70 MIN Guard Function ON to ON Width MIN Correction Operation Diagram (TODxyA Correction)

Figure 30.71 shows the output waveform when MIN guard correction with ON to ON width is performed on TODxyB.

In the correction operation for TODxyB, the downcounter DCNTDxy starts down-counting at the timing from OFF to ON of the ON to ON width MIN correction waveform and guarantees the Max value of the ON width of the output waveform.

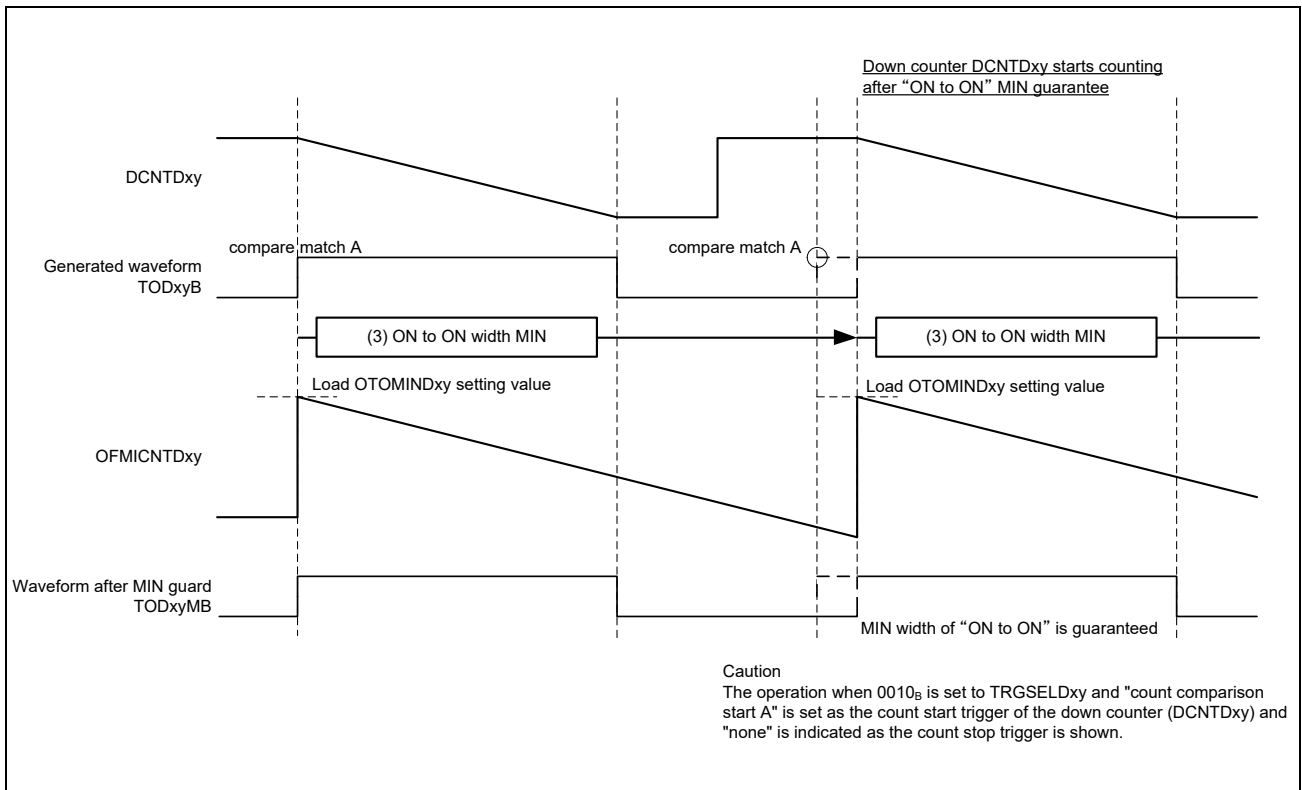


Figure 30.71 MIN Guard Function ON to ON Width MIN Correction Operation Diagram (TODxyB Correction)

Figure 30.72 shows the output waveform when MIN guard correction of ON width and MIN guard correction of OFF width are performed for TODxyA and TODxyB.

Each compensation counter loads the setting value from OFF to ON or ON to OFF timing of the output waveform after MIN guard.

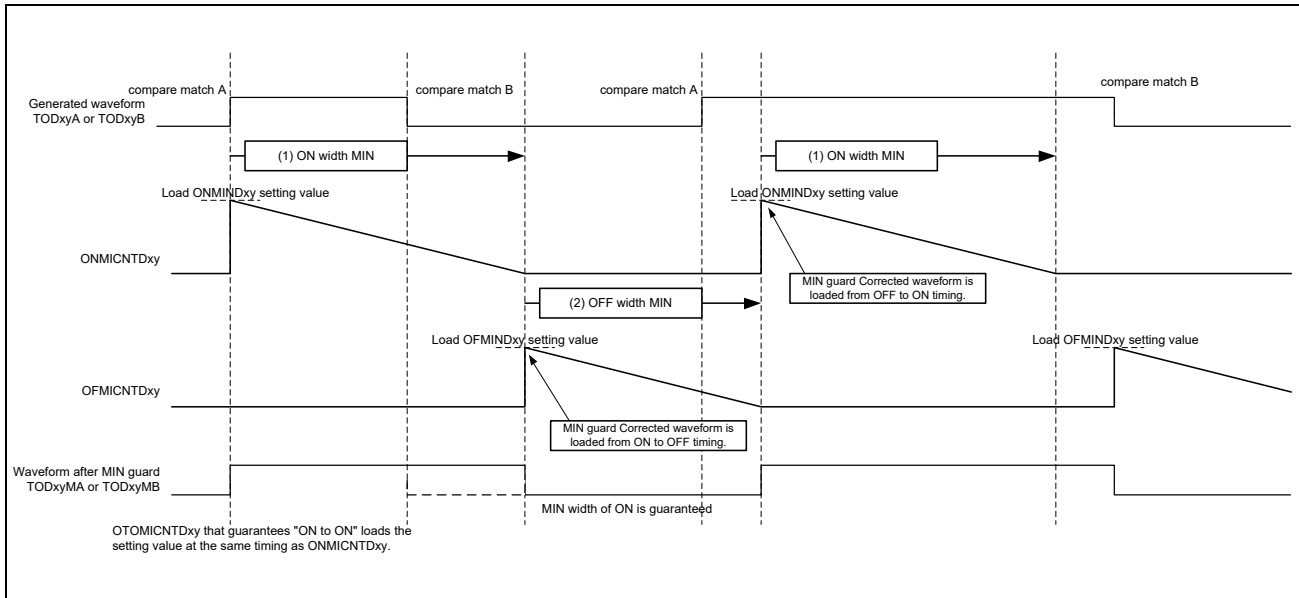


Figure 30.72 MIN Guard Function ON Width, OFF Width MIN Correction Operation Diagram

Figure 30.73 shows the output waveform when MIN guard correction of OFF width and MIN guard correction of ON to ON width are performed for TODxyA.

Both corrections are always valid because MIN guard correction of OFF width and MIN guard correction of ON to ON width individually perform correction.

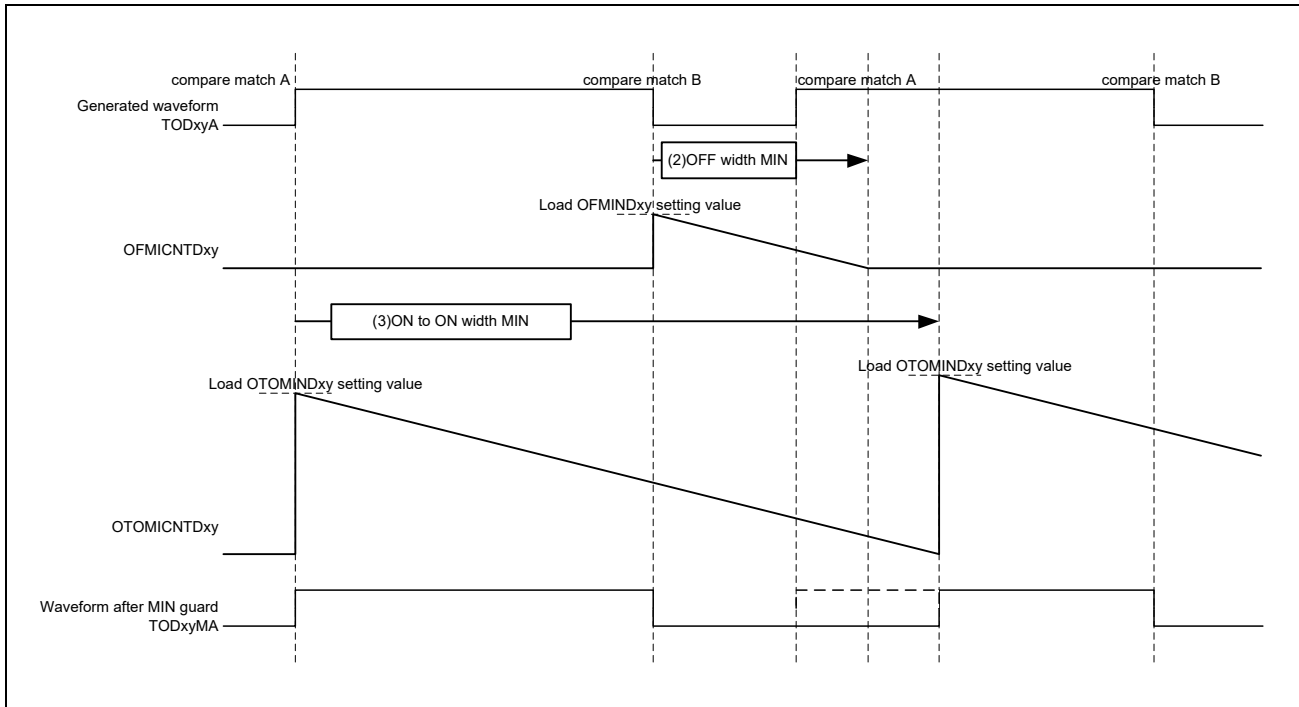


Figure 30.73 MIN guard function OFF width, ON to ON width MIN correction operation diagram

Caution

For the above operation, the MIN guard correction setting OTOMINDxy of the ON to ON width has no substantial meaning unless this set the sum of ON width MIN guard correction ONMINDxy and OFF width MIN guard correction.

Figure 30.74 shows the operation when ONMINDxy is changed during operation.

ONMICNDxy has a down counter configuration, and when setting value is changed during operation, the new setting value is loaded into ONMICNTDxy at the timing when the output waveform after the next MIN guard turns from OFF to ON.

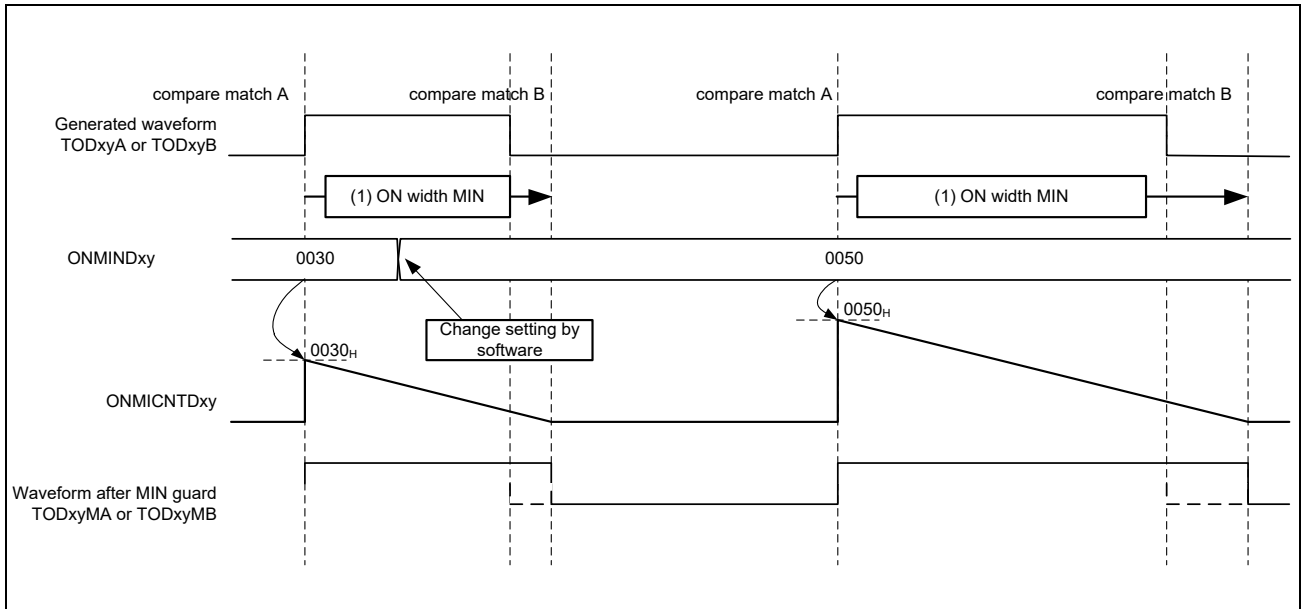


Figure 30.74 Change Setting while MIN Guard Function is Running

Figure 30.75 shows the operation when the generated waveform is turned from OFF to ON again while ON width is being corrected.

The MIN guard function does not hold the number of pulses input during the ON - MIN guard, so even if a waveform is input multiple times during the ON - MIN guard, it will be output as one pulse.

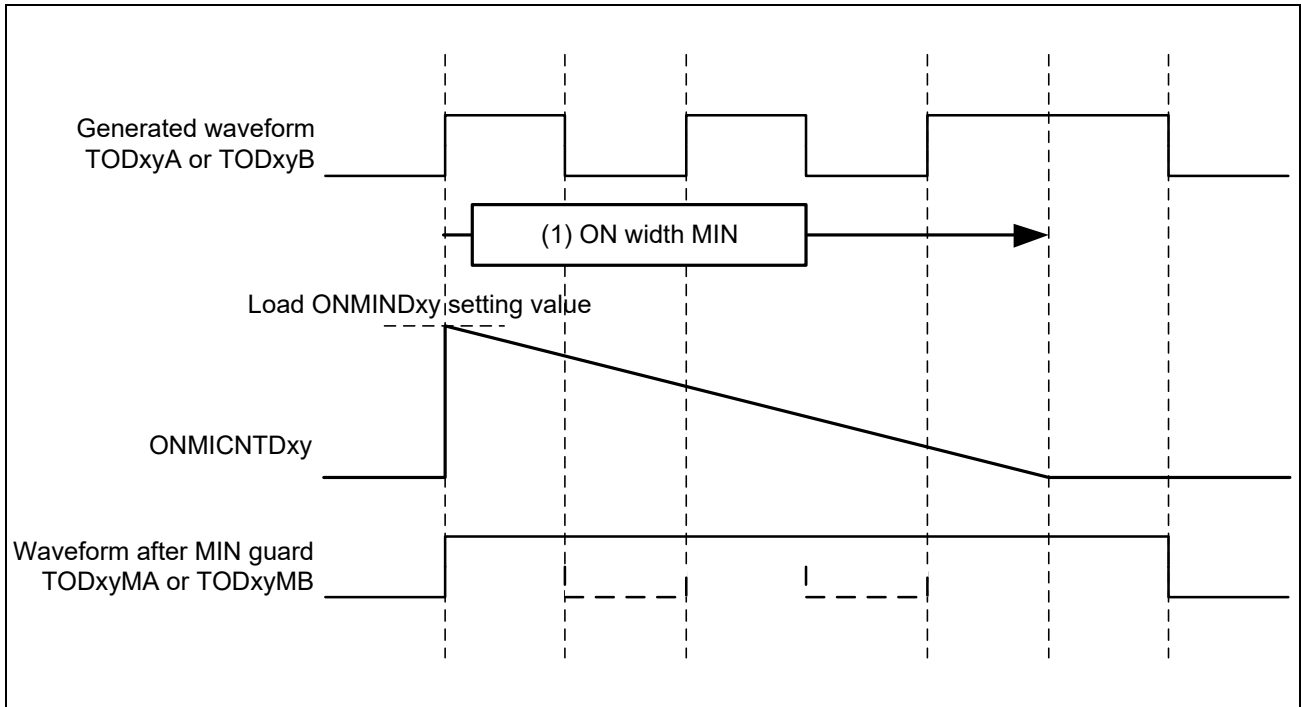


Figure 30.75 MIN Guard Regeneration of Correcting Waveform

Figure 30.76 shows the detailed timing when MIN guard correction is applied to TOD00A.

Interrupts are output from TOD00MA OFF to ON and ON to OFF timing after MIN guard correction, and the values of TCNT1D0 and TCNT2D0 are captured to ONCAP1D00, ONCAP2D00, OFCAP1D00, and OFCAP2D00.

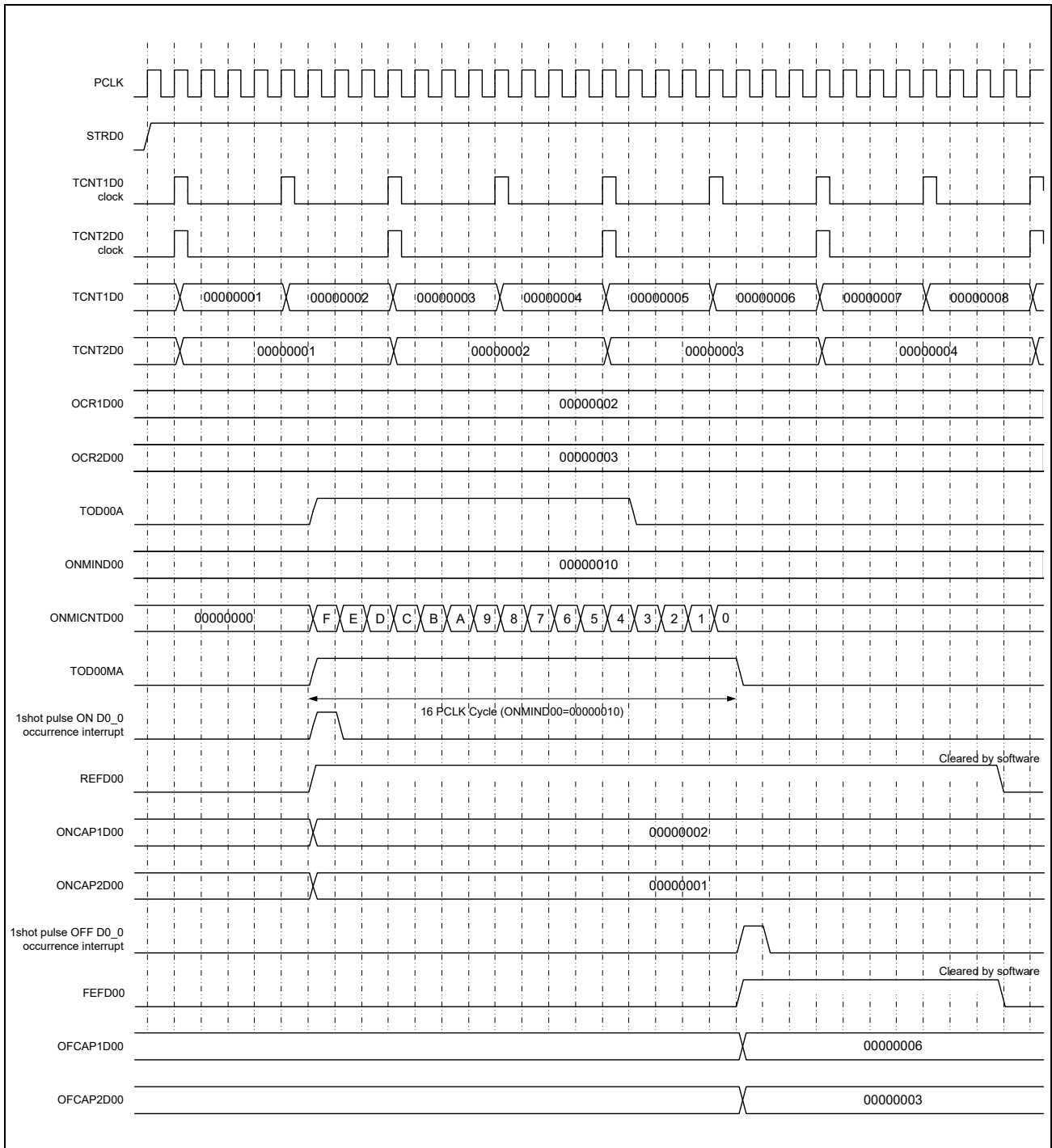


Figure 30.76 Detailed Timing Diagram of MIN Guard Function

30.9 Timer E

30.9.1 Overview of Operation

The timer E block is a PWM output timer that consists of SB_E timer E subblocks.

Timer E subblocks realize the following features:

- Output of waveform with a duty cycle of 0 to 100% by setting cycle-setting register and duty cycle setting register
- The values of the cycle-setting and duty-cycle-setting registers are updated every PWM cycle. The values in the cycle reload register and duty cycle reload register are reloaded as update data. The reloading function can be enabled and disabled.
- Writing 00000000_H to the counter can forcibly end PWM cycle.
- The shutoff input pins (POEx) shut off the PWM output. (They can also be used as the timer F input pins.) This feature is supported by subblocks E0 to E5. Other subblocks does not support this feature.
- On-state duty and off-state duty modes available
- Interrupt requests can be issued on cycle match (compare match between cycle-setting register and timer counter), that is, interrupts are issued every cycle.
- Compare matches of the duty cycle setting register generate interrupt requests periodically.
- Compare matches between TCNTExy and CYLRExy can be generated forcibly by setting the forced compare match trigger register.

A timer E subblock Ex consists of 4 channels. Components of each channel include a 24-bit up counter (TCNTExy), a 24-bit cycle-setting register (CYLRExy), a 24-bit duty-cycle-setting register (DTRExy), a 24-bit cycle reload register (CRLDExy), a 24-bit duty-reload register (DRLDExy), and a controller.

Subblocks E0 to E5 have a shutoff input pin (POEx), and each channel of each subblock has a PWM output pin (TOExy). (The default value of TOExy: 0)

POEx can also be used as TIF0A to TIF2A and TIF0B to TIF2B of timer F. For timer E, signals of TIF0A to TIF2A and TIF0B to TIF2B that have been processed by a noise canceler are used, which means that their noise canceler feature can be used. To use the noise canceler for the shutoff input signal of the PWM output shutoff feature, the timer F noise canceler setting is necessary as well as the timer E setting. For more about the setting, see **Section 30.10, Timer F**.

Table 30.154 shows the correspondence between TIF0A/B to TIF2A/B and POEx.

Table 30.154 Correspondence between Timer F Primary Inputs and Timer E Shutoff Inputs

Timer F Primary Inputs	Timer E Shutoff Inputs
TIF0A	POE0
TIF1A	POE1
TIF2A	POE2
TIF0B	POE3
TIF1B	POE4
TIF2B	POE5

Be noted that the timer F measurement signals become the shutoff input signals when TIF0A/B to TIF2A/B are used as POEx inputs and the corresponding timer F subblocks are operated simultaneously.

Figure 30.77 shows a block diagram of timer E.

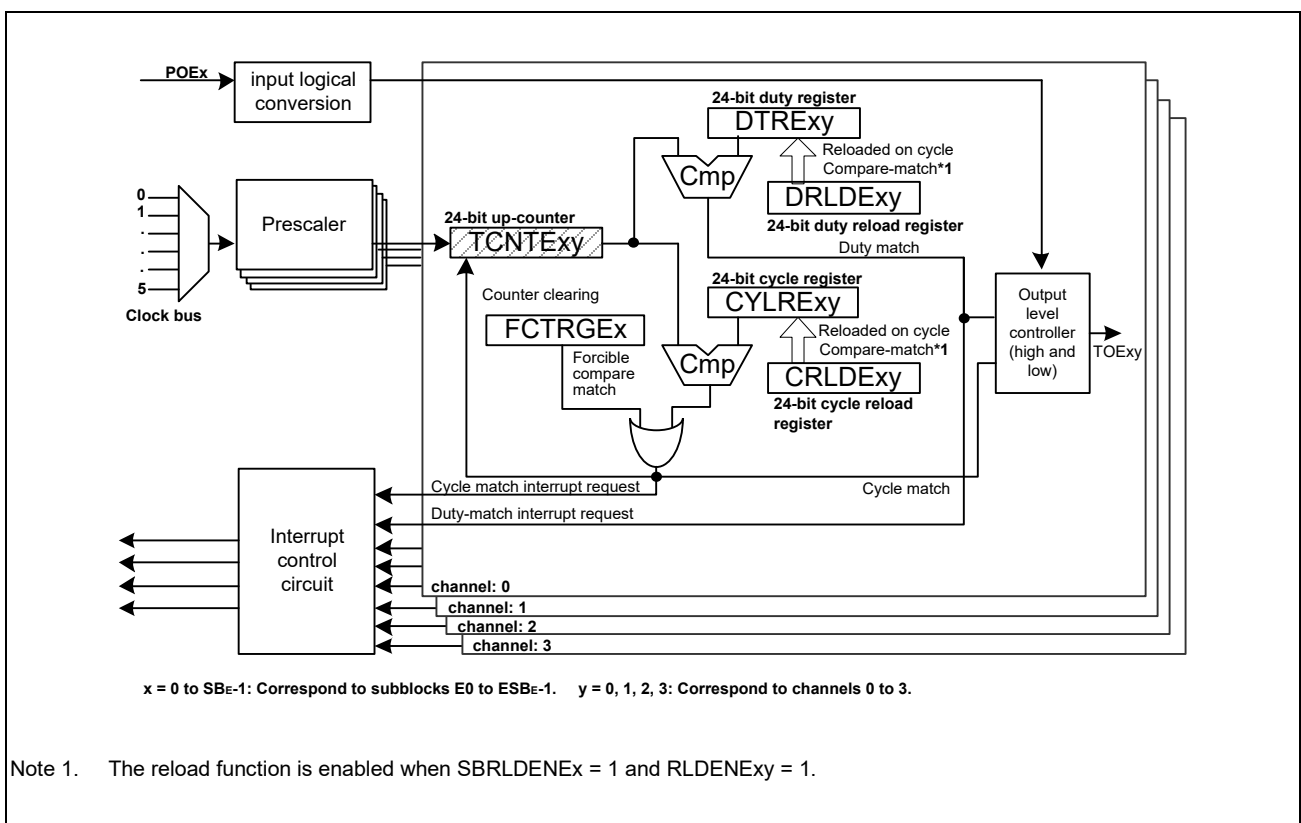


Figure 30.77 Block Diagram of Timer E Subblock

30.9.2 Registers Related to Timer E

30.9.2.1 TSTRE — Timer Start Register E

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	STRE8	STRE7	STRE6	STRE5	STRE4	STRE3	STRE2	STRE1	STRE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.155 TSTRE Register Contents

Bit Position	Bit Name	Function
15-SB _E	—	These bits are not used. When read, “0” is always returned. When writing, always write 0.
SB _E -1 to 0	STRE _x	Subblock Ex Start 0: Subblock Ex is disabled 1: Subblock Ex is enabled

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

TSTRE controls subblocks E0 to ESB_E-1.

The timer E counters run when the timer E enable bit (TEE) in the ATU master enable register (ATUENR), timer start register E (TSTRE), and subblock starting register E (SSTRE) must be set.

TSTRE is initialized to 0000_H after reset.

CAUTION

To synchronize timer operation for all channels in the subblock, as well as an SSTRE_x register, use the TSTRE register to enable or disable operation for each subblock.

30.9.2.2 SBRLNE — Subblock Reload Enable Register E

Value after reset: 01FF_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SBRLD ENE8	SBRLD ENE7	SBRLD ENE6	SBRLD ENE5	SBRLD ENE4	SBRLD ENE3	SBRLD ENE2	SBRLD ENE1	SBRLD ENE0
Value after reset	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.156 SBRLNE Register Contents

Bit Position	Bit Name	Function
15-SB _E	—	These bits are not used. When read, “0” is always returned. When writing, always write 0.
SB _E -1 to 0	SBRLDEN _{Ex}	Subblock Ex reload enable 0: The reload function for subblock Ex is disabled. 1: The reload function for subblock Ex is enabled.

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

The subblock reload enable register E (SBRLNE) is an 8-bit/16-bit readable/writable register enables and disables the reload function. The reload function is enabled when both the SBRLDEN_{Ex} and RLDEN_{Exy} bits are set to “1”, and is disabled in the other settings.

SBRLNE is initialized to 01FF_H after reset.

Subblock Ex Reload Enable Bit (SBRLDEN_{Ex})

When the SBRLDEN_{Ex} and RLDEN_{Exy} bits are set to “1” to enable the reload function, the values of the cycle reload register (CRLDE_{xy}) and the duty reload register (DRLDE_{xy}) are transferred to the cycle-setting register and duty cycle setting register, respectively, upon a cycle match between the timer counter (TCNTE_{xy}) and the cycle-setting register (CYLRE_{xy}).

30.9.2.3 SSTREx — Subblock Starting Registers Ex

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SSTREx3	SSTREx2	SSTREx1	SSTREx0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

Table 30.157 SSTREx Register Contents

Bit Position	Bit Name	Function
7 to 4	—	These bits are not used. When read, "0" is always returned. When writing, always write 0.
3	SSTREx3	Counter Ex3 start 0: Counter of channel 3 in subblock Ex is disabled 1: Counter of channel 3 in subblock Ex is enabled
2	SSTREx2	Counter Ex2 start 0: Counter of channel 2 in subblock Ex is disabled 1: Counter of channel 2 in subblock Ex is enabled
1	SSTREx1	Counter Ex1 start 0: Counter of channel 1 in subblock Ex is disabled 1: Counter of channel 1 in subblock Ex is enabled
0	SSTREx0	Counter Ex0 start 0: Counter of channel 0 in subblock Ex is disabled 1: Counter of channel 0 in subblock Ex is enabled

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

The subblock starting registers Ex (SSTREx) are 8-bit readable/writable registers. These registers control stop/start of timer counters of 4 channels of each subblock. This control is effective only for subblocks enabled by the timer start register (TSTRE).

SSTREx are initialized to 00_H after reset.

Counter Exy Start Bit (SSTRExy)

Enables and disables timer counter Exy (TCNTExy).

SSTRExy	Function
0	Counting of TCNTExy is disabled. (Default)
1	Counting of TCNTExy is enabled.

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

When this bit is cleared to 0, TCNTExy is disabled. TCNTExy retains the previous value while it is stopped. When this bit is set to 1 again, TCNTExy is resumed from the retained value.

Even if the counter Exy start bit is set to 1, counting does not start unless the enable bit of timer E is set to 1 by the ATU master enable register (ATUENR).

CAUTION

The prescalers run independently of the setting of the SSTRExy bit and are not initialized when TCNTExy starts counting. Therefore, during the time between the activation and the start of actual count operation by the above counter, hardware-related uncertainty shorter than the period of selected count source (resolution) accompanies.

30.9.2.4 PSCREx — Prescaler Registers Ex

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	PSCE _x [2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

Table 30.158 PSCREx Register Contents

Bit Position	Bit Name	Function
7 to 3	—	These bits are not used. Fix these bits to 0.
2 to 0	PSCE _x [2:0]	Division ratio These bits set the division ratio of the prescaler.

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

Prescaler registers Ex are 8-bit readable/writable registers.

Each subblock of timer E has one prescaler Ex that divides the frequency of the clock supplied via the clock bus. The register sets the division ratio of the prescaler Ex.

When the value in prescaler register Ex (PSCRE) is changed, the prescaler Ex updates the value on its underflow. Timer counter Exy (TCNTE_{xy}) (y = 0 to 3) in the same subblock is driven by the clock output from prescaler Ex.

The settable value in prescaler register Ex (PSCRE) ranges from 0_H to 7_H. The division ratio is given below.

$$\text{Division ratio of prescaler} = \frac{1}{\text{PSCE}_{x[2:0]} + 1} \quad (\text{setting range: } 1/1 \text{ to } 1/8)$$

A duty cycle of 50% for the prescaler Ex output clock is not guaranteed. The high level width is equal to the cycle of the PCLK and a low level is output in the remaining cycle of the prescaler E output clock.

Prescaler Ex runs when the TEE bit in the ATU master enable register (ATUENR) and the subblock Ex start bit (STRE_x) in timer start register E (TSTRE) are both set to 1.

30.9.2.5 PSCCRExy — Prescaler Channel Registers Exy

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	PSCCExy[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3

Table 30.159 PSCCRExy Register Contents

Bit Position	Bit Name	Function
7 to 0	PSCCExy[7:0]	Division Ratio These bits set the division ratio of the prescaler.

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Prescaler channel registers Exy are 8-bit readable/writable registers.

Each channel of timer E has one prescaler Exy that divides the frequency of the clock supplied via the clock bus. The register sets the division ratio of the prescaler Exy.

When the value in prescaler channel register Exy is changed, the prescaler Exy updates the value on its underflow. Timer counter E (TCNTExy) (y = 0 to 3) in the same channel is driven by the clock output from prescaler Exy.

The settable value in prescaler channel register Exy ranges from 00_H to FF_H. The division ratio is given below.

$$\text{Division ratio of prescaler} = \frac{1}{\text{PSCCExy}[7:0] + 1} \quad (\text{setting range: } 1/1 \text{ to } 1/256)$$

A duty cycle of 50% for the prescaler Exy output clock is not guaranteed. The high level width is equal to the cycle of the PCLK and a low level is output in the remaining cycle of the prescaler E output clock.

Prescaler Exy runs when the TEE bit in the ATU master enable register (ATUENR) and the subblock Ex start bit (STREx) in timer start register E (TSTRE) are both set to 1. The setting of the counter Exy start bit (SSTRExy) in the subblock start register Ex (SSTREx) does not affect operation of the prescaler Exy.

30.9.2.6 TCREx — Timer Control Register Ex

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	PSCSEL	—	—	—	—	CKSELEX[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R/W	R/W	R/W

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

Table 30.160 TCREx Register Contents

Bit Position	Bit Name	Function
7	PSCSEL	Prescaler select 0: Select the prescaler registers Ex (PSCREx: 3-bit settings for subblock units). 1: Selects the prescaler channel registers Exy (PSCCRExy: 8 bit settings for channel units).
6 to 3	—	These bits are not used. When read, "0" is always returned. When writing, always write 0.
2 to 0	CKSELEX[2:0]	TCNTExy clock select These bits select the counter clock of prescaler E from clock-bus lines 0 to 5.

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Timer control registers Ex (TCREx) are 8-bit readable/writable registers that select the counter clock of prescaler E from clock-bus lines 0 to 5. 24-bit up counter (TCNTExy) is driven by the clock output from prescaler E.

Prescaler Select (PSCSEL)

The prescaler can be selected in subblock units or in channel units. This selection makes a difference in the range of division ratios that can be selected for the prescaler.

Subblock Ex Clock Select (CKSELEX[2:0])

The subblock Ex clock select (CKSELEX) selects the count source of the prescaler of subblock Ex.

CKSELEX			Function
[2]	[1]	[0]	
0	0	0	Clock-bus line 0 is selected as counter clock of prescaler E (Default)
0	0	1	Clock-bus line 1 is selected as counter clock of prescaler E
0	1	0	Clock-bus line 2 is selected as counter clock of prescaler E
0	1	1	Clock-bus line 3 is selected as counter clock of prescaler E
1	0	0	Clock-bus line 4 is selected as counter clock of prescaler E
1	0	1	Clock-bus line 5 is selected as counter clock of prescaler E
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

30.9.2.7 RLDCREx — Reload Control Registers Ex

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	RLDENEx3	RLDENEx2	RLDENEx1	RLDENEx0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

Table 30.161 RLDCREx Register Contents

Bit Position	Bit Name	Function
7 to 4	—	These bits are not used. When read, “0” is always returned. When writing, always write 0.
3 to 0	RLDENEx3 to RLDENEx0	Reload enable Ex3 to Ex0 0: Reload function on cycle match is disabled. 1: Reload function on cycle match is enabled.

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Reload control registers Ex (RLDCREx) are 8-bit readable/writable registers. RLDENExy enables and disables the reload function. The reload function is enabled when both the SBRLDENEx and RLDENExy bits are set to “1”, and is disabled in the other settings.

RLDCREx is initialized to 00_H after reset.

Reload Enable Bit (RLDENExy)

When the SBRLDENEx and RLDENExy bits are set to “1” to enable the reload function, the values of the cycle reload register Exy(CRLDExy) and the duty reload register Exy(DRLDExy) are transferred to the cycle-setting register and duty cycle setting register, respectively, upon a cycle match between the timer counter Exy(TCNTExy) and the cycle-setting register Exy(CYLRExy).

30.9.2.8 POECREx — Output Shutoff Control Register Ex

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POECRKEY[7:0]							—	—	—	POE POLx	POE ENx3	POE ENx2	POE ENx1	POE ENx0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	R	R	R	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_E-1 (MAX:5): Corresponding to subblocks E0 to ESB_E-1

Note 1. Values written to the write key code are not retained.

Table 30.162 POECREx Register Contents

Bit Position	Bit Name	Function
15 to 8	POECRKEY[7:0] Write key code	9C _H : Values of POEEN _{xy} and POEPOL _x can be changed. Other than 9C _H : Values of POEEN _{xy} and POEPOL _x cannot be changed.
7 to 5	—	These bits are not used. When read, "0" is always returned. When writing, always write 0.
4	POEPOL _x	Shutoff input active level selection 0: Shutoff input is active low. 1: Shutoff input is active high.
3	POEEN _{x3}	Output shutoff enable/disable
2	POEEN _{x2}	0: Output shutoff is disabled.
1	POEEN _{x1}	1: Output shutoff is enabled.
0	POEEN _{x0}	

Note: x = 0 to SB_E-1 (MAX:5): Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Output shutoff control register Ex (POECREx) is a 16-bit readable/writable register. To write a value to this register, the data to be written must be provided and the write key code must be set in bits 15 to 8. This feature is not supported for subblocks 6 to 8 because there is no POE₆ to POE₈ inputs.

POECREx is initialized to 0000_H after reset.

Write Key Code (POECRKEY)

This code controls the write permission for bits POEEN_{xy} (y = 0 to 3) and POEPOL_x. Setting is required to bits 15 to 8 as well as the data to be written to bits POEEN_{xy} (y = 0 to 3) and POEPOL_x. Data written to these bits is not retained; Value 00_H is always returned when being read.

Shutoff Input Active Level Selection (POEPOLx)

Controls the active level of the shutoff input (POE_x). The setting of this bit applies to all channels of the subblock.

Output Shutoff Enable/Disable (POEENxy)

Controls disable/enable of the output shutoff feature. This bit can be set for each channel.

For details of the output shutoff feature, see **Figure 30.81** in **Section 30.9.3, Details of Operation**.

30.9.2.9 SOLVLE_x — Output Shutoff Level Setting Register Ex

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWMSLV _{x3}	PWMSLV _{x2}	PWMSLV _{x1}	PWMSLV _{x0}
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note: x = 0 to SB_E-1 (MAX:5): Corresponding to subblocks E0 to ESB_E-1

Table 30.163 SOLVLE_x Register Contents

Bit Position	Bit Name	Function
7 to 4	—	These bits are not used. When read, "0" is always returned. When writing, always write 0.
3	PWMSLV _{x3}	Output level H/L selection
2	PWMSLV _{x2}	0: The low level is output to TOE _x y when output is shutoff.
1	PWMSLV _{x1}	1: The high level is output to TOE _x y when output is shutoff.
0	PWMSLV _{x0}	

Note: x = 0 to SB_E-1 (MAX:5): Corresponding to subblocks E0 to ESB_E-1

The output shutoff level setting register Ex (SOLVLE_x) is an 8-bit readable/writable register.

SOLVLE_x is initialized to 00_H after reset.

Output Level H/L Selection (PWMSLV_{xy})

This bit specifies the output level of TOE_xy in the output shutoff state.

30.9.2.10 TSREx — Timer Status Registers Ex

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DMFEx 3	DMFEx 2	DMFEx 1	DMFEx 0	OVFEx 3	OVFEx 2	OVFEx 1	OVFEx 0	CMFEx 3	CMFEx 2	CMFEx 1	CMFEx 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

Table 30.164 TSREx Register Contents

Bit Position	Bit Name	Function
15 to 12	—	These bits are not used. When read, “0” is always returned. When writing, always write 0.
11 to 8	DMFEx3 to DMFEx0	Duty match flags Ex3 to Ex0 0: No duty match has occurred. 1: A Duty match has occurred.
7 to 4	OVFEx3 to OVFEx0	Overflow flags Ex3 to Ex0 0: Counter E has not overflowed 1: Counter E has overflowed
3 to 0	CMFEx3 to CMFEx0	Cycle match flags Ex3 to Ex0 0: Cycle match has not occurred 1: Cycle match has occurred

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

The timer status register Ex (TSREx) is an 8-bit or a 16-bit read-only register. In subblocks E0 to E8, these registers indicate occurrence of compare match (cycle match) of cycle-setting registers (CYLRExy) and timer counters (TCNTExy), compare match (duty match) of duty cycle setting registers (DTRExy) and timer counters (TCNTExy), and overflow of TCNTExy.

The overflow flag is the status flag for indicating the occurrence of overflow. It does not generate interrupts. The cycle match flag and the duty match flag are status flags for requesting interrupts. Interrupts can be issued by setting the corresponding bits of the TIEREx register. When interrupt is permitted at the TIEREx register, setting the corresponding flags issues interrupt requests. Setting a bit of the timer status clear register Ex (TSCREx) clears the corresponding flag. When set and clear competed, clear is given priority to. Also, by setting the corresponding bit of the TIEREx register, it is possible to generate a sDMA / DTS transfer request. If a duty match (or cycle match) occurs again while the duty match flag DMFExy (or the cycle match flag CMFExy) is set, a transfer request to sDMA / DTS can not be generated. To generate a transfer request to sDMA / DTS, it is necessary to clear the flag by “CPU interrupt processing” or “sDMA / DTS chain transfer”.

TSREx is initialized to 0000_H after reset.

Duty Match Flag Exy (DMFExy)

DMFExy	Function
0	[Clearing condition] When 1 is written to the DMFCExy bit of the timer status clear register Ex (TSCREx). (Default)
1	[Setting condition] When the counter clock is input in the condition where the timer counter Exy (TCNTExy) value coincides with the duty cycle setting register Exy (DTRExy) value.

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

This flag cannot be set to 1 or 0 by software. The next duty match can be processed even if DMFExy is 1 (even if the flag is not cleared). In this case, 1 is rewritten to these bits. However, an interrupt request or sDMA / DTS transfer request can be realized only when DMFExy is 0 (only when the flag is cleared).

Cycle Match Flag Exy (CMFExy)

CMFExyl	Function	
0	[Clearing conditions] When 1 is written to the CMFCExy bit of the timer status clear register Ex (TSCREx).	(Default)
1	[Setting condition 1] When timer counter Exy (TCNTExy) is incremented while it is the same value as cycle-setting register Exy (CYLRExy) [Setting condition 2] When a compulsion cycle match (1 write to FCMTxy) occurred.	

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

These bits cannot be set to 1 or 0 by software. Even if these bits are 1, meaning that the flag has not been cleared, the next cycle match can be input. In this case, 1 is rewritten to these bits. However, an interrupt request or sDMA / DTS transfer request can be realized only when CMFExy is 0 (only when the flag is cleared).

Overflow Flag Exy (OVFExy)

OVFExy	Function	
0	[Clearing conditions] When 1 is written to the OVFCExy bit of the timer status clear register Ex (TSCREx).	(Default)
1	[Setting condition] When the 24 higher-order bits of timer counter Exy (TCNTExy) value changes from FFFF FF _H to 0000 00 _H .	

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

These bits are set to 1 when timer counter Exy (TCNTExy) overflows. This flag cannot be set to 1 or 0 by software.

Overflow occurs when the 24 higher-order bits of timer counter Exy is incremented while it is FFFF FF_H. Writing 0000 00_H to the timer counter Exy has no effect on these bits.

When writing a value to the timer counter and incrementation occur simultaneously while the counter is FFFF FF_H, the overflow flag is set to 1 but the counter value is changed to the written value instead of 0000 00_H.

As to cycle match, since the counter is cleared to 0000 01_H, overflow will not occur. However, it may occur when the value in the cycle-setting register is changed during counter in operation.

When overflow and cycle match occur simultaneously, overflow is not detected (the counter is incremented while it is FFFF FF_H and CYLREx is FFFF FF_H). In this case, only the processing upon detection of cycle match is performed. If counter value is 0000 01_H and the reload function is enabled, cycle reload or duty-cycle reload is performed. When overflow and duty match occur simultaneously, the same operation is performed because a cycle match occurs with duty cycle of 100%.

30.9.2.11 TSCREx — Timer Status Clear Registers Ex

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DMFCE x3	DMFCE x2	DMFCE x1	DMFCE x0	OVFCE x3	OVFCE x2	OVFCE x1	OVFCE x0	CMFCE x3	CMFCE x2	CMFCE x1	CMFCE x0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W	W	W	W	W	W	W	W	W

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

Table 30.165 TSCREx Register Contents

Bit Position	Bit Name	Function
15 to 12	—	These bits are not used. When read, “0” is always returned. When writing, always write “0”.
11 to 8	DMFCE _{xy}	Duty match flag clear enable Ex _y 0: Disabled (Default) 1: Clears DMFCE _{xy} of timer status register Ex (TSREx) to 0.
7 to 3	OVFCE _{xy}	Overflow flag clear enable Ex _y 0: Disabled (Default) 1: Clears OVFCE _{xy} of timer status register Ex (TSREx) to 0.
3 to 0	CMFCE _{xy}	Cycle match flag clear enable Ex _y 0: Disabled (Default) 1: Clears CMFCE _{xy} of timer status register Ex (TSREx) to 0.

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

The timer status clear registers Ex (TSCREx) are 8-bit or a 16-bit readable/writable registers. When read, “0” is always returned. TSCREx specifies flag clearing in the events of overflow of timer counters Ex_y (TCNTE_{xy}), cycle match of TCNTE_{xy} and cycle-setting registers Ex_y (CYLRE_{xy}), and duty match of TCNTE_{xy} and duty cycle setting registers Ex_y (DTRE_{xy}).

TSCREx is readable/writable in 16-bit units. When read, “0” is always returned.

TSCREx is initialized to 0000_H after reset.

Duty Match Flag Clear Enable Ex_y (DMFCE_{xy})

This flag specifies flag clearing at the time of duty match of TCNTE_{xy} and the duty cycle setting register Ex_y (DTRE_{xy}).

Setting this bit enables clearing of the duty match flag Ex_y (DMFCE_{xy}) of the timer status register Ex (TSREx). When this bit is read, “0” is always returned.

DMFCE _{xy}	Function
0	Disabled (Default)
1	Clears DMFCE _{xy} of timer status register Ex (TSREx) to 0.

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Overflow Flag Clear Enable Exy (OVFCExy)

This bit specifies flag clearing at the time of overflow of the timer counter Ex (TCNTExy).

Setting this bit enables clearing of the overflow flag Exy (OVFExy) of the timer status register Ex (TSREx). When this bit is read, “0” is always returned.

OVFCExy	Function	
0	Disabled	(Default)
1	Clears OVFExy of timer status register E (TSREx) to 0.	

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Cycle Match Flag Clear Enable Exy (CMFCExy)

This flag specifies flag clearing at the time of cycle match of TCNTExy and the cycle-setting register Exy (CYLRExy).

Setting this bit enables clearing of the cycle match flag Exy (CMFExy) of the timer status register Ex (TSREx). When this bit is read, “0” is always returned.

CMFCExy	Function	
0	Disabled	(Default)
1	Clears CMFExy of timer status register Ex (TSREx) to 0.	

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

30.9.2.12 TIEREx — Timer Interrupt Enable Registers Ex

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DMEE _x ₃	DMEE _x ₂	DMEE _x ₁	DMEE _x ₀	—	—	—	—	CMEE _x ₃	CMEE _x ₂	CMEE _x ₁	CMEE _x ₀
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

Table 30.166 TIEREx Register Contents

Bit Position	Bit Name	Function
15 to 12	—	These bits are not used. When read, “0” is always returned. When writing, always write 0.
11 to 8	DMEE _x ₃ to DMEE _x ₀	Duty match interrupt enable Ex3 to Ex0 Specifies enable/disable of interrupt requests issued by duty match flags DMFEx _y .
7 to 4	—	These bits are not used. When read, “0” is always returned. When writing, always write 0.
3 to 0	CMEE _x ₃ to CMEE _x ₀	Cycle match interrupt enable Ex3 to Ex0 Specifies enable/disable of interrupt requests issued by cycle match flags CMFEx _y .

The timer interrupt enable registers Ex (TIEREx) are 8-bit or a 16-bit readable/writable registers. TIEREx controls enable/disable of interrupts triggered by compare match (cycle match) of the cycle-setting register (CYLREx_y) and TCNTEx_y and compare match (duty match) of the duty cycle setting register (DTREx_y) and TCNTEx_y.

TIEREx is initialized to 0000_H after reset.

Duty Match Interrupt Enable Exy (DMEE_x_y)

This flag specifies enable/disable of interrupt requests triggered by duty match of DTREx_y.

Setting this bit enables interrupts issued by the duty match flag Ex (DMFEx_y) of the timer status register (TSREx).

DMEE _x _y	Function
0	Interrupts by DMFEx _y are disabled. (Default)
1	Interrupts by DMFEx _y are enabled.

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Cycle Match Interrupt Enable Exy (CMEE_x_y)

This flag specifies enable/disable of interrupt requests triggered by cycle matches of CYLREx_y.

A cycle match flag (CMFEx_y) of the timer status register (TSREx) can be output as an interrupt by setting this bit.

CMEE _x _y	Function
0	Interrupts by CMFEx _y are disabled. (Default)
1	Interrupts by CMFEx _y are enabled.

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

CAUTION

Interrupt requests by cycle match Ex (CMFExy) and duty match Exy (DMFExy) of timer subblock Ex are issued as the logical sum of CMFExy and DMFExy. Whether the interrupt request is triggered by a cycle match or a duty match of the counter can be determined by referencing TSREx. Clear the corresponding interrupt request flag of TSREx when processing the interrupt.

30.9.2.13 TOCREx — Timer Output Control Registers Ex

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TONEEx3	TONEEx2	TONEEx1	TONEEx0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

Table 30.167 TOCREx Register Contents

Bit Position	Bit Name	Function
7 to 4	—	These bits are not used. When read, “0” is always returned. When writing, always write 0.
3 to 0	TONEExy	TOExy output inversion select Inversion setting of the TOExy output signal (on-state duty/off-state duty selection)

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Timer output control registers Ex (TOCREx) are 8-bit readable/writable registers that select whether or not a signal on the output pin TOExy in each subblock Ex is inverted.

TOCREx is initialized to 00_H after reset.

TOExy Output Inversion Select (TONEExy)

These bits select whether to invert the PWM output signal from TOExy.

TONEExy	Function
0	Normal output from the output pin (TOExy) (Default)
1	Inverted output from the output pin (TOExy)

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

When the on-state duty/off-state duty setting of TOCREEx is changed, the TOExy output is inverted immediately after the change of the register setting by the next clock cycle (PCLK). The operating state of the timer counter (TCNTExy) has no effect on the mode switching.

The initial level on the PWM output pin is “0” (TONEExy = 0).

30.9.2.14 FCTRGE_x — Forced Compare Match Trigger Registers Ex

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	FCMT _{x3}	FCMT _{x2}	FCMT _{x1}	FCMT _{x0}
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

Table 30.168 FCTRGE_x Register Contents

Bit Position	Bit Name	Function
7 to 4	—	These bits are not used. When read, “0” is always returned. When writing, always write 0.
3 to 0	FCMT _{xy}	Forced cycle match trigger 0: disable (initial value) 1: Forcibly generate a cycle match between TCNT _E _{xy} and CYLRE _{xy} .

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1

The forced compare match trigger registers Ex (FCTRGE_x) are 8-bit writable registers. When FCTRGE_x is read, “0” is always returned. FCTRGE_x is used to specify whether to generate a cycle match between TCNT_E_{xy} and CYLRE_{xy} of each subblock forcibly.

FCTRGE_x is initialized to 00_H after reset.

CAUTION

This bit can be set to “1” only when the operation of the timer counter Exy (TCNT_E_{xy}) is stopped. (When any bit of TEE, STRE_x, or SSTRE_{xy} is 0).

30.9.2.15 TCNTExy — Timer Counters Exy

Value after reset: 0000 0100_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TCNTExy															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNTExy								—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Note 1. These bits are not used. Fix these bits to "0".

Timer counters Exy (TCNTExy) are 32-bit readable/writable registers allocated in channel y of subblock Ex. These registers are started by setting the TEE bit in the ATU master enable register (ATUENR), the subblock Ex start bit (STREx) in the timer start register (TSTRE), and the counter Exy start bit (SSTRExy) in subblock starting register (SSTREx). When the 8 lower-order bits are read, 0 is always read.

The counter clock is selected by the TCNTEx clock select bits (CKSELEx[2:0]) in timer control register Ex (TCREx), prescaler register (PSCREx) and prescaler channel registers Exy (PSCCRExy) of timer E.

When a cycle match with the cycle register Exy (CYLRExy) occurs, or when a cycle match between TCNTExy and CYLRExy is forced generated by setting the forced compare match trigger register Ex (FCTRGEx), The 24 higher-order bits of TCNTExy is initialized to 0000 01_H. For example, when the 24 higher-order bit value of the cycle-setting register is N and the counter value is to be incremented from N to N + 1, the counter value is changed to 1. This enables counting from 1 to N and PWM pulses with the cycle time of N is produced.

These counters can count from 0000 01_H to FFFF FF_H (when the 24 higher-order bit value of the cycle-setting register is FFFF FF_H).

When writing 0000 00_H to these counters, a PWM cycle is terminated and a new PWM cycle is started from 0000 01_H in the next clock cycle. While the counter value holds 0000 00_H, when TONEExy = 0, the PWM output retains the previous value and outputs a level of 1 at the beginning of the new cycle. When the PWM cycle is terminated before duty match, the duty cycle for that PWM cycle is 100% (1 is always output), that is, a level of 0 will not be output between PWM cycles. For details on writing 0000 00_H to these counters, see **Figure 30.79** and **Figure 30.80**.

When TCNTExy or CYLRExy is rewritten during the counter in operation, a cycle match may not occur even if the value of the counter reaches FFFF FF_H. In this case, the counter value is changed from FFFF FF_H to 0000 00_H in the next counter clock cycle. A PWM cycle is terminated in a way similar to writing 0000 00_H. The counter value is incremented to 0000 01_H and a new PWM cycle is started. When the reload function is enabled, reloading of the cycle or duty cycle is also performed.

TCNTExy is initialized to 0000 0100_H after reset.

30.9.2.16 CYLRE_{xy} — Cycle-Setting Registers Exy

Value after reset: FFFF FF00_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CYLRE _{xy}															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CYLRE _{xy}								—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Note 1. These bits are not used. Fix these bits to "0".

CYLRE_{xy} are 32-bit readable/writable registers that store the cycle of PWM. The settable 24 higher-order bit value of the ranges from 0000 01_H to FFFF FF_H. When the 8 lower-order bits are read, 0 is always read.

The value in CYLRE_{xy} is constantly compared with the value in the timer counter Exy (TCNTE_{xy}). When they match, TCNTE_{xy} is cleared to 0000 01_H. A compare match between CYLRE_{xy} and TCNTE_{xy} can be generated forcibly by setting the FCTRGE_x register. When the SBRLDEN_{Ex} bit in the subblock reload enable register E (SBRLENE) and the RLDEN_{Exy} bit in the reload control register Ex (RLDCRE_x) are set to 1, the values in the cycle reload register Exy (CRLDE_{xy}) and duty reload register Exy (DRLDE_{xy}) are transferred to cycle-setting register Exy (CYLRE_{xy}) and duty cycle setting register Exy (DTRE_{xy}), respectively. Writing 0000 00_H to the 24 higher-order bits of TCNTE_{xy} can forcibly end PWM cycle and start the same reloading as above at the next clock cycle. Interrupt request to CPU can be made by a compare match (cycle match) of the cycle-setting register.

To rewrite to CYLRE_{xy} during TCNTE_{xy} in operation, note that the value to be set may lead to malfunction. When TCNTE_{xy} in operation is rewritten, a cycle match between TCNTE_{xy} and DTRE_{xy} may not be detected and TCNTE_{xy} continues to be incremented even if the counter value exceeds the value in CYLRE_{xy}. In this case, unwanted PWM waveforms are output.

CYLRE_{xy} is initialized to FFFF FF00_H after reset.

30.9.2.17 DTRExy — Duty Cycle Setting Registers Exy

Value after reset: FFFF FF00_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTRExy															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTRExy								—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	0 ^{*1}	0 ^{*1}	0 ^{*1}	0 ^{*1}	0 ^{*1}	0 ^{*1}	0 ^{*1}	0 ^{*1}
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Note 1. These bits are not used. Fix these bits to "0".

DTRExy are 32-bit readable/writable registers that store the duty cycle of PWM. The settable 24 higher-order bit value of ranges from 0000 00_H to FFFF FF_H. When the 8 lower-order bits are read, 0 is always read.

The value in DTRExy is constantly compared with the value in timer counter Exy (TCNTExy). When they match, the output level on the pin for the corresponding channel becomes 0. When the values in CYLRExy and TCNTExy match while the SBRLDENEx bit in the subblock reload enable register E (SBRLENE) and the RLDENExy bit of the reload control register Ex (RLDCREx) are set to 1, the value in DRLDExy is reloaded to DTRExy. Writing 0000 00_H to TCNTExy can forcibly end PWM cycle and start the same reloading as above at the next clock cycle.

The settable value in DTRExy ranges from 0 to the value in CYLRExy. When 0 is set, the duty cycle is 0% and the same value as CYLRExy is set, the duty cycle is 100%. Do not set DTRExy to a value greater than CYLRExy.

To rewrite to DTRExy during TCNTExy in operation, note that the value to be set may lead to malfunction. When TCNTExy in operation is rewritten, a duty match may not be detected. In this case, unwanted PWM waveforms may be output.

DTRExy is initialized to FFFF FF00_H after reset.

30.9.2.18 CRLDE_{xy} — Cycle Reload Registers Exy

Value after reset: FFFF FF00_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRLDE _{xy}															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRLDE _{xy}								—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Note 1. These bits are not used. Fix these bits to "0".

Cycle reload registers xy (CRLDE_{xy}) are 32-bit readable/writable register, and can be set to 0000 01_H to FFFF FF_H as the cycle of PWM outputs. When the 8 lower-order bits are read, 0 is always read.

When the reload function is enabled (when the SBRLDEN_{Ex} bit in the subblock reload enable register (SBRLNE) and the RLDEN_{xy} bit in the reload control register (RLDCRE_{Ex}) are set to "1"), the value in this register is transferred to the cycle-setting register Exy (CYLRE_{xy}) on cycle match. Writing 0000 00_H to TCNTExy can forcibly end PWM cycle and start the same reloading as above at the next clock cycle.

CRLDE_{xy} is initialized to FFFF FF00_H after reset.

30.9.2.19 DRLDExy — Duty Reload Registers Exy

Value after reset: FFFF FF00_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DRLDExy															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRLDExy								—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Note: x = 0 to SB_E-1: Corresponding to subblocks E0 to ESB_E-1. y = 0, 1, 2, 3: Corresponding to channels 0 to 3.

Note 1. These bits are not used. Fix these bits to "0".

Duty reload registers Exy (DRLDExy) are a 32-bit readable/writable register, and can be set to 0000 00_H to FFFF FF_H as the duty cycle. When the 8 lower-order bits are read, 0 is always read.

When the reload function is enabled (when the SBRLDEN_{Ex} bit in the subblock reload enable register (SBRLNE) and the RLDE_{Nxy} bit in the reload control register (RLDCRE_{Ex}) are set to "1"), the value in this register is transferred to the duty cycle setting register Exy (DTRE_{Exy}) on cycle match. Writing 0000 00_H to TCNT_{Exy} can forcibly end PWM cycle and start the same reloading as above at the next clock cycle.

DRLDExy is initialized to FFFF FF00_H after reset.

30.9.3 Details of Operation

Timer E consists of 24-bit timer counter Exy (TCNTExy), 24-bit cycle-setting register Exy (CYLRExy), 24-bit duty cycle setting register Exy (DTRExy), 24-bit cycle reload register Exy (CRLDExy), and 24-bit duty reload register Exy (DRLDExy). Timer E can be used as a PWM timer.

TCNTExy starts counting up when a channel is selected by subblock starting register (SSTREx) after a subblock is selected by timer start register (TSTRE). When TONEExy = 0, a logical zero level is output on pin TOExy on the first edge of the counter clock after TCNTExy matches duty cycle setting register Exy (DTRExy), or a logical one level is output on pin TOExy on the first edge the counter clock after TCNTExy matches cycle-setting register Exy (CYLRExy). After a match with the cycle-setting register, cycle match occurs, the counter is set to 0000 01_H on the next counter clock edge and starts counting up again. A compare match between TCNTExy and the cycle match register Exy (CYLRExy) can be generated forcibly by setting the forced compare match trigger register Ex (FCTRGE_x).

Subsequently, duty match and cycle match are repeated, producing a PWM output on pin TOExy.

However, external output level retains the default value of “0” for one cycle which is from starting up the counter to the first cycle match. In addition, when a forced cycle match is generated by the FCTRGE_x register while the counter is stopped, the inactive level is output when the duty is 0%, and the active level is output when the duty is not 0%.

For duty determination, when the reload function is enabled (SBRLDENEx = 1 and RLDENExy = 1), it judges whether the duty of the timer E is set to 0% by the duty reload register Exy (DRLDExy). When the reload function is disabled (SBRLDENEx = 0 or RLDENExy = 0), it judges by the duty register Exy (DTRExy).

The settable PWM cycle ranges from 0000 01_H to FFFF FF_H. The settable duty cycle ranges from 0% to 100%. The values set in the cycle-setting register and duty cycle setting register are used as the cycle width and duty cycle width, respectively. When the duty cycle setting register is set to 0000_H, the output level is 0 and remains unchanged (when TONEExy = 0, duty cycle = 0%). When the values in duty cycle setting register and cycle-setting register are the same, the output level is 1 and remains unchanged (when TONEExy = 1, duty cycle = 100%). The value in duty cycle setting register must be equal to or less than the value in cycle-setting register.

An interrupt request can be generated per cycle by a compare match (cycle match) of the cycle-setting register.

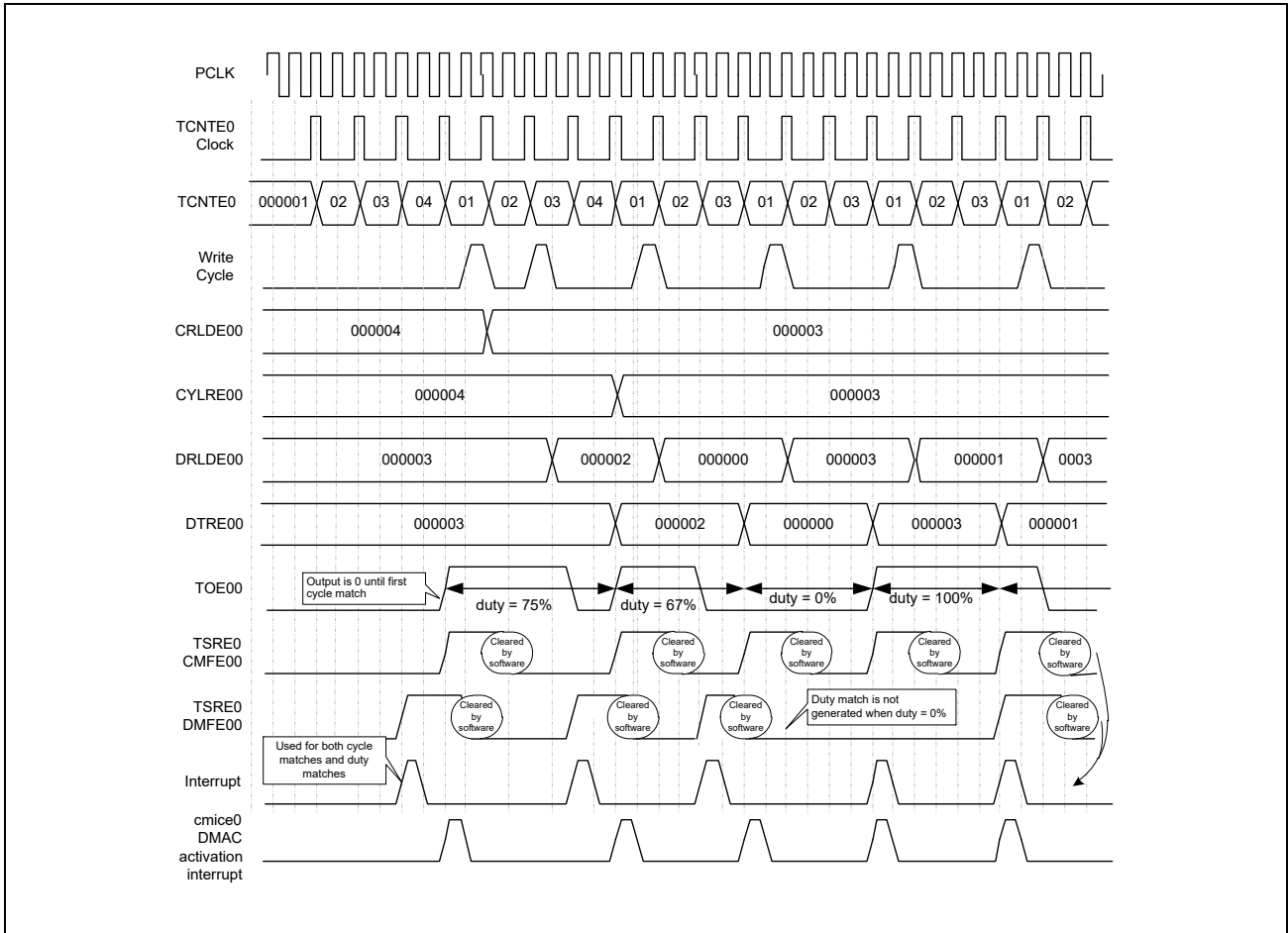


Figure 30.78 Operation of PWM (1)

The duty cycle setting and cycle-setting registers have respective reload registers. When the values in the up-counter and cycle-setting register match, the values in the duty cycle reload and cycle reload registers are loaded to the duty cycle setting and cycle-setting registers, respectively. The loaded data is updated in the next PWM cycle after loading. The reload function is enabled and disabled by the SBRLDEN_{Ex} bit in the subblock reload enable register (SBRLENE) and reload enable bit (RLDEN_{Ex}) in the reload control register (RLDCRE_{Ex}).

In timer E, a PWM output cycle is terminated by writing 0000 00_H to the counter (TCNTE_{Ex}). The counter is changed from 0000 00_H to 0000 01_H on the next counter clock and the counter is restarted. When the counter value is changed, the values in the duty cycle reload and cycle reload registers are loaded to the duty cycle setting and cycle-setting registers, respectively.

Figure 30.78 shows the operation of the PWM timer of channel 0, block E0, in which the duty cycle is changed from 75% to 67%, 0%, and to 100% in every cycle.

Figure 30.79 shows the PWM output cycle terminated by writing 0000 00_H to the counter and the counter restarted.

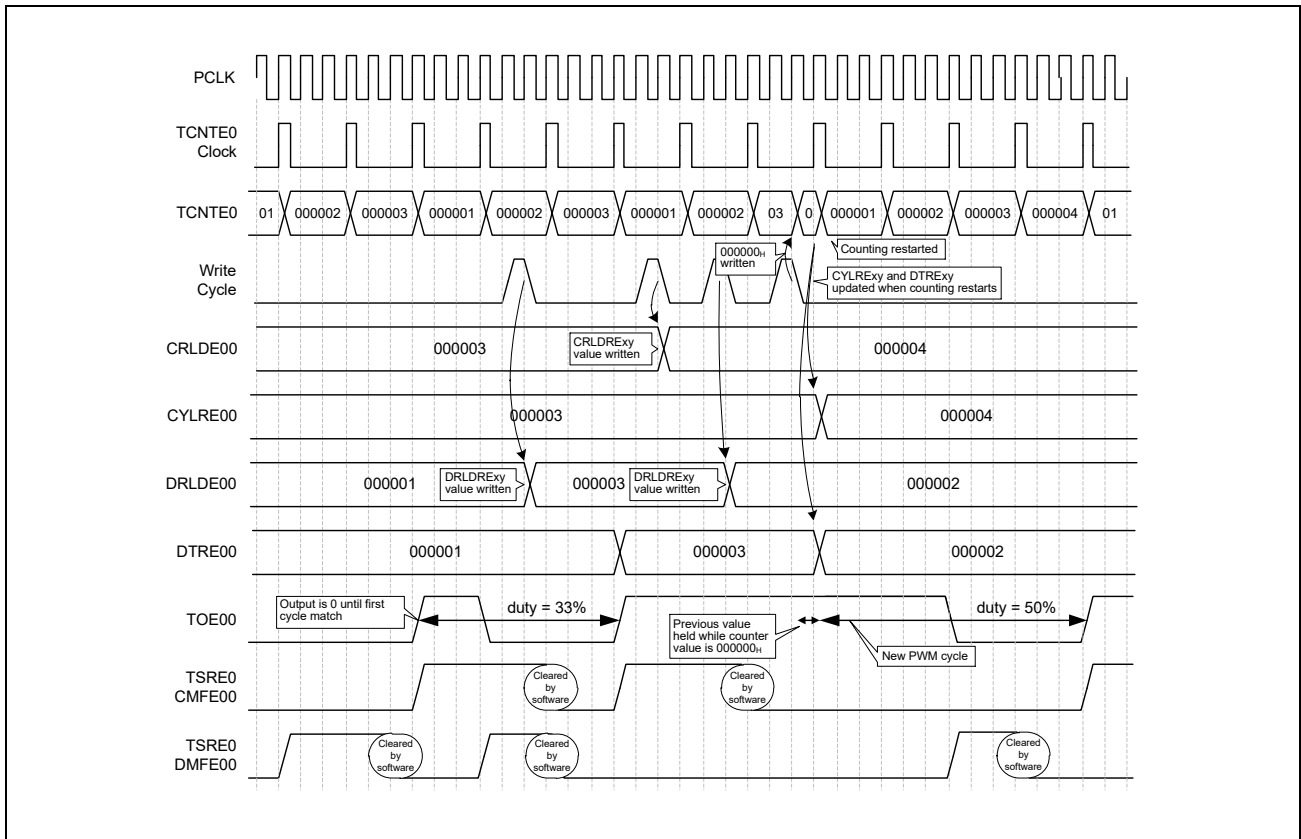


Figure 30.79 Operation of PWM (2)

Writing 0000 00_H to the 24 higher-order bits of the counter changes the 24 higher-order bits values of the counter to 0000 00_H. The output waveform (TOE00) does not change. If the reload function is enabled, an input of counter clock after writing 0000 00_H transfers the values of the cycle reload register and the duty reload register to the cycle-setting register and the duty cycle setting register, respectively. At the same time, counting starts and PWM output also starts.

A waveform is output in off-state duty (active-low output) mode when the off-state duty mode is selected by setting the corresponding bit in the timer output control register Ex (TOCREx). The output waveform on pin TOExy is inverted on the next PCLK cycle after setting.

Figure 30.80 shows an example of a waveform when switching on- and off-state duty modes. By selecting the off-state duty mode before the counter is started, the initial output level on the PWM output pin TOE00 is 1. After the counter started until the first cycle match, the level on pin TOE00 retains 1. On the following cycle match and duty match, the output levels are alternated. When the PWM cycle is forcibly terminated by writing 0000 00_H to the 24 higher-order bits of the counter, TOE00 retains the previous value. At the timing in which the 24 higher-order bits of the counter is incremented to 0000 01_H, a new PWM cycle is started.

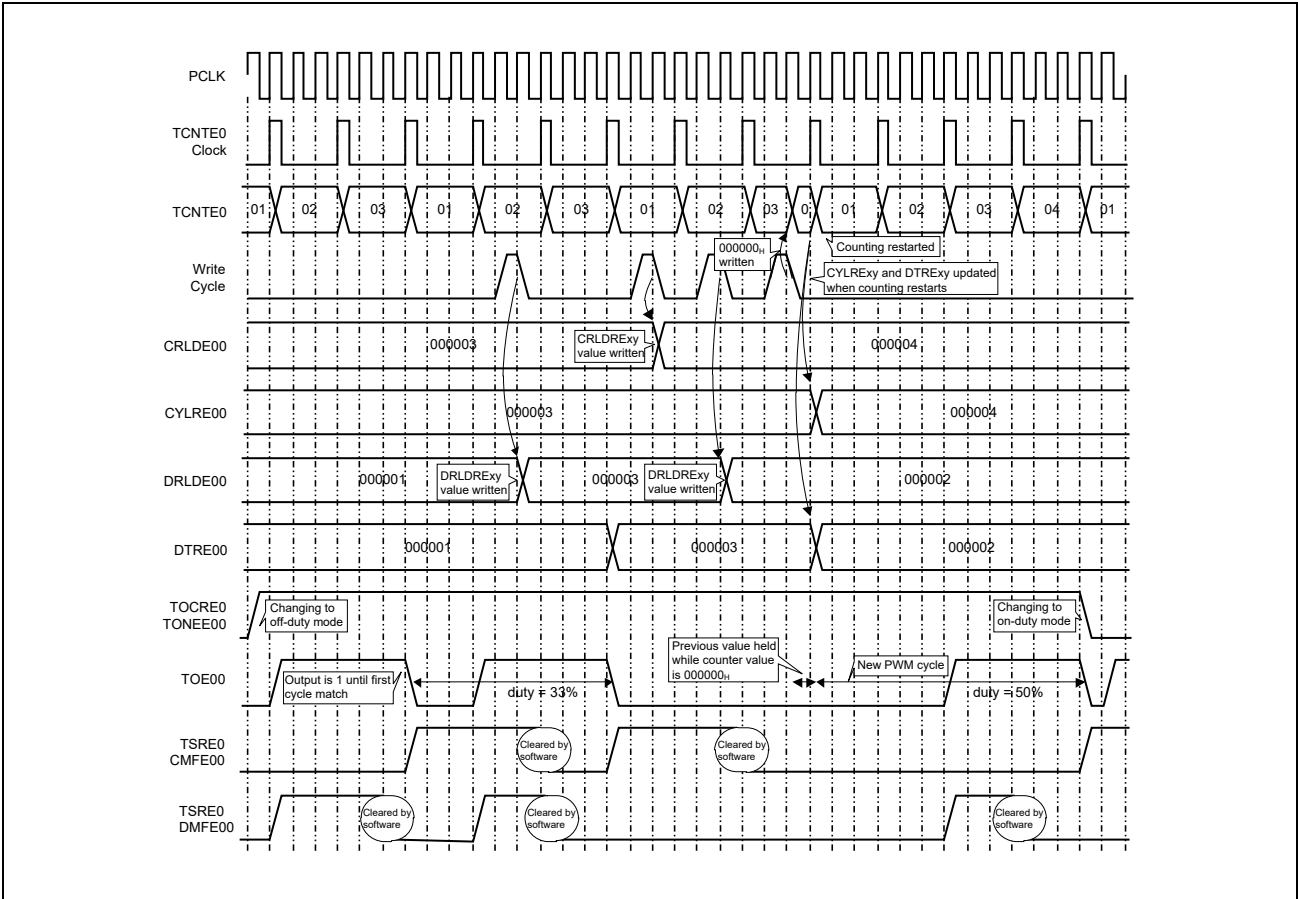


Figure 30.80 Operation of PWM (3)

Figure 30.81 illustrates the output waveform when PWM output is shutoff. PWM output is shutoff if the output shutoff enable/disable selection bit (POEEN_{xy}) of the output shutoff control register Ex (POECRE_{Ex}) is enabled and shutoff input (POE_{Ex}) is enabled during a duty period (between a cycle match in the previous PWM cycle to a duty match). POE_{Ex} being enabled during non-duty period does not trigger a shutoff. During an output shutoff period, the signal is output to TOE_{xy} at the level specified by the output level H/L selection bit (PWMSLV_{xy}) of the output shutoff setting register Ex (SOLVLE_{Ex}). This output level is not affected by the inversion setting (on-state/off-state duty) of the corresponding bit of the timer output control register (TOCRE_{Ex}).

The state of output shutoff is released if POE_{Ex} is disabled at the time of a cycle match. The state of PWM output being shutoff continues to the next cycle match if POE_{Ex} is in the enabled state at the above-mentioned timing.

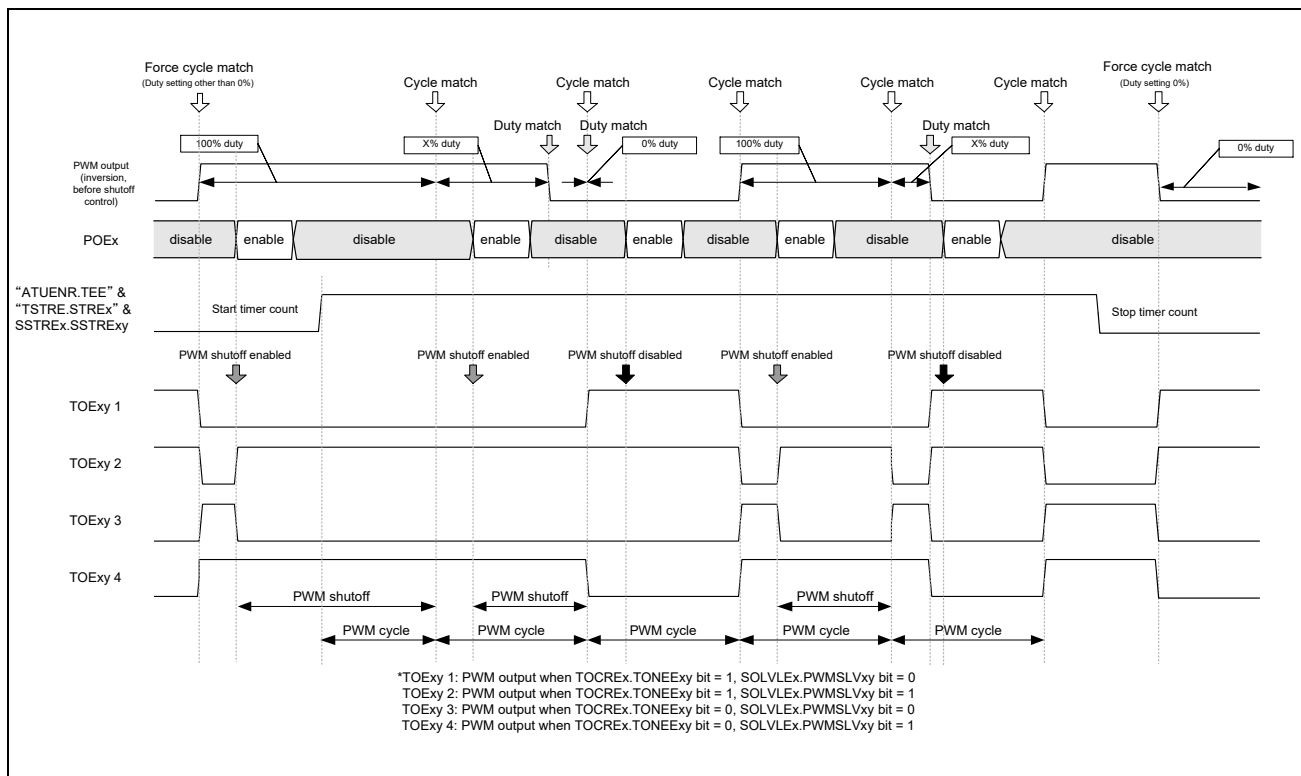


Figure 30.81 Example of PWM Output Shutoff

- Supplementary note:

The shutdown state continues until the next cycle match, even if POE_{Ex} is temporarily enabled or disabled within the duty period. When the setting is for a duty cycle of 100% (except when it becomes 0% due to reloading), the shutdown state can newly be set by enabling POE_{Ex}, even at the time of a cycle match. Shutdown or release from shutdown is in accord with POE_{Ex} for forced ending of the PWM cycle when the counter is 0000 00_H, if the setting is not for a duty cycle of 0% (including it becoming 0% due to reloading). After POE_{Ex} has been disabled, only a request for release from shutdown is accepted if the setting is for a duty cycle of 0% (including it becoming 0% due to reloading).

30.10 Timer F

30.10.1 Overview of Operation

The timer F block consists of SG_F timer F subblocks.

Timer F subblocks realize the following functions:

- Edge counting in a specified period:
Counts the number of edges input to the external input pin (TIFxA)
- Effective edge interval counting:
Measures time until a specified number of edges is input to the external pin (TIFxA).
- Measurement of time during high/low input levels:
Measures a total amount of time when a high or low level is input to the external input pin (TIFxA). The duration of measurement is designated as the number of pulses input to the external pin.
- Measurement of PWM input waveform timing:
Measures the off-duty period and cycle time of the PWM waveform input to the external pin (TIFxA). The duration of measurement is designated as the number of PWM cycles input to the external pin.
- Rotation speed/pulse measurement (for the subblock 3 to SB_F-1 only):
Every time an edge is input to the external pin (TIFxA), the following values are retained edge count, time stamp at edge input, edge input interval (cycle), and high/low input level immediately before input.
- Up/down event count (for subblocks 0 to 2 only):
TIFxA of the two external pins (TIFxA, TIFxB) is used to count as the count source. TIFxB switches between up-counting and down-counting.
- Four-time multiplication event count (for subblocks 0 to 2 only):
Counting operation is executed using two external input pins (TIFxA, TIFxB) as the count sources. Signals in the pins switch between up-counting and down-counting.

Input signals from the external input pins TIFxA and TIFxB can be subject to the noise cancellation function using the input cancellation function. Noise-canceled signals from TIA00 to TIA0SB_A-1 of timer A can be used instead of external signal input pin TIFxA.

Inputs of TIF0A to TIF2A and TIF0B and TIF2B can also be used as inputs (shutoff inputs) for POE0 to POE5 of timer E. For the correspondence between TIFxA/B and POE_x, see **Table 30.154** of timer E.

Note that, if any of TIF0A to TIF2A and TIF0B to TIF2B is used as a shutoff input of timer E, the shutoff input becomes the measurement target of the corresponding timer F subblock.

Configuration

Figure 30.82 is a block diagram of timer F. Components of a timer F subblock include two external outputs (TIFxA and TIFxB), two 32-bit counters (ECNTAFx and ECNTCFx), three 32-bit general registers (GRAFx, GRCFx, and GRDFx), three 32-bit backup registers (BGRAFx*¹, BGRCFx, and BGRDFx*²), 16-bit up/down counters (ENCTBFx), 16-bit general registers (GRBFx), an input processor (edge detection and noise canceler), and a controller.

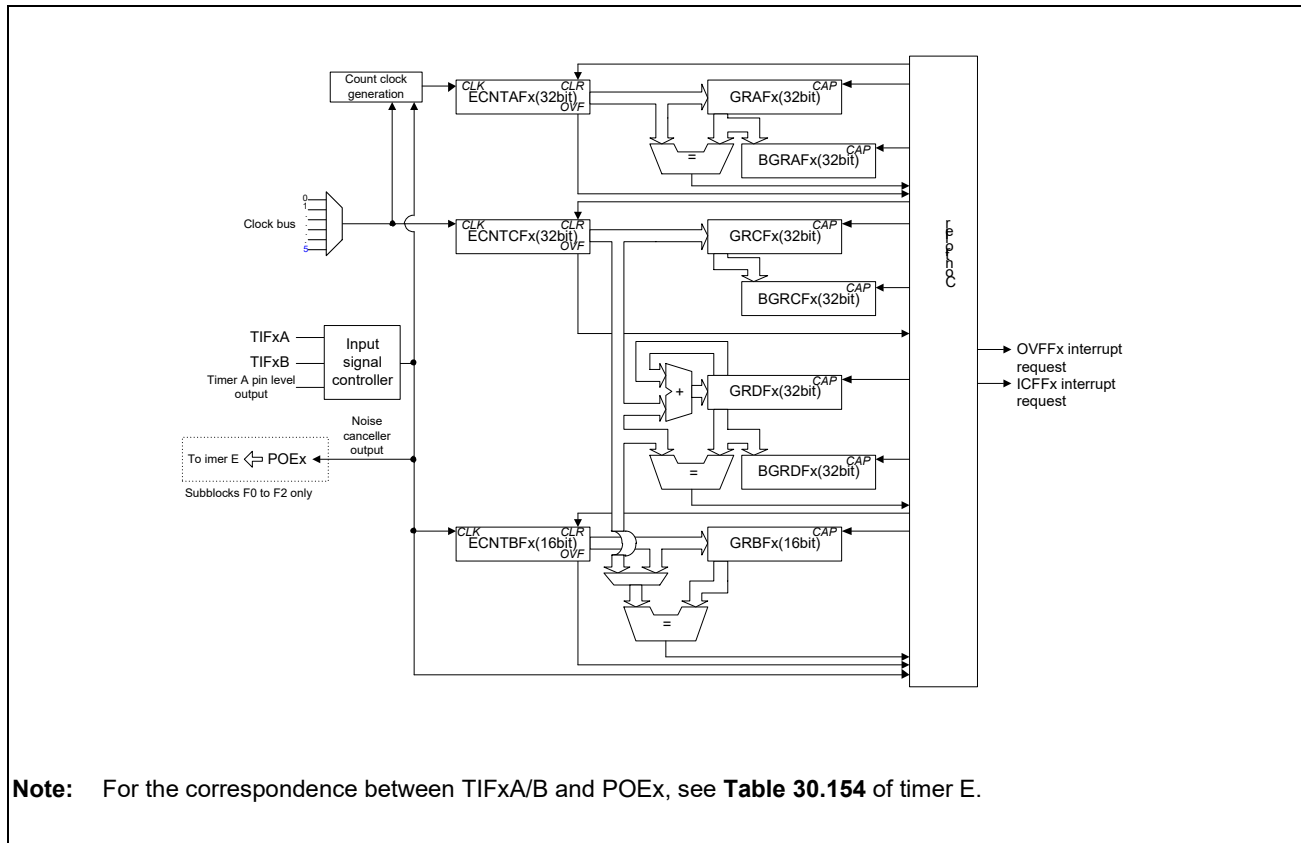


Figure 30.82 Block Diagram of Timer F Subblock

Note 1. The BGRAFx register is provided for subblocks 3 to SB_F-1 only.

Note 2. The BGRDFx register is provided for subblocks 3 to SB_F-1 only.

Interrupt request

The timer F can output two types of interrupts totaling $SB_F \times 2$ interrupts.

- OVF_{Fx} interrupt requests (SB_F interrupts ($0 \leq x \leq SB_F-1$)): An interrupt is output when one of the three counters (ECNTAF_x, ECNTBF_x, ECNTCF_x) in the subblock F_x has overflowed or underflowed (only in ECNTBF_x). They can also be used as compare match interrupts in PWM input waveform measurement mode and “rotation speed/pulse measurement” mode. To which counter the interrupt belongs can be known by referring to the timer status register F (ISRF). This request is received by the INTC block and the designated processing is performed.
- ICF_{Fx} interrupt requests (SB_F interrupts ($0 \leq x \leq SB_F-1$)): The interrupt is output when a count value capturing in the timer subblock F_x occurs. This request is received by INTC and performs designated processing by interrupt request.

30.10.2 Registers Related to Timer F

30.10.2.1 TSTRF — Timer Start Register F

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STRF 15	STRF 14	STRF 13	STRF 12	STRF 11	STRF 10	STRF 9	STRF 8	STRF 7	STRF 6	STRF 5	STRF 4	STRF 3	STRF 2	STRF 1	STRF 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.169 TSTRF Register Contents

Bit Position	Bit Name	Function
31 to SB _F	—	These bits are not used. When read, "0" is always returned. When writing, always write 0.
SB _F -1 to 0	STRF _x	Timer F start 0: Stops counting of ECNTAF _x , ECNTBF _x , and ECNTCF _x . 1: Permits counting of ECNTAF _x , ECNTBF _x , and ECNTCF _x .

Note: x indicates the subblock number (channel number) of 0 to SB_F-1.

TSTRF is an 8-bit/16-bit/32-bit readable/writable register that specifies whether to operate or stop each subblock (timer F0 to FSB_F-1) in the timer F.

Count operation is not executed unless TFE bit in ATU master enable register (ATUENR) is enabled even if the start bit in timer F is set to enable the count operation.

TSTRF is initialized to 0000 0000_H after reset.

Counter F Start (STRF_x)

These bits specify whether to operate or stop two time counters (ECNTAF_x, ECNTCF_x) in subblocks (F0 to FSB_F-1) of timer F and event counter (ECNTBF_x).

When this bit is cleared to 0, ECNTAF_x, ECNTBF_x, and ECNTCF_x stop operation. Counter value is retained at stop state. When this bit is set to 1 once again, the operation starts at the retained value.

Count operation is not executed unless TFE bit in ATU master enable register (ATUENR) is enabled even if the start bit in timer F is set enable the count operation.

CAUTION

The prescalers run independently of the setting of the STRF_x bit and are not initialized at the start of counter. Therefore, during the time between the activation and the start of actual count operation by the above counter, hardware-related uncertainty shorter than the period of selected count source (resolution) accompanies.

30.10.2.2 NCMCR1F — Noise Cancellation Mode Channel Register 1F

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NCM1F 15	NCM1F 14	NCM1F 13	NCM1F 12	NCM1F 11	NCM1F 10	NCM1F 9	NCM1F 8	NCM1F 7	NCM1F 6	NCM1F 5	NCM1F 4	NCM1F 3	NCM1F 2	NCM1F 1	NCM1F 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.170 NCMCR1F Register Contents

Bit Position	Bit Name	Function
31 to SB _F	—	These bits are not used. When read, "0" is always returned. When writing, always write 0.
SB _F -1 to 0	NCM1F _x	Channel x Noise Cancel Mode Specify the operating mode of noise canceler in channel x. 0: Premature-transition cancellation mode 1: Minimum time-at-level cancellation mode (when NCMF = 0 and NCM2F _x = 0) 1: Level accumulation cancellation mode (when NCMF = 0 and NCM2F _x = 1)

Note: x indicates the subblock number (channel number) of 0 to SB_F-1.

NCMCR1F is an 8-bit/16-bit/32-bit readable/writable register that selects the operation mode of noise canceller for each channel unit.

In premature-transition cancellation mode, after an input signal level change is detected, any input signal level change in the specified period is ignored. This mode regards a signal level change in the specified period as noise after the first level change is detected.

In minimum time-at-level cancellation mode, after an input signal level change is detected, the first and subsequent level changes are ignored unless the input signal level remains the same over the specified period. This mode regards a signal level change that occurs within a shorter period as noise.

In level accumulation cancellation mode, input signal levels are accumulated and the input level is regarded to have reached 0 or 1 when the accumulation result becomes 0 or the specified value.

Each period is specified by the noise canceler register of each channel and the noise canceler counter measures time.

Noise Cancellation Mode Bit of Channel x (NCM1Fx)

Specify the operation mode of noise canceller in channel x.

CAUTION

This register is only effective when the NCMF bit of the noise cancellation mode register (NCMR) in the common control unit is "0". Furthermore, when the corresponding channel bit (NCM1Fx) in this register is set to "1", the mode can be set to premature-transition cancellation mode, minimum time-at-level cancellation mode, or level accumulation cancellation mode according to the state of the corresponding channel bit in the noise cancellation mode channel register 2F (NCMCR2F).

Note: x indicates 0 to SB_F-1.

30.10.2.3 NCMCR2F — Noise Cancellation Mode Channel Register 2F

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NCM2F 15	NCM2F 14	NCM2F 13	NCM2F 12	NCM2F 11	NCM2F 10	NCM2F 9	NCM2F 8	NCM2F 7	NCM2F 6	NCM2F 5	NCM2F 4	NCM2F 3	NCM2F 2	NCM2F 1	NCM2F 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.171 NCMCR2F Register Contents

Bit Position	Bit Name	Function
31 to SB _F	—	These bits are not used. When read, "0" is always returned. When writing, always write 0.
SB _F -1 to 0	NCM2F _x	Specify the operating mode of noise canceler in channel x. 0: Minimum time-at-level cancellation mode (when NCMF = 0 and NCM1F _x = 1) 1: Level accumulation cancellation mode (when NCMF = 0 and NCM1F _x = 1)

Note: x indicates the subblock number (channel number) of 0 to SB_F-1.

Noise cancel mode channel register 2F is an 8-bit/16-bit/32-bit readable/writable register that selects the operation mode of the noise canceler for each channel.

In minimum time-at-level cancellation mode, after an input signal level change is detected, the first and subsequent level changes are ignored unless the input signal level remains the same over the specified period. This mode regards a signal level change that occurs within a shorter period as noise.

In level accumulation cancellation mode, input signal levels are accumulated and the input level is regarded to have reached 0 or 1 when the accumulation result becomes 0 or the specified value.

Each period is specified by the noise canceler register of each channel and the noise canceler counter measures time.

Noise Cancellation Mode Bit 2 of Channel x (NCM2F_x)

Specify the operation mode of noise canceller in channel x.

CAUTION

This register is only effective when the NCMF bit of the noise cancellation mode register (NCMR) in the common control unit is "0". The channel mode when the corresponding bit (NCM2F_x) of this register is set to 1 can be set to the minimum time-at-level cancellation mode or the level accumulation cancellation mode depending on the setting of the corresponding bit of the noise cancel mode channel register 1F (NCMCR1F).

Note: x indicates 0 to SB_F-1.

30.10.2.4 NCCRF — Noise Canceller Control Register F

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NCEF	NCEF	NCEF	NCEF	NCEF	NCEF	NCEF	NCEF	NCEF	NCEF	NCEF	NCEF	NCEF	NCEF	NCEF	NCEF
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.172 NCCRF Register Contents

Bit Position	Bit Name	Function
31 to SB _F	—	These bits are not used. When read, “0” is always returned. When writing, always write 0.
SB _F -1 to 0	NCEFx	Noise canceler enable Fx 0: The noise cancellation is disabled for TIFxA and TIFxB. 1: The noise cancellation is enabled for TIFxA and TIFxB.

Note: x indicates the subblock number (channel number) of 0 to SB_F-1.

The noise canceler control register F (NCCRF) is an 8-bit/16-bit/32-bit readable/writable register that specifies enable/disable of the noise cancellation function of each subblock (timer F0 to timer FSB_F-1) of timer F.

NCCRF is initialized to 0000 0000_H after reset.

Noise Canceller Enable Fx (NCEFx)

Specify to enable/disable the noise canceller in each subblock. Regarding the subblocks 2 to 0, each subblock has noise cancellers TIFxA and TIFxB but enabling/disabling these cancellers cannot be specified independently. Setting the NCEFx bit to 1 enables each noise canceller in TIFxA and TIFxB.

When a level change of the external input signal TIFxA or TIFxB is detected while this bit is set to “1”, it is processed in premature-transition cancellation mode, minimum time-at-level cancellation mode, or level accumulation cancellation mode depending on the setting in the noise cancel mode register (NCMR) of common controller, noise cancel mode channel register 1F (NCMCR1F) of timer F, and noise cancel mode channel register 2F (NCMCR2F) of timer F.

In premature-transition cancellation mode, when a level change of the input signal is detected, the change is output as the signal whose noise is removed and the corresponding noise canceler counter (NCNTFA0 to NCNTFASB_F-1, NCNTFB0 to NCNTFB2) starts counting up. Subsequent level changes are masked until the value in the counter reaches the value in the noise canceler register (NCRFA0 to NCRFASB_F-1 and NCRFB2 to NCRFB0). When this compare match occurs, the level of the input signal at that time is output as the signal after noise cancellation.

When these bits are cleared to “0” while NCNTFAx and NCNTFBx are being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes of TIFAx and TIFBx are also masked over this period.

In minimum time-at-level cancellation mode, when a level change of the external input signal is detected, the corresponding noise canceler counter (NCNTFA0 to NCNTFASB_{F-1}, NCNTFB0 to NCNTFB2) starts counting up. If subsequent level changes are not detected until the value in the counter reaches the value in the noise canceler register (NCRFA0 to NCRFASB_{F-1} and NCRFB2 to NCRFB0), the previously accepted level change is output as the signal after noise cancellation upon compare match of the counter and noise canceler register. When the subsequent level change is detected before the values in the counter and noise canceler register match, all the changes are treated as noise. Therefore, the noise canceler assumes that the levels of external input signals have not changed and does not change the signal after cancellation.

When these bits are cleared to “0” while NCNTFA_x and NCNTFB_x are being incremented, counting continues until the values in the counter and the noise canceler register match or a level change on the input signal is detected.

In level accumulation cancellation mode, the corresponding noise canceler counter (NCNTFA0 to NCNTFASB_{F-1}, NCNTFB0 to NCNTFB2) increments or decrements according to the input signal level. The noise canceler counter increments (up-counting) when the input signal level is high, and decrements (down-counting) when the input signal level is low. Up-counting continues until the noise canceler counter reaches the value set in the noise canceler register (NCRFA0 to NCRFASB_{F-1} and NCRFB2 to NCRFB0). Down-counting continues until NCNTA_x reaches 00_H.

When a compare match (with NCRFA0 to NCRFASB_{F-1} and NCRFB2 to NCRFB0) occurs during up-counting, the noise canceler output is updated to “1”. When a compare match (with 0000_H) occurs during down-counting, the noise canceler output is updated to “0”.

In minimum time-at-level cancellation mode and premature-transition cancellation mode, level changes are always detected by PCLK regardless of the selected noise-canceler clock. In level accumulation cancellation mode, input level sampling is performed by the noise canceler clock (PCLK or PCLK/128).

For details of operations for noise cancellation, see **Figure 30.1**, **Figure 30.2**, and **Figure 30.3**.

30.10.2.5 PVFCRF — Private Function Control Register F

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PVFCRKEY[7:0]								—	—	—	—	—	—	GRDFCMEN	BKCRWEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R	R	R	R	R	R/W	R/W

Note 1. Values written to the write key code are not retained.

Table 30.173 PVFCRF Register Contents

Bit Position	Bit Name	Function
15 to 8	PVFCRKEY[7:0]	Write key code C9 _H : GRDFCMEN and BKCRWEN values can be changed. Other than C9 _H : GRDFCMEN and BKCRWEN values cannot be changed.
7 to 2	—	These bits are not used. When read, "0" is always returned. When writing, always write 0.
1	GRDFCMEN	GRDFx compare match enable 0: GRDFx compare match function is disabled in PWM input waveform measurement mode. 1: GRDFx compare match function is enabled in PWM input waveform measurement mode.
0	BKCRWEN	Backup control register Fx write enable 0: The BKCRFx register is not writable. 1: The BKCRFx register is writable.

Note: x indicates the subblock number (channel number) of 0 to SB_F-1.

The private function control register F (PVFCRF) is a 16-bit readable/writable register. To write a value to this register, write key code setting is required to bits 15 to 8 as well as the data to be written.

PVFCRF is initialized to 0000_H after reset.

Write Key Code (PVFCRKEY)

This code controls the write permission for bits GRDFCMEN and BKCRWEN. Setting is required to bits 15 to 8 as well as the data to be written to bits GRDFCMEN and BKCRWEN. Data written to this bit is not retained; Value 00_H is always returned when being read.

GRDFx Compare Match Enable (GRDFCMEN)

Controls enable/disable of the compare match between time measurement counter CFx (ECNTCFx) and general register DFx (GRDFx). This bit takes effect in PWM input waveform mode, and is ignored in other modes. (GRDFx compare match function is disabled.)

The setting of this bit is applied to all subblocks of timer F. Subblock-individual setting is not possible.

Backup Control Register Fx Write Enable (BKCRWEN)

Controls write permission to the backup control register Fx (BKCRFx).

The setting of this bit is applied to all subblocks of timer F. Subblock-individual setting is not possible.

30.10.2.6 TCR1Fx — Timer Control Registers 1Fx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CKSELFx[2:0]			MDFx[2:0]			EGSELFx[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_F-1: Corresponding to subblocks F0 to FSB_F-1

Table 30.174 TCR1Fx Register Contents

Bit Position	Bit Name	Function
7 to 5	CKSELFx[2:0]	Clock select Fx Specify the clock sources for the time counters (ECNTAFx, ECNTCFx). 000: Clock bus 0 001: Clock bus 1 010: Clock bus 2 011: Clock bus 3 100: Clock bus 4 101: Clock bus 5 110: Setting prohibited 111: Setting prohibited
4 to 2	MDFx[2:0]	Timer operation mode Fx Specify the operation mode for the corresponding subblocks Fx. 000: Edge counting in a specified period 001: Effective edge interval counting 010: Measurement of time during high/low input levels 011: Setting prohibited 100: Measurement of PWM input waveform timing 101: Rotation speed/pulse measurement 110: Up/down event count 111: Four-time multiplication event count
1, 0	EGSELFx[1:0]	Edge select Fx Specify the edge sense mode of TIFxA input or TIA00 to TIA0SB _A -1 input. 00: Edge detection disabled 01: Rising edge 10: Falling edge 11: Both edges

Note: x indicates an integer from 0 to SB_F-1.

Timer control register Fx (TCR1Fx) is an 8-bit readable/writable register that specifies the operation mode of the subblocks F0 to FSB_F-1.

TCR1Fx is initialized to 00_H after reset.

Clock Select Fx (CKSELFx)

Specify the clock sources for the two time counters (ECNTAF_x, ECNTCF_x) in the subblocks F0 to FSB_{F-1}.

Setting these bits to a value from “000” to “101” selects the corresponding clock bus (0 to 5) as the clock source. Do not specify “110” and “111”. If specified, the operation is not guaranteed.

Timer Operation Mode Fx (MDFx)

Specify the operation mode for the timer subblocks F0 to FSB_{F-1}. There are seven modes: up/down event count, four-time multiplication event count, edge counting in a specified period, effective edge interval counting, measurement of time during high/low input levels, measurement of PWM input waveform timing, and rotation speed/pulse measurement.

Do not set rotation speed/pulse measurement for subblocks other than subblocks 3 to SB_{F-1}.

Edge Select Fx (EGSELFx)

Specifies the event input (TIF_xA) of timer subblocks F0 to FSB_{F-1} and the edge sense mode of pin inputs (TIA00 to TIA0SB_{A-1}) from timer A.

Edge detection is done for signals that have passed through the noise canceller. Therefore, edge detection is done to the external input (TIF_xA, TIF_xB) if the noise cancellation function is disabled, and to signals after noise cancel if the noise cancellation function is enabled.

While measurement of time during high/low input levels is specified, when this bit selects the falling edge, measurement of time during high level is specified. When this bit selects the rising edge, measurement of time during low level is specified. Do not select both edges.

While measurement of PWM input waveform timing and rotation speed/pulse measurement are specified, when this bit selects the rising edge, the period between the two rising edges is regarded as the PWM cycle and the low level period is regarded as the off-duty period. If the falling edge is selected, the period between the two falling edges is regarded as the PWM cycle and the high-level period is regarded as the off-duty period. Do not select both edges.

When up/down event count mode and four-time multiplication event count mode are specified, be sure to designate both the rising and falling edges. If otherwise selected, the operation is not guaranteed.

CAUTION

The TIF_xB pin is only available in up/down event count and four-time multiplication event count mode. TIF_xB operates always detecting both the rising and falling edges. In the other modes, TIF_xB does not detect edges.

30.10.2.7 TCR2F_x — Timer Control Registers 2F_x

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	EISELEF _x	—	—	—	—	EISELF _x [2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R/W	R/W	R/W

Note: x = 0 to SB_F-1: Corresponding to subblocks F0 to FSB_F-1

Table 30.175 TCR2F_x Register Contents

Bit Position	Bit Name	Function
7	EISELEF _x	Event input selection enable F _x Selects the event input. 0: Selects TIF _x A. 1: Selects the signal processed by noise cancellation of TIA.
6 to 3	—	These bits are not used. Fix these bits to 0.
2 to 0	EISELF _x	Event input selection F _x Selects the TIA signal when the signal processed by noise cancellation of TIA is selected by EISELEF _x . 000: Selects the signal of TIA00 of timer A processed by noise canceling. 001: Selects the signal of TIA01 of timer A processed by noise canceling. 010: Selects the signal of TIA02 of timer A processed by noise canceling. 011: Selects the signal of TIA03 of timer A processed by noise canceling. 100: Selects the signal of TIA04 of timer A processed by noise canceling. 101: Selects the signal of TIA05 of timer A processed by noise canceling. 110: Selects the signal of TIA06 of timer A processed by noise canceling. 111: Selects the signal of TIA07 of timer A processed by noise canceling.

Note: x indicates an integer from 0 to SB_F-1.

The timer control register 2F_x (TCR2F_x) is an 8-bit readable/writable register that specifies the event input of each subblock (timer F0 to timer FSB_F-1) of timer F.

TCR2F0 to TCR2FSB_F-1 are initialized to 00_H after reset.

Event Input Selection Enable F_x (EISELEF_x)

Selects an event input for each subblock (timer F0 to timer FSB_F-1) included in timer F.

By setting this bit, noise-canceled signals from TIA00 to TIA0SB_A-1 of timer A can be used as the event input instead of external input pin TIF_xA.

Event Input Selection F_x (EISELF_x)

When EISELEF_x is set to 1, noise-canceled signal TIA00 to TIA0SB_A-1 of timer A can be selected as the event input for each subblock (timer F0 to timer FSB_F-1) included in timer F.

30.10.2.8 TIERFx — Timer Interrupt Enable Registers Fx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	OVECFx	OVEBFx	OVEAFx	ICIEFx
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note: x = 0 to SB_F-1: Corresponding to subblocks F0 to FSB_F-1

Table 30.176 TIERFx Register Contents

Bit Position	Bit Name	Function
7 to 4	—	These bits are not used. When read, “0” is always returned. When writing, always write 0.
3	OVECFx	Overflow interrupt enable CFx 0: Interrupt by OVFCFx disabled 1: Interrupt by OVFCFx enabled
2	OVEBFx	Overflow interrupt enable BFx 0: Interrupt by OVFBFx disabled 1: Interrupt by OVFBFx enabled
1	OVEAFx	Overflow interrupt enable AFx 0: Interrupt by OVFAFx disabled 1: Interrupt by OVFAFx enabled
0	ICIEFx	Input capture interrupt enable Fx 0: Interrupt by ICFFx disabled 1: Interrupt by ICFFx enabled

Note: x indicates an integer from 0 to SB_F-1.

Timer interrupt enable registers Fx (TIERF0 to TIERFSB_F-1) are 8-bit readable/writable registers that specify whether to enable or disable the interrupt corresponding to the timer status register Fx (TSRFx).

TIERF0 to TIERFSB_F-1 are initialized to 00_H after reset.

Overflow Interrupt Enable CFx (OVECFx)

Specifies whether to enable or disable interrupt requests of the state (OVFCFx) corresponding to overflow of timer measurement counter Cx (ECNTCFx) (when it is in PWM input waveform measurement mode and the GRDFCMEN bit of the PVFCRF register is 0), corresponding to compare match of ECNTCFx and GRDFx (when it is in PWM input waveform measurement mode and the GRDFCMEN bit of the PVFCRF register is 1), or corresponding to compare match of ECNTCFx and GRBFx (rotation speed/pulse measurement mode).*¹

Overflow Interrupt Enable BFx (OVEBFx)

Specifies whether to enable or disable the interrupt of the state (OVFBFx) corresponding to the overflow/underflow of the event counter x (ECNTBFx).*¹

Overflow Interrupt Enable AFx (OVEAFx)

Specifies whether to enable or disable the interrupt of the state (OVFAFx) corresponding to the overflow of the time measurement counter Ax (ECNTAFx).*¹

Input Capture Interrupt Enable Fx (ICIEFx)

Specifies whether to enable or disable the interrupt of the state (ICFFx) corresponding to the input capture or compare match detection status in subblock Fx (F0 to FSB_F-1).*1

Note 1. The overflow of interrupt of the timer subblock Fx is requested as the logical sum of the interrupts OVFAFx, OVFBFx, and OVFCFx, by referring to TSRFx, which counter generated the interrupt by overflow or underflow can be known. Clear the corresponding interrupt request flag of TSRFx when processing the interrupt.

30.10.2.9 BKCRFx — Backup Control Register Fx

Value after reset: 00_H

[Subblocks F00 to F02]

Bit	7	6	5	4	3	2	1	0
	—	—	BKENC _{Fx}	—	—	—	ARSWC _{Fx}	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W* ¹	R	R	R	R/W* ¹	R

Note: x = 0 to SB_F-1: Corresponding to subblocks F0 to FSB_F-1

[Subblocks F03 to FSB_F-1]

Bit	7	6	5	4	3	2	1	0
	—	BKENA _{Fx}	BKENC _{Fx}	BKEND _{Fx}	—	ARSWA _{Fx}	ARSWC _{Fx}	ARSWD _{Fx}
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W* ¹	R/W* ¹	R/W* ¹	R	R/W* ¹	R/W* ¹	R/W* ¹

Note: x = 0 to SB_F-1: Corresponding to subblocks F0 to FSB_F-1

Note 1. Writing to this register is permitted only when the BKCRWEN bit of the PVFCRF register is 1.

Table 30.177 BKCRFx Register Contents

Bit Position	Bit Name	Function
7	—	This bit is not used. When read, “0” is always returned. When writing, always write 0.
6	BKENA _{Fx} * ¹	Backup enable A Saving GRA _{Fx} in BGRA _{Fx} when CDR _{Fx} is read 0: Not saved 1: Saved
5	BKENC _{Fx}	Backup enable C Saving GR _{Cx} in BGR _{Cx} when CDR _{Fx} is read 0: Not saved 1: Saved
4	BKEND _{Fx} * ¹	Backup enable D Saving GR _{Dx} in BGR _{Dx} when CDR _{Fx} is read 0: Not saved 1: Saved
3	—	This bit is not used. When read, “0” is always returned. When writing, always write 0.
2	ARSWA _{Fx} * ¹	Mapping to access register switching A To FFE6 5054 _H + (40 _H × x) 0: GRA _{Fx} is mapped. 1: BGRA _{Fx} is mapped.
1	ARSWC _{Fx}	Mapping to access register switching C To FFE6 5064 _H + (40 _H × x) 0: GR _{Cx} is mapped. 1: BGR _{Cx} is mapped.
0	ARSWD _{Fx} * ¹	Mapping to access register switching D To FFE6 5068 _H + (40 _H × x) 0: GR _{Dx} is mapped. 1: BGR _{Dx} is mapped.

Note 1. BKENA_{Fx}, BKEND_{Fx}, ARSWA_{Fx}, and ARSWD_{Fx} exist only in subblocks F03 to FSB_F-1. They are not deployed in F00 to F02. When these bits are read, “0” is always returned. When writing, write “0”.

Note: x indicates an integer from 0 to SB_F-1.

The backup control register F_x (BKCRF00 to BKCRFSB_{F-1}) is an 8-bit readable/writable register. Writing to BKCRF00 to BKCRFSB_{F-1} is permitted only when the BKCRWEN bit of the private function control register F (PVFCRF) is 1.

BKCRF00 to BKCRFSB_{F-1} are initialized to 00_H after reset.

Backup Enable Bits A, C, and D (BKENAF_x, BKENC_{Fx}, BKEND_{Fx})

These bits permit or prohibit value saving of GRAF_x to BGRAF_x, GRCF_x to BGRCF_x, and GRDF_x to BGRDF_x when capture output register F_x (CDRF_x) is read.

Access Register Switching Bits A, C, and D (ARSWAF_x, ARSWCF_x, ARSWDF_x)

These bits specify whether BGRAF_x is mapped instead of GRAF_x to address FFE6 5054_H + (40_H × x), whether BGRCF_x is mapped instead of GRCF_x to address FFE6 5064_H + (40_H × x), and whether BGRDF_x is mapped instead of GRDF_x to address FFE6 5068_H + (40_H × x).

CAUTION

Setting of this register always takes effect regardless of the operation mode. Value saving and mapping change are performed even for registers that do not operate in some operation mode.

30.10.2.10 TSRFx — Timer Status Registers Fx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	OVFCFx	OVFBFx	OVFAFx	ICFFx
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Note: x = 0 to SB_F-1: Corresponding to subblocks F0 to FSB_F-1

Table 30.178 TSRFx Register Contents

Bit Position	Bit Name	Function
7 to 4	—	These bits are not used. When read, “0” is always returned. When writing, always write 0.
3	OVFCFx	Overflow flag CFx 0: No overflow in ECNTCFx 1: ECNTCFx overflows
2	OVFBFx	Overflow flag BFx 0: No overflow or underflow in ECNTBFx 1: ECNTBFx overflows or underflows
1	OVFAFx	Overflow flag AFx 0: No overflow in ECNTAFx 1: ECNTAFx overflows
0	ICFFx	Input capture flag Fx 0: Input capture is not detected in the subblock Fx 1: Input capture in subblock Fx detected

Note: x indicates an integer from 0 to SB_F-1.

Timer status registers Fx (TSRF0 to TSRF_{SB_F-1}) are 8-bit readable registers that indicate overflows in the time counters A and C, overflow or underflow in the event counter, and input capture occurrence.

This flag is a status flag that indicates occurrence of an interrupt request. For overflow flags AFx, BFx, and CFx (OVFAx, OVFBx, and OVFCx), interrupt requests can be issued by setting the TIERFx register. Setting a bit of the timer status clear register Fx (TSCRFx) clears the corresponding flag.

For input capture flag Fx (ICFFx), an interrupt request is made again if an interrupt source takes effect while this flag is set. An interrupt request is made even if there is a conflict between clearing by the corresponding timer status clear register and setting by an interrupt source.

For overflow flags, an interrupt request is issued when the corresponding flag is set by setting the corresponding bit of timer interrupt enable register Fx (TIERFx).

TSRF0 to TSRF_{SB_F-1} are initialized to 00_H after reset.

Overflow/Compare Match Flag CFx (OVFCFx)

Values of this flag indicate different conditions depending on the operation mode and the setting of the GRDFx compare match enable bit (GRDFCMEN) of the private function control register (PVFCRF). When the PWM input waveform measurement mode is set, value 0 of the GRDFCMEN bit makes this flag indicate an overflow of time measurement counter Cx (ECNTCFx), and value 1 of the GRDFCMEN bit makes this flag indicate a compare match of ECNTCFx and GRDFx. In “rotation speed/pulse measurement” mode, this flag indicates a compare match of ECNTCFx and GRBFx regardless of the GRDFCMEN bit setting.

This flag cannot be set to 1 or 0 by software.

- Setting (to 1) condition
(Measurement of PWM input waveform timing mode)
[GRDFCMEN = 0]
When the value of ECNTCFx has overflown (FFFF FFFF_H to 0000 0000_H)
[GRDFCMEN = 1]
When the values of ECNTCFx and GRDFx coincide.
(Rotation speed/pulse measurement mode)
When ECNTCFx and GRDFx (the value of zero extension to the 16 lower-order bits) coincide
- Clearing (to 0) condition
When 1 is written to OVFCF_x of the timer status clear register F_x (TSCRF_x).

Overflow Flag BF_x (OVFBF_x)

By this flag, overflow or underflow of the event counter F_x (ECNTBF_x) can be monitored. This flag cannot be set to 1 or 0 by software.

- Setting (to 1) condition
When ECNTBF_x overflows (FFFF_H to 0000_H) or underflows (0000_H to FFFF_H)
- Clearing (to 0) condition
When 1 is written to OVFCBF_x of the timer status clear register F_x (TSCRF_x).

Overflow Flag AF_x (OVFAF_x)

By this flag, overflow of the time counter AF_x (ECNTAF_x) can be monitored. This flag cannot be set to 1 or 0 by software.

- Setting (to 1) condition
When ECNTAF_x overflows (FFFF FFFF_H to 0000 0000_H)
- Clearing (to 0) condition
When 1 is written to OVFCAF_x of the timer status clear register F_x (TSCRF_x).

Input Capture/Compare Match Flag F_x (ICFF_x)

By this bit, the detection state of input capture and compare match in the subblock F_x (F₀ to FSB_F-1) can be monitored. This flag cannot be set to 1 or 0 by software.

- Setting (to 1) condition
When input capture is detected in subblock F_x
- Clearing (to 0) condition
When 1 is written to ICFCF_x of the timer status clear register F_x (TSCRF_x)

30.10.2.11 TSCRF_x — Timer Status Clear Register F_x

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	OVFCCF _x	OVFCBF _x	OVFCFAF _x	ICFCF _x
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Note: x = 0 to SB_F-1: Corresponding to subblocks F0 to FSB_F-1

Table 30.179 TSCRF_x Register Contents

Bit Position	Bit Name	Function
7 to 4	—	These bits are not used. When read, “0” is always returned. When writing, always write 0.
3	OVFCCF _x	Overflow flag clear FC enable 0: Disabled (Default) 1: Clears OVFCF _x of timer status register F (TSRF _x) to 0.
2	OVFCBF _x	Overflow flag clear FB enable 0: Disabled (Default) 1: Clears OVFBF _x of timer status register F (TSRF _x) to 0.
1	OVFCFAF _x	Overflow flag clear FA enable 0: Disabled (Default) 1: Clears OVFAF _x of timer status register F (TSRF _x) to 0.
0	ICFCF _x	Overflow flag clear F enable 0: Disabled (Default) 1: Clears ICFF _x of timer status register F (TSRF _x) to 0.

Note: x indicates an integer from 0 to SB_F-1.

The timer status clear register F_x (TSCRF00 to TSCRF_{SB_F-1}) is an 8-bit writable register. This register specifies flag clearing in the events of an overflow of time measurement counter A or C, an overflow or an underflow of an event counter, and an input capture occurrence.

TSCRF_x is readable/writable only in 8-bit units. When this register is read, “0” is always returned.

TSCRF_x is initialized to 00_H after reset.

Overflow Flag Clear FC Enable (OVFCCF_x)

When overflow flag FC_x (OVFCF_x) of timer status register F_x (TSRF_x) is set to 1, writing 1 to this register clears OVFCF_x to 0. This register is always read as 0.

Overflow Flag Clear FB Enable (OVFCBF_x)

When overflow flag FB_x (OVFBF_x) of timer status register F_x (TSRF_x) is set to 1, writing 1 to this register clears OVFBF_x to 0. This register is always read as 0.

Overflow Flag Clear FA Enable (OVFCFAF_x)

When overflow flag FA_x (OVFAF_x) of timer status register F_x (TSRF_x) is set to 1, writing 1 to this register clears OVFAF_x to 0. This register is always read as 0.

Input Capture Flag Clear B0 Enable (ICFCFx)

When input capture flag Fx (ICFFx) of timer status register Fx (TSRFx) is set to 1, writing 1 to this register clears ICFFx to 0. This register is always read as 0.

30.10.2.12 ECNTAFx — Timer Measurement Counters AFx

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECNTAFx[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECNTAFx[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_F-1: Corresponding to subblocks F0 to FSB_F-1

Table 30.180 ECNTAFx Register Contents

Bit Position	Bit Name	Function
31 to 0	ECNTAFx	Time measurement count AFx Up-counter A

Note: x indicates an integer from 0 to SB_F-1.

Timer measurement counters AFx (ECNTAF0 to ECNTAFSB_F-1) are 32-bit readable/writable registers. These registers are readable/writable only in 32-bit units. Do not write or read 8-bit or 16-bit data for these registers.

This register provided one for each subblock, and executes up-count operation using the input clock. One clock bus from clock buses 0 to 5 can be selected as the input clock according to the setting of the corresponding control register. The input clocks for ECNTAFx and ECNTCFx are the same. Clock source cannot be set independently.

When clearing the counter is done at the counting-up timing, ECNTAFx is cleared to 0000 0001_H, and to 0000 0000_H in other cases.

ECNTAF0 to ECNTAFSB_F-1 are initialized to 0000 0000_H after reset.

30.10.2.13 ECNTBFx — Event Counters Fx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECNTBFx[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_F-1: Corresponding to subblocks F0 to FSB_F-1

Table 30.181 ECNTBFx Register Contents

Bit Position	Bit Name	Function
15 to 0	ECNTBFx	Event count Fx Up/down counter

Note: x indicates an integer from 0 to SB_F-1.

Event counters Fx (ECNTBF0 to ECNTBFSB_F-1) are 16-bit readable/writable registers. This register is provided one for each subblock, and executes up-count/down-count operation using the input clock. The input clock is given two external input pins (TIFxA, TIFxB). The external pin and edge used to count differs according to the setting of the corresponding control register (operation mode and edge select). The input clock in each mode is listed in Table 30.182. When clearing the counter is done at the count-up timing, ECNTBFx is cleared to 0001_H, and to 0000_H in other cases. ECNTBF0 to ECNTBFSB_F-1 are initialized to 0000_H after reset.

Table 30.182 Event Counter Input Clocks and Count Edges for Each Operation Mode of Timer

Operation Mode	Input Clock	Count Edge
Edge counting in a specified period	TIFxA	Selectable by EGSELFx
Effective edge interval counting	TIFxA	Selectable by EGSELFx
Measurement of time during high/ low input levels	TIFxA	Selectable by EGSELFx (other than both edges)
Measurement of PWM input waveform timing	TIFxA	Selectable by EGSELFx (other than both edges)
Rotation speed/pulse measurement	TIFxA	Selectable by EGSELFx (other than both edges)
Up/down event count	TIFxA (Count direction is specified by TIFxB level)	Both rising/falling edges
Four-time multiplication event count	TIFxA, TIFxB	Both rising/falling edges

30.10.2.14 ECNTCFx — Time Measurement Counters CFx

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECNTCFx[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECNTCFx[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_F-1: Corresponding to subblocks F0 to FSB_F-1

Table 30.183 ECNTCFx Register Contents

Bit Position	Bit Name	Function
31 to 0	ECNTCFx	Time counter CFx Up-counter C

Note: x indicates an integer from 0 to SB_F-1.

Timer counters CFx (ECNTCF00 to ECNTCF_{SB_F}-1) are 32-bit readable/writable registers. They are up-counters. This register is readable/writable only in 32-bit units. Do not write or read 8-bit or 16-bit data for this register.

This register is provided one for each subblock, and is enabled only in measurement of PWM input waveform timing and rotation speed/pulse measurement modes. This register does not execute count operation in other modes.

This register executes up-count operation using the input clock. One clock bus from clock buses 0 to 6 can be selected as the input clock according to the setting of the corresponding control register. The input clocks for ECNTAFx and ECNTCFx are the same. Clock source cannot be set independently.

The timing of external input and the count value clearing by a compare match with ECNTBFx are synchronized with the count clock of ECNTCFx. At this time, ECNTCFx is cleared to 0000 0001_H.

ECNTCFx are initialized to 0000 0000_H after reset.

30.10.2.15 GRAF_x — General Registers AF_x

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GRAF _x [31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GRAF _x [15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: $x = 0$ to SB_F-1 : Corresponding to subblocks F0 to FSB_F-1

Table 30.184 GRAF_x Register Contents

Bit Position	Bit Name	Function
31 to 0	GRAF _x	General registers AF _x Input capture value or output compare match value for the time counter A.

Note: x indicates an integer from 0 to SB_F-1 .

General registers AF_x (GRAF00 to GRAFSB_{F-1}) are 32-bit readable/writable registers. This register is readable/writable only in 32-bit units. Do not write or read 8-bit or 16-bit data for this register. This register is provided one for each subblock, and has two functions such as input capture register and output compare register for the time measurement counter A_x (ECNTAF_x).

Do not set GRAF_x to 0000 0000_H to function this register as the compare match register. Note that if 0000 0000_H is set, incorrect measurement may occur.

GRAF00 to GRAFSB_{F-1} are initialized to FFFF FFFF_H after reset.

GRAF00 to GRAFSB_{F-1} are mapped to target addresses and become readable and writable when the ARSWAF_x bit of backup control register F_x (BKCRF_x) is 0 (default: 0).

30.10.2.16 BGRAF_x — Backup General Registers AF_x

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BGRAF _x [31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BGRAF _x [15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 03 to SB_F-1: Corresponding to subblocks F03 to FSB_F-1

Table 30.185 BGRAF_x Register Contents

Bit Position	Bit Name	Function
31 to 0	BGRAF _x [31:0]	When the BKENAF _x bit of BKCRF _x is 1, this register retains the value of general register AF _x (GRAF _x) at the time of reading the capture output register (CDRF _x).

Note: x indicates an integer from 03 to SB_F-1.

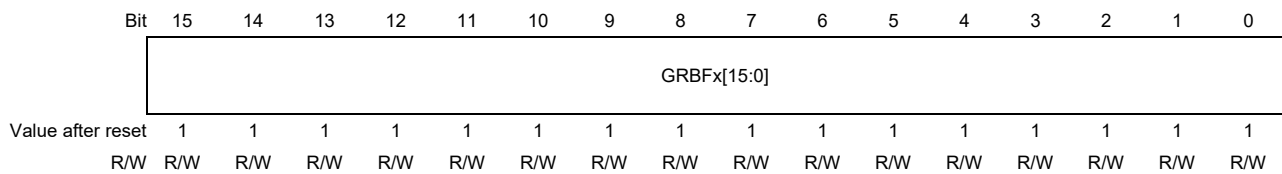
Backup general registers AF_x (BGRAF03 to BGRAFSB_F-1) are 32-bit read-only registers. BGRAF_x is provided one for each of subblock 03 to subblock SB_F-1. If the BKENAF_x bit of backup control register F_x (BKCRF_x) is 1, this register retains the content of general register AF_x (GRAF_x) when capture output register F_x (CDRF_x) is read.

BGRAF03 to BGRAFSB_F-1 are initialized to FFFF FFFF_H after reset.

BGRAF_x is mapped to target addresses and becomes readable when the ARSWAF_x bit of backup control register F_x (BKCRF_x) is 1 (default: 0).

30.10.2.17 GRBF_x — General Registers BF_x

Value after reset: FFFF_H



Note: x = 0 to SB_F-1: Corresponding to subblocks F00 to FSB_F-1

Table 30.186 GRBF_x Register Contents

Bit Position	Bit Name	Function
15 to 0	GRBF _x	General registers BF _x Input capture value or output compare match value for event counter.

Note: x indicates an integer from 0 to SB_F-1.

General registers BF_x (GRBF00 to GRBF_{SB_F}-1) are 16-bit readable/writable registers. GRBF_x is provided one for each subblock, and has two functions such as input capture register and output compare register for the event counter (ECNTBF_x).

Do not set 0000_H when using GRBF_x as the compare match register. If 0000_H is set, incorrect measurement may occur.

GRBF00 to GRBF_{SB_F}-1 are initialized to FFFF_H after reset.

30.10.2.18 GRCFx — General Registers CFx

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GRCFx[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GRCFx[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_F-1: Corresponding to subblocks F00 to FSB_F-1

Table 30.187 GRCFx Register Contents

Bit Position	Bit Name	Function
31 to 0	GRCFx	General registers CFx Input capture value for the time counter C

Note: x indicates an integer from 0 to SB_F-1.

General registers CF_x (GRCF00 to GRCF SB_F-1) are 32-bit readable/writable registers, and can be read from and written to in 32-bit units. Do not write or read 8-bit or 16-bit data for this register.

GRCF_x is provided one for each subblock, and has a function as the input capture register for the time measurement counter C_x (ECNTCF_x). Triggered by a compare match between ECNTBF_x and GRB_x (in measurement of PWM input waveform timing mode) or edge input of the TIF_xA pin (in rotation speed/pulse measurement mode), ECNTCF_x count number is taken in at the next ECNTCF_x up-count timing.

GRCF_x is valid only in measurement of PWM input waveform timing or rotation speed/pulse measurement mode. Capture operation is not executed in other modes.

GRCF00 to GRCF SB_F-1 are initialized to FFFF FFFF_H after reset.

GRCF00 to GRCF SB_F-1 are mapped to target addresses and become readable and writable when the ARSWCF_x bit of backup control register F_x (BKCRF_x) is 0 (default: 0).

30.10.2.19 BGRCF_x — Backup General Registers CF_x

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BGRCF _x [31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BGRCF _x [15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: $x = 0$ to SB_F-1 : Corresponding to subblocks F00 to FSB_{F-1}

Table 30.188 BGRCF_x Register Contents

Bit Position	Bit Name	Function
31 to 0	BGRCF _x [31:0]	When the BKENC _{Fx} bit of BKCRF _x is 1, this register retains the value of general register (GRCF _x) at the time of reading the capture output register CF _x (F).

Note: x indicates an integer from 0 to SB_F-1 .

Backup general registers CF_x (BGRCF00 to BGRCFSB_{F-1}) are 32-bit read-only registers.

BGRCF_x is provided one for each subblock. If the BKENC_{Fx} bit of backup control register F_x (BKCRF_x) is 1, this register retains the content of general register CF_x (GRCF_x) when capture output register F_x (CDRF_x) is read.

BGRCF00 to BGRCFSB_{F-1} are initialized to FFFF FFFF_H after reset.

BGRCF_x is mapped to target addresses and becomes readable when the ARSWCF_x bit of backup control register F_x (BKCRF_x) is 1 (default: 0).

30.10.2.20 GRDF_x — General Registers DF_x

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GRDF _x [31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GRDF _x [15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: $x = 0$ to SB_F-1 : Corresponding to subblocks F00 to FSB_{F-1}

Table 30.189 GRDF_x Register Contents

Bit Position	Bit Name	Function
31 to 0	GRDF _x	General registers DF _x In PWM input waveform measurement mode: Compare match value of time measurement counter C In “rotation speed/pulse measurement” mode: (only for $x = 3$ to SB_F-1) accumulated value of time measurement counter C

Note: x indicates an integer from 0 to SB_F-1 .

General registers DF_x (GRDF00 to GRDFSB_{F-1}) are 32-bit readable/writable registers that can be read and written only in 32-bit units. Do not read or write these registers in 16 bit or 8-bit units.

GRDF_x is provided one for each subblock. This register operates differently depending on the subblock. For subblocks F00 to F02, this register works only in PWM input waveform measurement mode. This register does not work in other modes.

For subblocks F03 to SB_F-1 , this register works in PWM input waveform measurement mode and in “rotation speed/pulse measurement” mode. This register does not work in other modes.

In PWM input waveform measurement mode, this register works as a compare match register, and is always compared with the value of time measurement counter C_x (ECNTCF_x). If the values of both registers coincide, the OVFCF_x bit of timer status register F_x (TSRF_x) is set to 1 at the next PCLK clock cycle. To make this register work as a compare match register, the GRDFCMEN bit of PVFCRF must be set.

In rotation speed/pulse measurement mode, this register works as an input capture register. Triggered by an edge input of the TIF_x pin, this register takes and accumulates the value of time measurement counter C_x (ECNTCF_x) at the next cycle of ECNTCF_x increment.

GRDF00 to GRDFSB_{F-1} are initialized to FFFF FFFF_H after reset.

GRDF03 to GRDFSB_{F-1} are mapped to target addresses and become readable and writable when the ARSWDF_x bit of backup control register F_x (BKCRF_x) is 0 (default: 0).

30.10.2.21 BGRDF_x — Backup General Registers DF_x

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BGRDF _x [31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BGRDF _x [15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.190 BGRDF_x Register Contents

Bit Position	Bit Name	Function
31 to 0	BGRDF _x [31:0]	When the BKENDF _x bit of BKCRF _x is 1, this register retains the value of general register DF _x (GRDF _x) at the time of reading the capture output register (CDRF _x).

Note: x indicates an integer from 3 to SB_F-1.

Backup general registers DF_x (BGRDF₃ to BGRDF_{SB_F-1}) are 32-bit read-only registers. BGRDF_x is readable only in 32-bit units. Do not read this register in 8-bit or 16-bit units.

BGRDF_x is provided one for each of subblocks F03 to subblock FSB_F-1. If the BKENDF_x bit of backup control register F_x (BKCRF_x) is 1, this register retains the content of general register DF_x (GRDF_x) when capture output register F_x (CDRF_x) is read.

BGRDF₃ to BGRDF_{SB_F-1} are initialized to FFFF FFFF_H after reset.

BGRDF_x is mapped to target addresses and becomes readable when the ARSWDF_x bit of backup control register F_x (BKCRF_x) is 1 (default: 0).

30.10.2.22 CDRFx — Capture Output Registers Fx

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDRFx[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDRFx[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: x = 0 to SB_F-1: Corresponding to subblocks F00 to FSB_F-1

Table 30.191 CDRFx Register Contents

Bit Position	Bit Name	Function
31 to 0	CDRFx	Capture output registers Fx Data retained in GRAFx, GRBFx, or ECNTBFx is read depending on the operation mode.

Note: x indicates an integer from 0 to SB_F-1.

The capture output register is a 32-bit read-only register. This register is provided one for each subblock. When this register is read, values in GRAFx, GRBFx, or ECNTBFx is read according to the operation mode. A 16-bit value in GRBFx and ECNTBFx are read from the 16 lower-order bits in CDRFx. In this case, the 16 higher-order bits in CDRFx are read as 0.

Registers corresponding to various modes are listed below. Writing to these registers are ignored.

Mode	Register
Edge count mode	GRBFx
Edge input interval measurement mode	GRAFx
Input high/low period measurement mode	
PWM input waveform measurement mode	
Rotation speed/pulse measurement mode	ECNTBFx
Up/down count mode	GRBFx
Four-time multiplication event count mode	

Note: CDRF0 to CDRFSB_F-1 are initialized to FFFF FFFF_H after reset.

Depending on the setting of bits BKENAFx, BKENCx, and BKENDFx of backup control register Fx (BKCRFx), reading CDRFx triggers value saving from general register AFx (GRAFx) to backup general register AFx (BGRAFx), from general register CFx (GRCFx) to backup general register CFx (BGRCFx), and from general register DFx (GRDFx) to backup general register DFx (BGRDFx). For more information, see the sections on backup general registers corresponding to individual general registers.

30.10.2.23 NCNTFAX — Noise Canceler Counters FAX

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NCNTFAX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_F-1: Corresponding to subblocks F00 to FSB_F-1

Table 30.192 NCNTFAX Register Contents

Bit Position	Bit Name	Function
15 to 0	NCNTFAX	Noise cancel counters FAX 16-bit count value

Note: x indicates an integer from 0 to SB_F-1.

Noise canceler counters registers FAX (NCNTFAX) are 16-bit readable/writable registers.

When the noise canceler is enabled by noise canceler control register Fx (NCCRFx) and the operation mode is the premature-transition cancellation mode or minimum time-at-level cancellation mode, a level change of external input pin TIFxA triggers the up-count operation ticked by the noise canceler counter clock provided by the prescaler. In level accumulation cancellation mode, up/down counting is performed based on the external input level.

NCNTFAX is readable/writable only in 16-bit units.

NCNTFAX is initialized to 0000_H after reset.

One of the three different operation modes, which are the premature-transition cancellation mode, the minimum time-at-level cancellation mode, and the level accumulation cancellation mode, takes place depending on the setting of the noise cancel mode selection bit (NCMSEL) and the timer F noise cancel mode bit (NCMF) of the noise cancel mode register (NCMR) of the common controller, as well as on a couple of timer F bits, which are the noise cancel mode bit (NCM1Fx) of the noise cancel mode channel register 1F (NCMCR1F) and the noise cancel mode bit (NCM2Fx) of the noise cancel mode channel register 2F (NCMCR2F).

- Premature-transition cancellation mode

NCNTFAX starts up-count operation triggered by the level change of input signal in TIFxA under the condition that the NCEFx bit is set to 1 and NCNTFAX is not in count operation. When the count number matches the value in the noise cancel register FAX (NCRFAX), this register stops the count operation, clearing the count value to 0000_H synchronizing with the next PCLK.

NCNTFAX executes the count operation regardless of the setting of the TFE bit in the ATU master enable register (ATUENR).

A level change at the start of count operation is output as it is as the signal that passed through the noise canceling operation. Although this signal is the subject of edge detection, the signal does not change because any input level change is masked until the count number matches the value of NCRFAX.

Even if the NCEFx bit is cleared during the count operation, the count operation continues until the count number matches the value of NCRFAX. The input signal is masked during all that time.

- Minimum time-at-level cancellation mode

NCNTFAX starts up-count operation triggered by the level change of input signal in TIFxA under the condition that the NCEF_x bit is set to 1 and NCNTFAX is not in count operation. When the level change of the input signal occurs during the count operation or the count number matches the value in the noise cancel register FAX (NCRFAX), this register stops the count operation, clearing the count value to 0000_H synchronizing with the next PCLK.

NCNTFAX executes the count operation regardless of the setting of the TFE bit in the ATU master enable register (ATUENR).

Signals that passed through the noise canceling operation can change only when the count number matches the value of NCRFAX according to the level change at the count start. If the count operation stops before the count number matches the value of NCRFAX, signals that passed through the noise canceling operation do not change because the level changes at the start or stop of counting are masked.

Even if the NCEF_x bit is cleared during the count operation, the count operation and noise canceling processing continue until a compare match occurs or the level of the input signal changes.

- Level accumulation cancellation mode

When the NCEF_x bit is set to 1, NCNTFAX performs up or down counting depending on the input signal level. Up-counting is performed when the input level is high, and if the counter value coincides with NCRFAX, the up-counting stops at the next PCLK cycle. Down-counting is performed when the input level is low, and if the counter value coincides with 0000_H, the down-counting stops at the next PCLK cycle.

Counting of NCNTFAX is performed regardless of the setting of the TFE bit of the ATU master enable register (ATUENR).

If up-counting results in a compare match of NCRFAX, the noise canceler output is updated to 1. If down-counting results in a compare match with 0000_H, the noise canceler output is updated to 0.

If the NCEF_x bit is cleared when counting is in progress, the noise cancel counter stops counting, and the value is changed from the noise canceler output to the current input signal level. Be noted that clearing the NCEF_x bit in level accumulation cancellation mode might cause an edge detection due to this value change.

30.10.2.24 NCNTFBx — Noise Canceler Counters FBx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NCNTFBx[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to 2: Corresponding to subblocks F00 to F02

Table 30.193 NCNTFBx Register Contents

Bit Position	Bit Name	Function
15 to 0	NCNTFBx	Noise cancel counters FBx 16-bit count value

Note: x indicates an integer from 0 to 2.

Noise cancel counters FBx (NCNTFBx) are 16-bit readable/writable registers.

These registers are available only in up/down event count mode and 4-time multiplication event count mode.

When the noise canceler is enabled by noise canceler control register Fx (NCCRF) and the operation mode is premature-transition cancellation mode or minimum time-at-level cancellation mode, a level change of external input pin TIFxB triggers the up-count operation ticked by the noise canceler counter clock provided by the prescaler. In level accumulation cancellation mode, up/down counting is performed based on the external input level.

NCNTFBx is readable/writable only in 16-bit units.

NCNTFBx is initialized to 0000_H after reset.

One of the three different operation modes, which are the premature-transition cancellation mode, the minimum time-at-level cancellation mode, and the level accumulation cancellation mode, takes place depending on the setting of the noise cancel mode selection bit (NCMSEL) and the timer F noise cancel mode bit (NCMF) of the noise cancel mode register (NCMR) of the common controller, as well as on a couple of timer F bits, which are the noise cancel mode bit (NCM1Fx) of the noise cancel mode channel register 1F (NCMCR1F) and the noise cancel mode bit (NCM2Fx) of the noise cancel mode channel register 2F (NCMCR2F).

- Premature-transition cancellation mode

NCNTFBx starts up-count operation triggered by the level change of input signal in TIFxB under the condition that the NCEFx bit is set to 1 and NCNTFBx is not in count operation. When the count number matches the value in the noise cancel register FBx (NCRFBx), this register stops the count operation, clearing the count value to 0000_H synchronizing with the next PCLK.

NCNTFBx executes the count operation regardless of the setting of the TFE bit in the ATU master enable register (ATUENR).

A level change at the start of count operation is output as it is as the signal that passed through the noise canceling operation. Although this signal is the subject of edge detection, the signal does not change because any input level change is masked until the count number matches the value of NCRFBx.

Even if the NCEFx bit is cleared during the count operation, the count operation continues until the count number matches the value of NCRFBx. The input signal is masked during all that time.

- Minimum time-at-level cancellation mode

NCNTFBx starts up-count operation triggered by the level change of input signal in TIFxB under the condition that the NCEF_x bit is set to 1 and NCNTFBx is not in count operation. When the level change of the input signal occurs during the count operation or the count number matches the value in the noise cancel register FBx (NCRFBx), this register stops the count operation, clearing the count value to 0000_H synchronizing with the next PCLK.

NCNTFBx executes the count operation regardless of the setting of the TFE bit in the ATU master enable register (ATUENR).

Signals that passed through the noise canceling operation can change only when the count number matches the value of NCRFBx, according to the level change at the count start. If the count operation stops before the count number matches the value of NCRFBx, signals that passed through the noise canceling operation do not change because the level changes at the start or stop of counting are masked.

Even if the NCEF_x bit is cleared during the count operation, the count operation and noise canceling processing continue until a compare match occurs or the level of the input signal changes.

- Level accumulation cancellation mode

When the NCEF_x bit is set to 1, NCNTFBx performs up or down counting depending on the input signal level. Up-counting is performed when the input level is high, and if the counter value coincides with NCRFBx, the up-counting stops at the next PCLK cycle. Down-counting is performed when the input level is low, and if the counter value coincides with 0000_H, the down-counting stops at the next PCLK cycle.

Counting of NCNTFBx is performed regardless of the setting of the TFE bit of the ATU master enable register (ATUENR).

If an up-count results in a compare match of NCRFBx, the noise canceler output is updated to 1. If down-counting results in a compare match with 0000_H, the noise canceler output is updated to 0.

If the NCEF_x bit is cleared when counting is in progress, the noise cancel counter stops counting, and the value is changed from the noise canceler output to the current input signal level. Be noted that clearing the NCEF_x bit in level accumulation cancellation mode might cause edge detection due to this value change.

30.10.2.25 NCRFAx — Noise Cancel Registers FAx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NCTFAx[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.194 NCRFAx Register Contents

Bit Position	Bit Name	Function
15 to 0	NCTFAx	Noise cancel time FAx TIFxA noise cancel period (16-bit compare value)

Note: x indicates an integer from 0 to SB_F-1.

Noise cancel registers (NCRFAx) are 16-bit readable/writable registers that set the upper limit of the noise canceler counter (NCNTFAx). When a period of 128 PCLK cycles is selected as the noise cancel clock, setting of FFFF_H can cancel noise for maximum of 0.21 second period (when PCLK = 40 MHz).

One of the three different operation modes, which are the premature-transition cancellation mode, the minimum time-at-level cancellation mode, and the level accumulation cancellation mode, takes place depending on the setting of the noise cancel mode selection bit (NCMSEL) and the timer F noise cancel mode bit (NCMF) of the noise cancel mode register (NCMR) of the common controller, as well as on a couple of timer F bits, which are the noise cancel mode bit (NCM1Fx) of the noise cancel mode channel register 1F (NCMCR1F) and the noise cancel mode bit (NCM2Fx) of the noise cancel mode channel register 2F (NCMCR2F).

- Premature-transition cancellation mode

While NCNTFAx is in count operation, the level change of the subsequent input signal is masked. Values in NCNTFAx and NCRFAx are always compared. If a compare match occurs, these registers clear the value in NCNTFAx synchronizing with the next PCLK, stop the count operation, and cancel the masking of the input signal.

- Minimum time-at-level cancellation mode

While NCNTFAx is in count operation, noise canceler processing waiting state is entered. Values in NCNTFAx and NCRFAx are always compared. If a compare match occurs, these registers clear the value in NCNTFAx synchronizing with the next PCLK, stop the count operation, and then cancel the masking of the input signal and the noise canceler outputs the input signal that has passed through the noise canceling processing.

- Level accumulation cancellation mode

When NCNTFAx up-counting is in progress, values of CNTFAx and NCRFAx are compared. When a compare match occurs, the up-counting of NCNTFAx stops at the next PCLK cycle. When NCNTFAx down-counting is in progress, NCNTFAx is compared with 0000_H.

NCRFAx is readable/writable only in 16-bit units.

NCRFAx is initialized to 0000_H after reset.

30.10.2.26 NCRFBx — Noise Cancel Registers FBx

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NCTFBx[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to 2: Corresponding to subblocks F00 to F02

Table 30.195 NCRFBx Register Contents

Bit Position	Bit Name	Function
15 to 0	NCTFBx	Noise cancel time FBx TIFxB noise cancel period (16-bit compare value)

Note: x indicates an integer from 0 to 2.

Noise cancel registers (NCRFBx) are 16-bit readable/writable registers that set the upper limit of the noise canceler counter (NCNTFBx). When a period of 128 PCLK cycles is selected as the noise cancel clock, setting of FFFF_H can cancel noise for maximum of 0.21 second period (when PCLK = 40 MHz). This register is effective only during up/down counting and the four-time multiplication event count mode.

One of the three different operation modes, which are the premature-transition cancellation mode, the minimum time-at-level cancellation mode, and the level accumulation cancellation mode, takes place depending on the setting of the noise cancel mode selection bit (NCMSEL) and the timer F noise cancel mode bit (NCMF) of the noise cancel mode register (NCMR) of the common controller, as well as on a couple of timer F bits, which are the noise cancel mode bit (NCM1Fx) of the noise cancel mode channel register 1F (NCMCR1F) and the noise cancel mode bit (NCM2Fx) of the noise cancel mode channel register 2F (NCMCR2F).

- Premature-transition cancellation mode

While NCNTFBx is in count operation, the level change of the subsequent input signal is masked. Values in NCNTFBx and NCRFBx are always compared. If a compare match occurs, these registers clear the value in NCNTFBx synchronizing with the next PCLK, stop the count operation, and cancel the masking of the input signal.

- Minimum time-at-level cancellation mode

While NCNTFBx is in count operation, noise canceler processing waiting state is entered. Values in NCNTFBx and NCRFBx are always compared. If a compare match occurs, these registers clear the value in NCNTFBx synchronizing with the next PCLK, stop the count operation. Simultaneously the noise canceler outputs the input signal that has passed through noise canceling processing.

- Level accumulation cancellation mode

When NCNTFBx up-counting is in progress, values of CNTFBx and NCRFBx are compared. When a compare match occurs, the up-counting of NCNTFBx stops at the next PCLK cycle. When NCNTFBx down-counting is in progress, NCNTFBx is compared with 0000_H.

NCRFBx is readable/writable only in 16-bit units.

NCRFBx is initialized to 0000_H after reset.

For details of operations for noise cancellation, see the description of the noise cancellation mode register in **Section 30.3.2, Registers Related to Common Controller**.

30.10.3 Details of Operation

30.10.3.1 Edge Counting in a Given Time

When a period over which edges are counted is set in GRAF_x, the number of edges within the period is obtained in GRBF_x. When no edge is detected within the period, 0 is set to GRBF_x. The period set to count is equivalent to the cycle of the ECNTAF_x clock (GRAF_x value). Operation of the timer F_x is described below. **Figure 30.83** shows an operation example. In this example, eight edges are input to 12 cycles of the count source clock. Timer counter ECNTAF_x and event counter ECNTBF_x are driven by the ECNTAF_x and ECNTBF_x clocks, respectively.

- ECNTAF_x: Measures time using one of the clock buses 0 to 5. When a compare match is detected, the count value is cleared synchronized with the next PCLK.
- ECNTBF_x: Counts edges of the signals provided from TIF_xA input. Edge types subject to count can be selected from among rising, falling, or both edges. The example given here counts the falling edges. A delay of two cycles in TIF_xA occurs because of synchronization processing. When a compare match in ECNTAF_x is detected, the count number is cleared synchronized with the next PCLK. In a case where edges subject to count are given simultaneously at a count clearing by a compare match, both operations are regarded to be done in one cycle, setting the count value to 0001_H. **Figure 30.84** shows an example of this.
- GRAF_x: Functions as the compare match register for ECNTAF_x. A compare match is detected when the count values in ECNTAF_x and GRAF_x agree.
- GRBF_x: Functions as the capture register for ECNTBF_x. When a compare match in ECNTAF_x is detected, this register captures the ECNTBF_x count number synchronizing with the next PCLK.
- Compare match interrupt request output: After compare match detection of ECNTAF_x, a compare match interrupt request is issued to CPU at the next PCLK cycle.
- ICFF_x: After detecting a compare match in ECNTAF_x, sets the ICFF_x flag synchronized with the next PCLK.
- ECNTCF_x, GRCF_x, GRDF_x: Do not function.

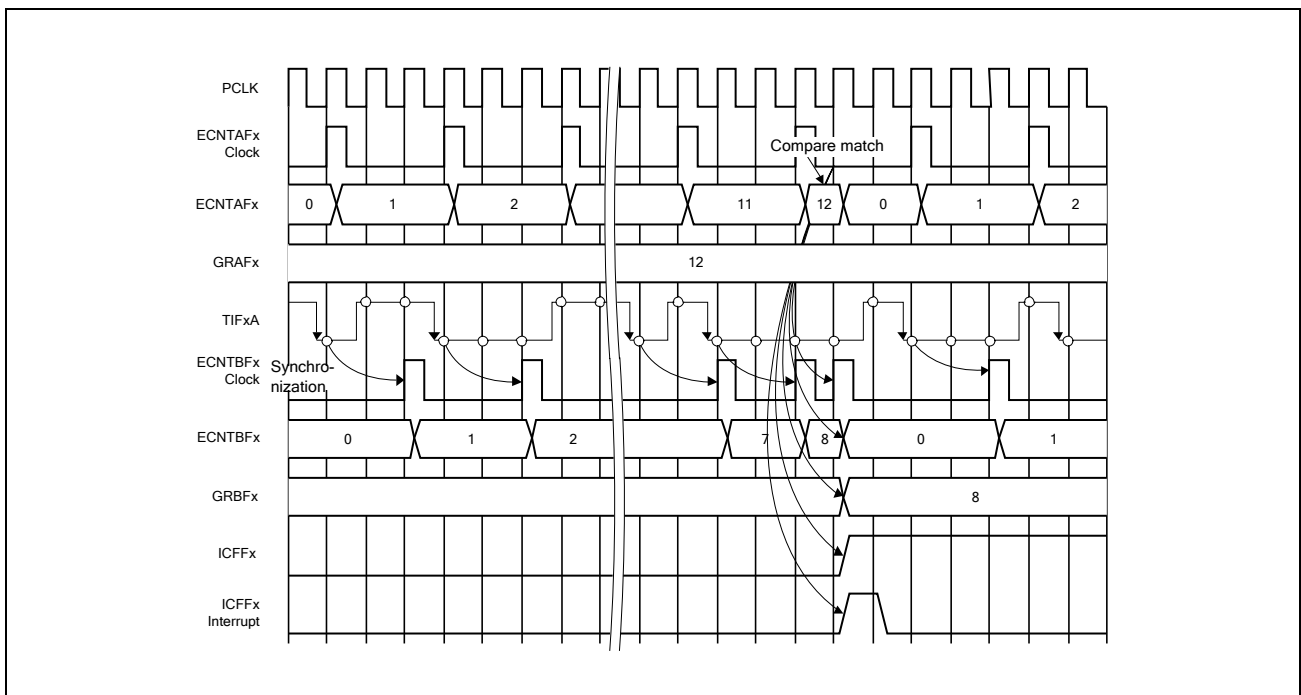


Figure 30.83 Operation Example of Edge Count in a Given Time

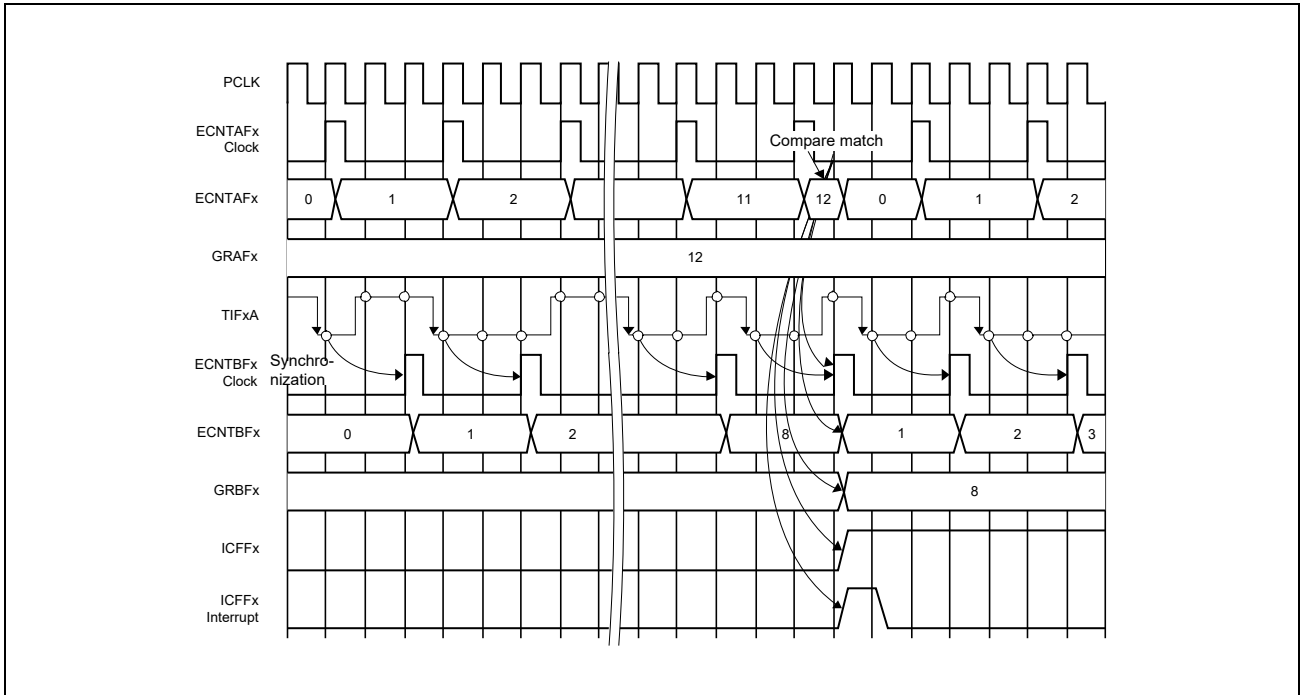


Figure 30.84 Operation Example 2 of Edge Count in a Given Time (Same Timing as Compare Match and Event)

30.10.3.2 Effective Edge Interval Counting

When a number of edges are set in GRBFx, the time necessary to count these edges is notified to GRAF_x. The average of input edge intervals is obtained by dividing the time by the number of edges. The outcome is given as the unit of the ECNTAF_x count source clock (GRAF_x). Operation of the timer F_x is described below. **Figure 30.85** shows an operation example. In this example, 13 cycles of the counter clock are needed to detect 12 input edges. Timer counter ECNTAF_x and event counter ECNTBF_x are driven by the ECNTAF_x and ECNTBF_x clocks, respectively.

- ECNTAF_x: Measures time using one of the clock buses 0 to 5. When a compare match between ECNTBF_x and GRBF_x is detected, the count value is cleared synchronized with the next ECNTAF_x clock. Since ECNTAF_x count clear occurs at the same time with count-up, the cleared value becomes 0000 0001_H.
- ECNTBF_x: Counts edges provided from TIFxA. Edge types subject to count can be selected from among rising, falling, or both edges. The example given here counts the falling edges. A delay of two cycles in TIFxA occurs because of synchronization processing. When a compare match is detected, the count number is cleared synchronized with the next PCLK.
- GRAF_x: Functions as the capture register for ECNTAF_x. When a compare match in ECNTBF_x is detected, this register captures the ECNTAF_x count number synchronizing with the next ECNTAF_x clock.
- GRBF_x: Functions as the compare match register for ECNTBF_x. A compare match is detected when the count values in ECNTBF_x and GRBF_x agree
- Compare match interrupt request: After a compare match detection of ECNTBF_x, a compare match interrupt request is issued to CPU at the next “ECNTAF_x Clock” cycle.
- ICFF_x: After detecting a compare match in ECNTBF_x, sets the ICFF_x flag synchronized with the next ECNTAF_x clock.
- ECNTCF_x, GRCF_x, GRDF_x: Do not function.

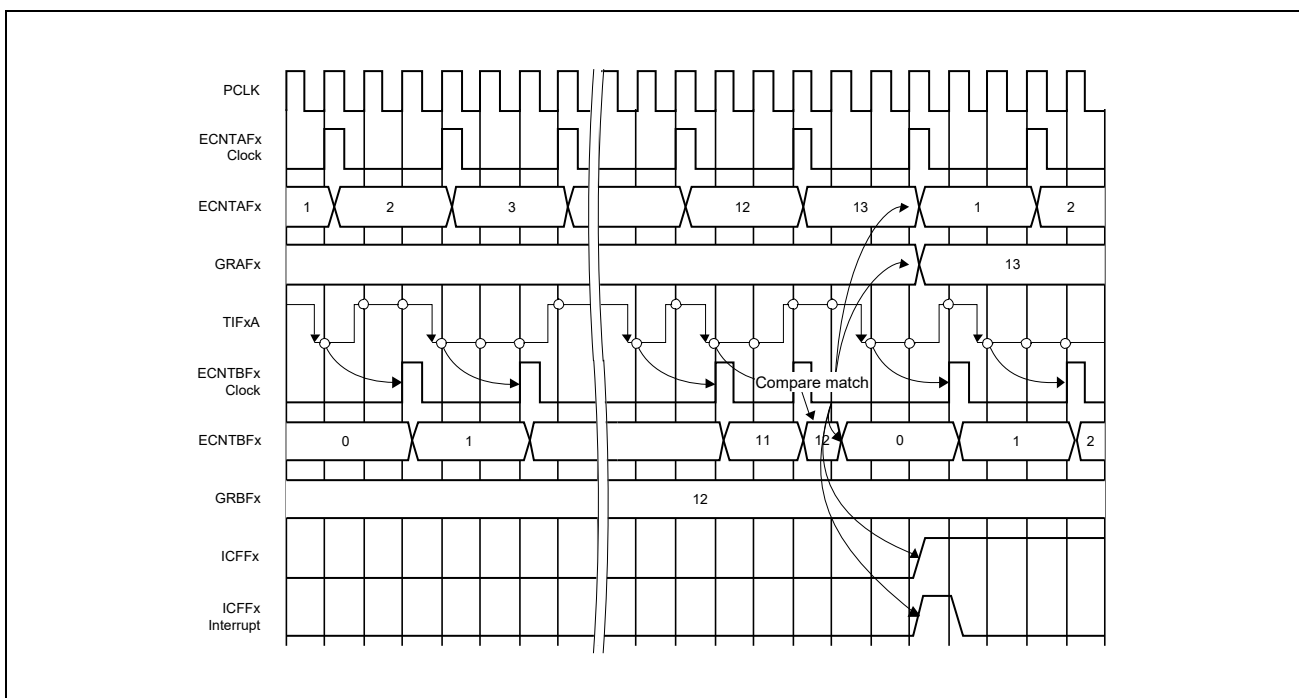


Figure 30.85 Operation Example of Effective Edge Interval Counting

30.10.3.3 Measurement of Time during High/Low Input Levels

Measures the time while TIFxA is driven high or low. The time obtained is indicated using the ECNTAFx clock source as the standard. The width of the measurement time is specified to GRBFx in the form of the pulse number provided for TIFxA (GRBFx value). Operation of the timer F is described below. **Figure 30.86** shows an operation example. This is the example in which the high level periods of the three pulses are measured as nine count source cycles. Timer counter ECNTAFx and event counter ECNTBFx are driven by the ECNTAFx and ECNTBFx clocks, respectively.

- ECNTAFx: Executes up-count using one of the clock buses 0 to 5 as a count source and TIFxA level as enable. Therefore, the time period in which TIFxA is in high level (EGSELFx = 2, falling edge selected), or TIFxA is in low level (EGSELFx = 1, rising edge selected) is measured. After detecting a compare match in ECNTBFx, this register clears the count number synchronizing with the next count source clock. If TIFxA is driven high (EGSELFx = 2) or low (EGSELFx = 1) at clearing count by the compare match, the count value becomes 0000 0001_H. **Figure 30.87** is an example of this.
- ECNTBFx: Counts the falling edge of the signal from the TIFxA input. Rising or falling edge is selectable to be counted. The operation example shows counting of the TIFxA falling edge. A delay of two cycles occurs because of synchronization processing. When a compare match is detected, the count number is cleared synchronized with the next PCLK.
- GRAFx: Functions as the capture register for ECNTAFx. When a compare match in ECNTBFx is detected, this register captures the ECNTAFx count number synchronizing with the next ECNTAFx clock.
- GRBFx: Functions as the compare match register for ECNTBFx. A compare match is detected when the count number in ECNTBFx and GRBFx agree.
- Compare match interrupt request: After a compare match detection of ECNTBFx, a compare match interrupt request is issued to CPU at the next “ECNTAFx Clock” cycle.
- ICFFx: After detecting a compare match in ECNTBFx, sets the ICFFx flag synchronized with the next ECNTAFx clock.
- ECNTCFx, GRCFx, GRDFx: Do not function.

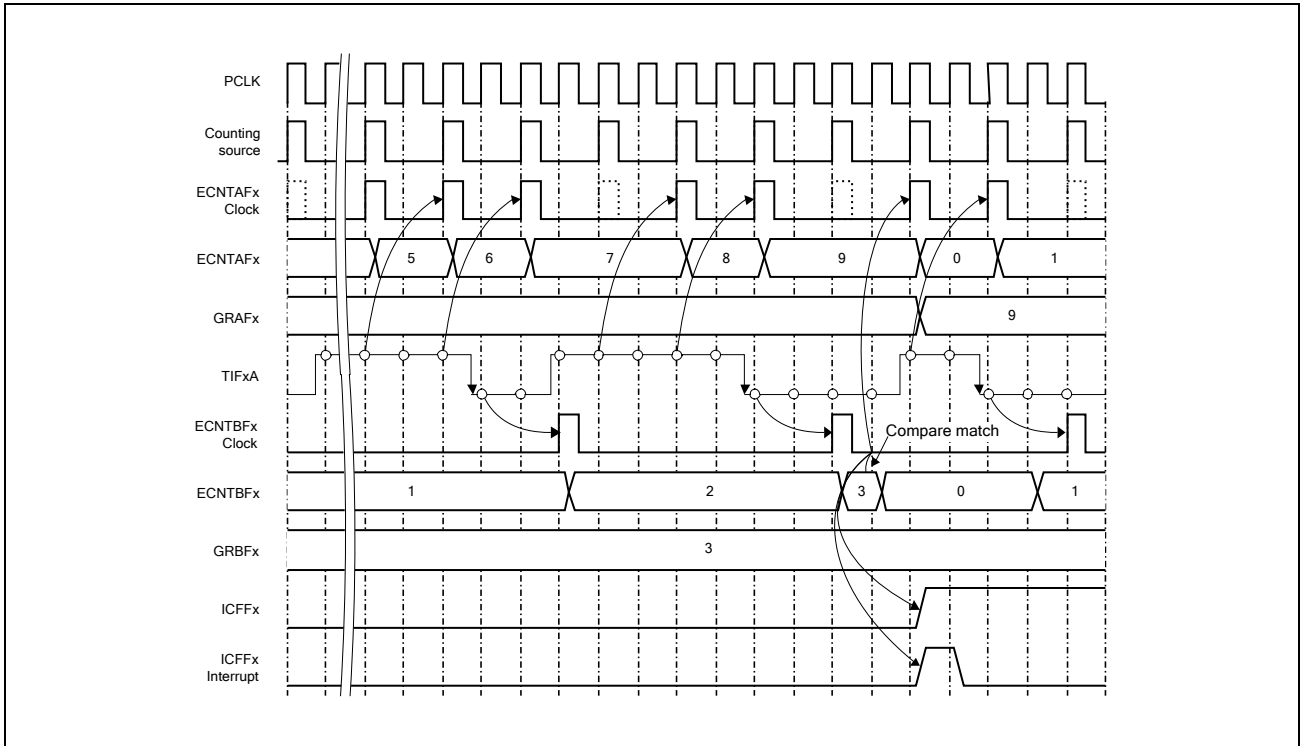


Figure 30.86 Operation Example of Measurement of Time during High Input Levels

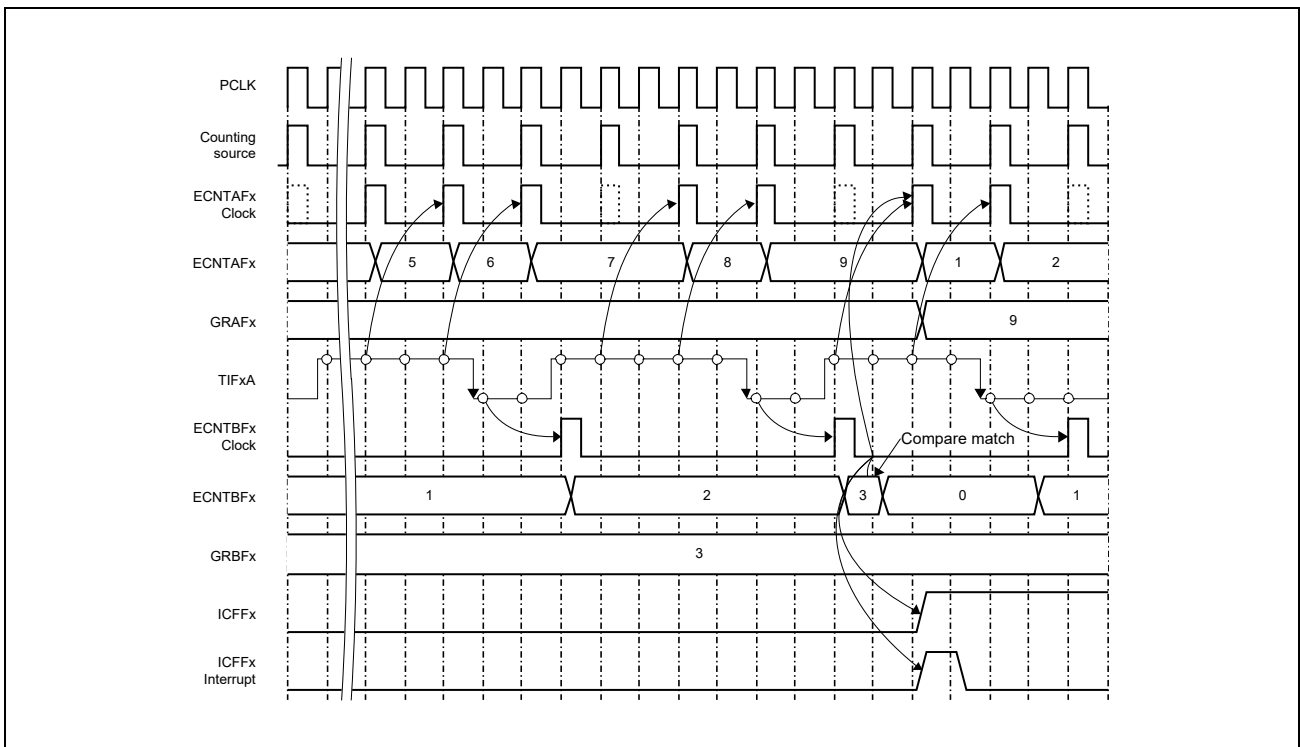


Figure 30.87 Operation Example of Measurement of Time during High Input Levels (TIFxA is in High Level When Capture is in Operation)

30.10.3.4 Measurement of PWM Input Waveform Timing

Measures the off-duty (non-active) period and cycle time of the PWM waveform input to TIFxA. The off-duty period is measured as the period of either the high or low level input on TIFxA, and the PWM cycle is measured as the interval between two rising or falling edges. Both are measured concurrently. The measured time is expressed in the number of cycles of the clock source for ECNTAFx. The duration of the measurement is set in GRBFx, which is specified as the number of PWM pulses input to TIFxA.

Compare match is possible for the number of PWM cycles. A compare match can issue an interrupt request to CPU.

Operation of timer F is described below. **Figure 30.88** shows an operation example. This is the example in which two PWM cycles in PWM waveform are measured as six counter clock cycles and the off-duty period (low-level period) is measured as four counter clock cycles.

The clocks for ECNTAFx, ECNTBFx, and ECNTCFx in this example provide the timing of counting or clearing operation of the time counter ECNTAFx, event counter ECNTBFx, and ECNTCFx, respectively.

- **ECNTAFx:** Executes up-count using one of the clock buses 0 to 5 as a count source and TIFxA input level as enable. Therefore, the time period in which TIFxA is in high level (EGSELFx = 2, falling edge selected), or TIFxA is in low level (EGSELFx = 1, rising edge selected) is measured. After detecting a compare match in ECNTBFx, this register clears the count number synchronizing with the next count source clock. If TIFxA is driven high (EGSELFx = 2) or low (EGSELFx = 1) at clearing count by the compare match, the count value becomes 0000 0001_H. The operation example shows measurement of low level.
- **ECNTBFx:** Counts the edge of the signal from the TIFxA input. Rising or falling edge is selectable to be counted. The operation example shows counting of the rising edge. A delay of two cycles occurs because of synchronization processing. When a compare match is detected, the count number is cleared synchronized with the next PCLK.
- **GRAFx:** Functions as the capture register for ECNTAFx. When a compare match in ECNTBFx is detected, this register captures the ECNTAFx count number synchronizing with the next ECNTAFx clock.
- **GRBFx:** Functions as the compare match register for ECNTBFx. A compare match is detected when the count number in ECNTBFx and GRBFx agree.
- **ECNTCFx:** Measures time using the same count source as ECNTAFx. This register clears the count number synchronizing with the next ECNTAFx clock after detecting a compare match in ECNTBFx. Since ECNTCFx count clear occurs at the same time with count-up, the cleared value is 0000 0001_H. A compare match with GRDFx does not clear ECNTCFx, and the counting continues.
- **GRCFx:** Functions as the capture register for ECNTCFx. This register captures the ECNTCFx count number synchronizing with the next ECNTAFx clock after detecting a compare match in ECNTBFx.
- **Input capture interrupt request:** After a compare match detection of ECNTBFx, an input capture interrupt request is issued to CPU at the next "ECNTAFx Clock" cycle.
- **ICFFx:** After detecting a compare match in ECNTBFx, sets the ICFFx synchronized with the next ECNTAFx clock.
- **GRDFx:** This register works as a compare match register of ECNTCFx. A compare match occurs when the ECNTCFx count coincides with GRDFx. (Effective when the GRDFCMEN bit of the private function control register (PVFCRF) is 1.)
- **Overflow interrupt request output:** After a compare match detection of ECNTCFx, an overflow interrupt request is issued to CPU at the next PCLK cycle. (Effective when the GRDFCMEN bit of the private function control register (PVFCRF) is 1.)
- **OVFCFx:** After detection of a compare match of ECNTCFx, OVFCFx is set at the next PCLK cycle. (Effective when the GRDFCMEN bit of the private function control register F (PVFCRF) is 1.)

Therefore, ECNTBFx (GRBFx) and ECNTAFx (GRAF_x) are operating in measurement of time during low input levels mode and ECNTBFx (GRBFx) and ECNTCFx (GRCF_x) are operating in effective edge interval counting mode.

The measured values in PWM input waveform measurement mode are saved in two 32-bit registers: GRAF_x and GRCF_x. Be noted that, when reading these two registers, a capture might occur in the period between reading one register and reading the other register, resulting in reading incorrect measurements (for example, when an interrupt request was processed in between).

Reading values from backup registers would provide two consistent values. For details, see **Section 30.10.3.9, Simultaneous Access of Multiple Registers**, described below.

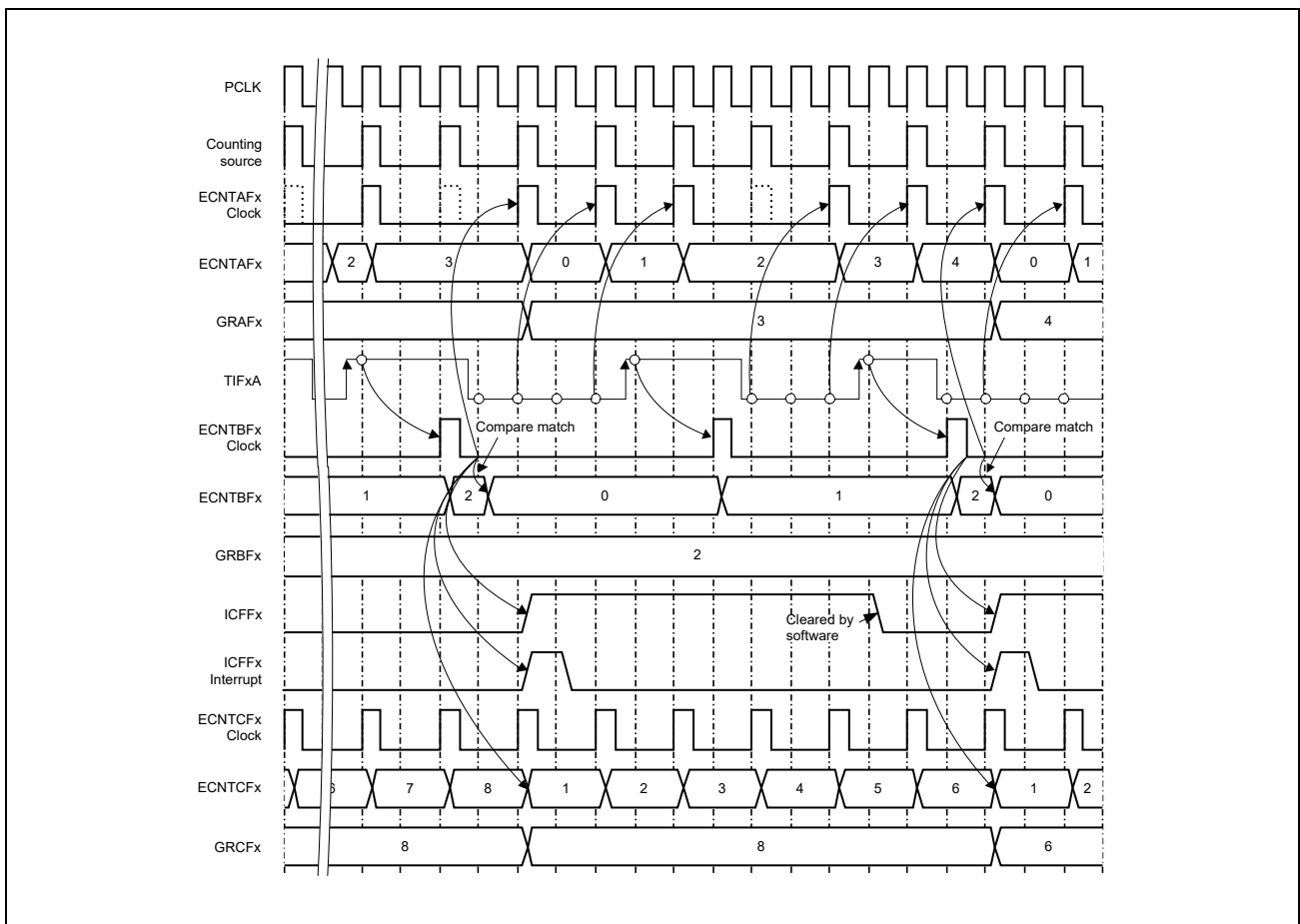


Figure 30.88 Operation Example of Measurement of PWM Input Waveform Timing

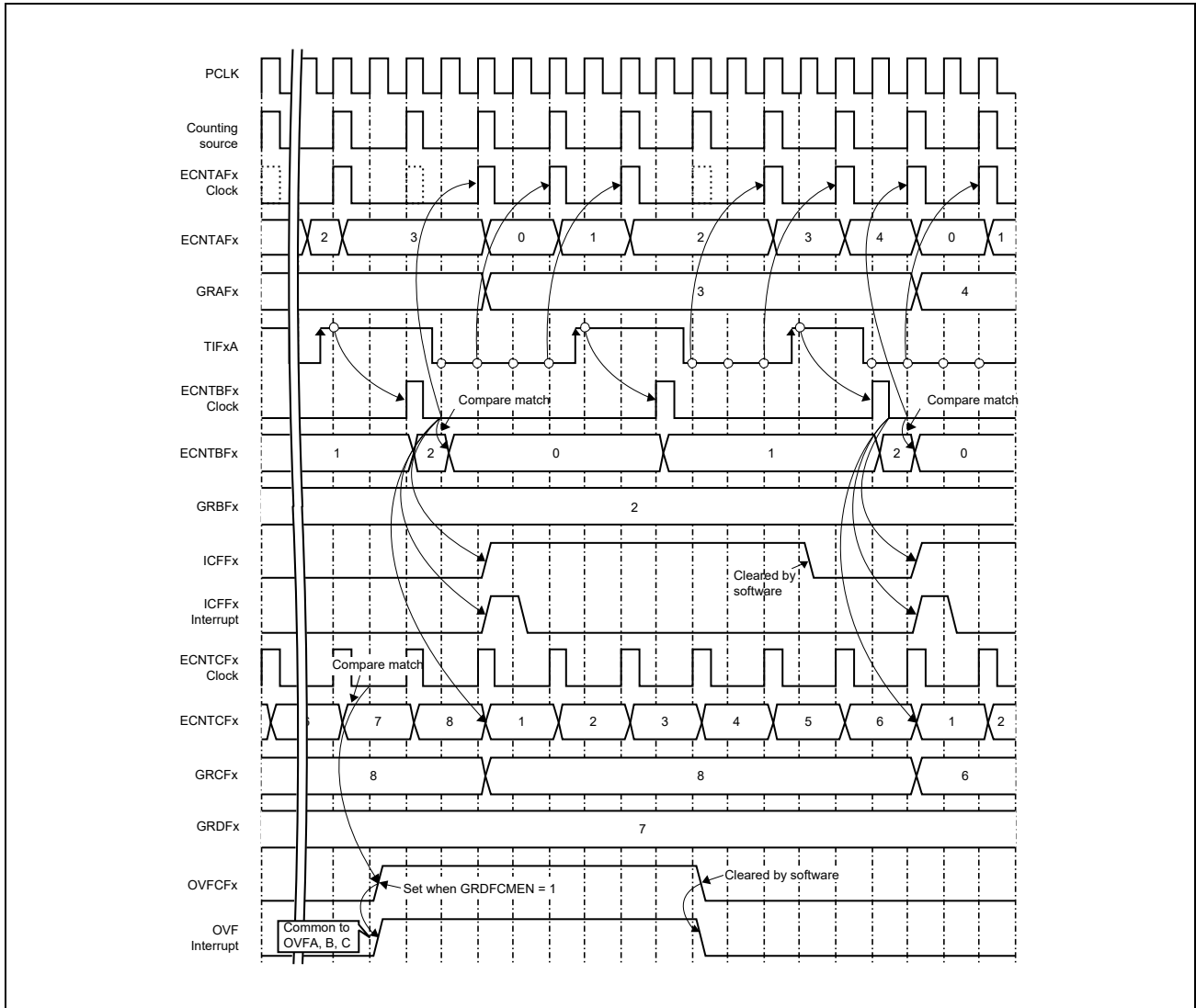


Figure 30.89 Operation Example of Measurement of PWM Input Waveform Timing

30.10.3.5 Rotation Speed/Pulse Measurement

Measures the number of edges input to TIFxA, the edge input time (time stamp), the off-duty period in the PWM waveform that emerges between the last input edge and the edge this time, and PWM cycle time.

The time obtained is expressed using the ECNTAFx clock source as the standard. The maximum interval of edge input can be set to GRBFx, which enables to output an interrupt request if the edge input interval exceeds the maximum value.

At this moment, the timer F operates as shown below. **Figure 30.89** shows an example of operation.

The ECNTAFx clock, ECNTBFx clock, and ECNTCFx clock show the timing of count operation or clearing for the time counter ECNTAFx, event counter ECNTBFx, and ECNTCFx, respectively.

- **ECNTAFx:** Executes up-count using one of the clock buses 0 to 5 as a count source and TIFxA input level as enable. Therefore, the time period in which TIFxA is in high level (EGSELFx = 2, falling edge selected), or TIFxA is in low level (EGSELFx = 1, rising edge selected) is measured. After inputting the edge to TIFxA, this register clears the count number synchronizing with the next count source clock. If TIFxA is driven high (EGSELFx = 2) or low (EGSELFx = 1) at clearing count by the compare match, the count value becomes 0000 0001_H.
- **ECNTBFx:** Counts the edge of the signal from the TIFxA input. Rising or falling edge is selectable to be counted. The operation example shows counting of the falling edge. A delay of two cycles occurs because of synchronization processing.
- **GRAFx:** Functions as the capture register for ECNTAFx. This register captures the ECNTAFx count number synchronizing with the next ECNTAFx clock after inputting the edge to TIFxA.
- **GRBFx:** Functions as the capture register for ECNTCFx. When the ECNTCFx count and the value in the 16 lower-order bits in GRBFx extended with 0 match, this register detects a compare match and set the OVFCFx to 1.
- **ECNTCFx:** Measures time using the same count source as ECNTAFx. This register clears the count number synchronizing with the next ECNTAFx clock after inputting the edge to TIFxA. Since ECNTCFx count clear occurs in the same timing, the cleared value is 0000 0001_H.
- **GRCFx:** Functions as the capture register for ECNTCFx. This register captures the ECNTCFx count number synchronizing with the next ECNTAFx clock after inputting the edge to TIFxA.
- **GRDFx:** Functions as the capture register for ECNTCFx. This register captures the ECNTAFx, whose number being accumulated to GRDFx, synchronizing with the next ECNTAFx clock after inputting the edge to TIFxA. The value to be added is the ECNTCFx value before clearing. (Capture value: GRDFx+ = ECNTCFx).
- **Input capture interrupt request output:** After an edge input to TIFxA, an input capture interrupt request is issued to CPU at the next “ECNTAFx Clock” cycle.
- **ICFFx:** Sets the ICFFx synchronizing with the next ECNTAFx clock after inputting the edge to TIFxA.
- **Overflow interrupt request output:** An overflow interrupt request is issued to CPU at the next PCLK cycle during which values of ECNTCFx and GRBFx coincide (zero extension of 16 lower-order bits).
- **OVFCFx:** Sets the OVFCFx synchronizing with the next PCLK after the values in ECNTCFx and GRBFx (zero extension of 16 lower-order bits) match.

At the time of an input capture interrupt request, reading ECNTBFx, GRAFx, GRCFxF, and GRDFx provides the number of edges, the off-state duty cycle, the PWM cycle, and the edge input time, respectively.

The capture timing of GRAFx, GRCFxF, and GRDFx synchronizes with the count clock of ECNTAFx. Note that if the edge input cycle is shorter than the ECNTAFx count clock cycle, incorrect measurement may occur.

The measured values in “rotation speed/pulse measurement” mode are saved in three 32-bit registers GRAFx, GRCFxF, and GRDFx. Be noted that, when reading these three registers, a capture might occur in the period between reading one register and reading another register, resulting in reading incorrect measurements (for example, when an interrupt request was processed in between).

Reading values from backup registers would provide three consistent values. For details, see **Section 30.10.3.9, Simultaneous Access of Multiple Registers**, described below.

Note: Do not set this mode for subblocks other than 03 to SB_F-1.

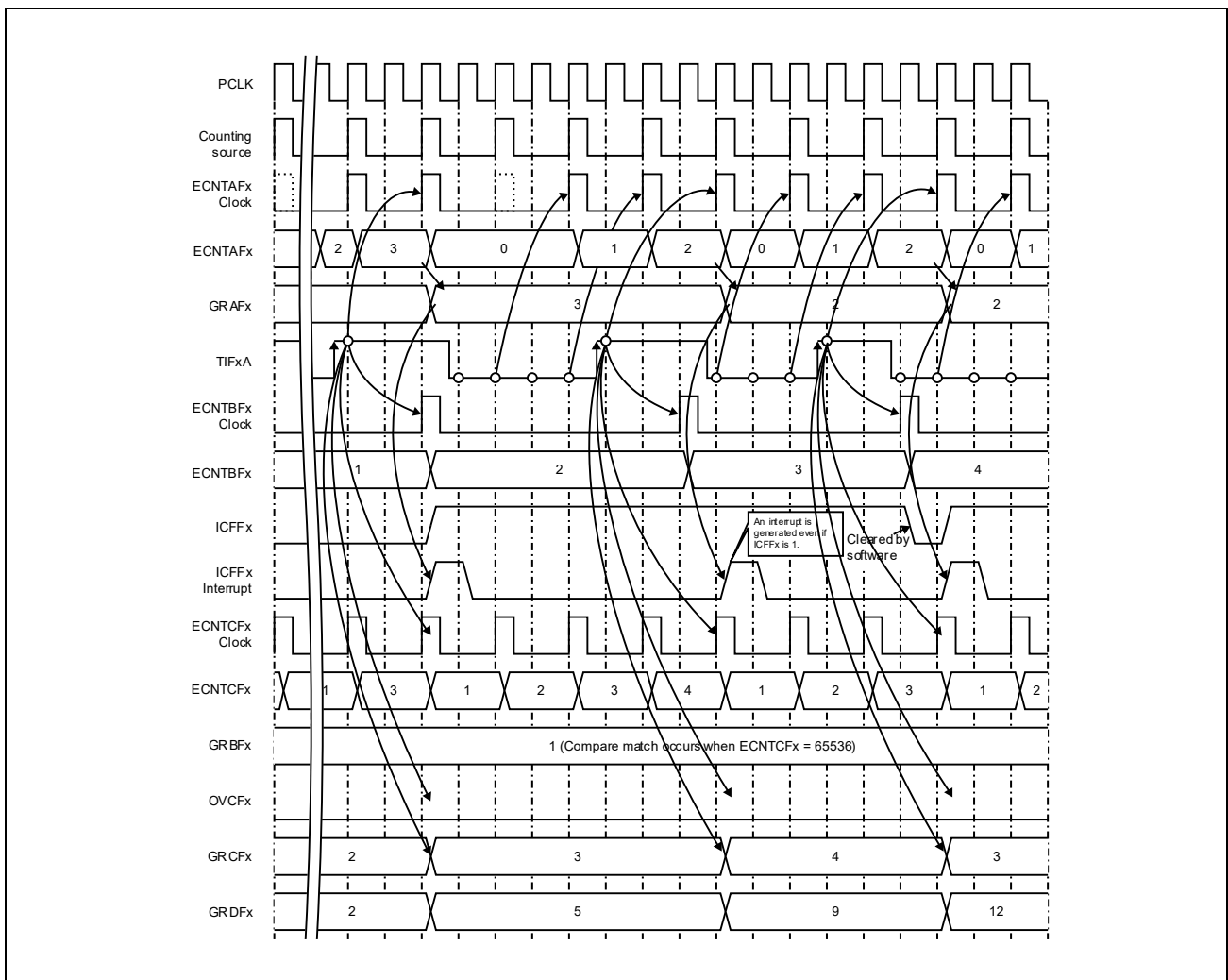


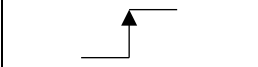
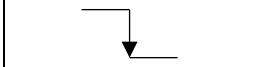
Figure 30.90 Operation Example of Rotation Speed/Pulse Measurement

30.10.3.6 Up/Down Event Count

This register uses the TIFxA pin, one of the two external input pins (TIFxA, TIFxB), as the count source, and TIFxB switches up-count to and from down-count. If a count period is designated to GRAFx the count number after designation can be obtained in GRBFx. The counting period is the period of ECNTAFx count source clock (GRAFx value). At this moment, the timer F operates as shown below. **Figure 30.90** shows an example of operation. The ECNTAFx clock and ECNTBFx clock show the timing that time counter ECNTAFx and event counter ECNTBFx execute the count operation or clearing, respectively.

- ECNTAFx: Measures time using one of the clock buses 0 to 5. When a compare match is detected, the count value is cleared synchronized with the next PCLK.
- ECNTBFx: Upcount/downcount operation is performed at both rising and falling edges of TIFxA. Count direction is determined by the TIFxB input level. (See **Table 30.196**.) Because of synchronization processing, a delay of two cycles occurs in TIFxA and TIFxB.
- GRAFx: Functions as the compare match register for ECNTAFx. A compare match is detected when the count number in ECNTAFx and GRAFx agree.
- GRBFx: Functions as the capture register for ECNTBFx. When a compare match in ECNTAFx is detected, this register captures the ECNTBFx count number synchronizing with the next PCLK.
- Input capture interrupt request output: After compare match detection of ECNTAFx, an interrupt request is issued to CPU at the next PCLK cycle.
- ICFFx: After detecting a compare match in ECNTAFx, sets the ICFFx flag synchronized with the next PCLK.
- ECNTCFx, GRCFx, GRDFx: Do not function.

Table 30.196 Count Direction in Up/Down Event Count Mode

Input	Count Direction	
	Upcount	Downcount
TIFxA		
TIFxB	Low	High

Note: Always set this mode only in subblock 0 to subblock 2.

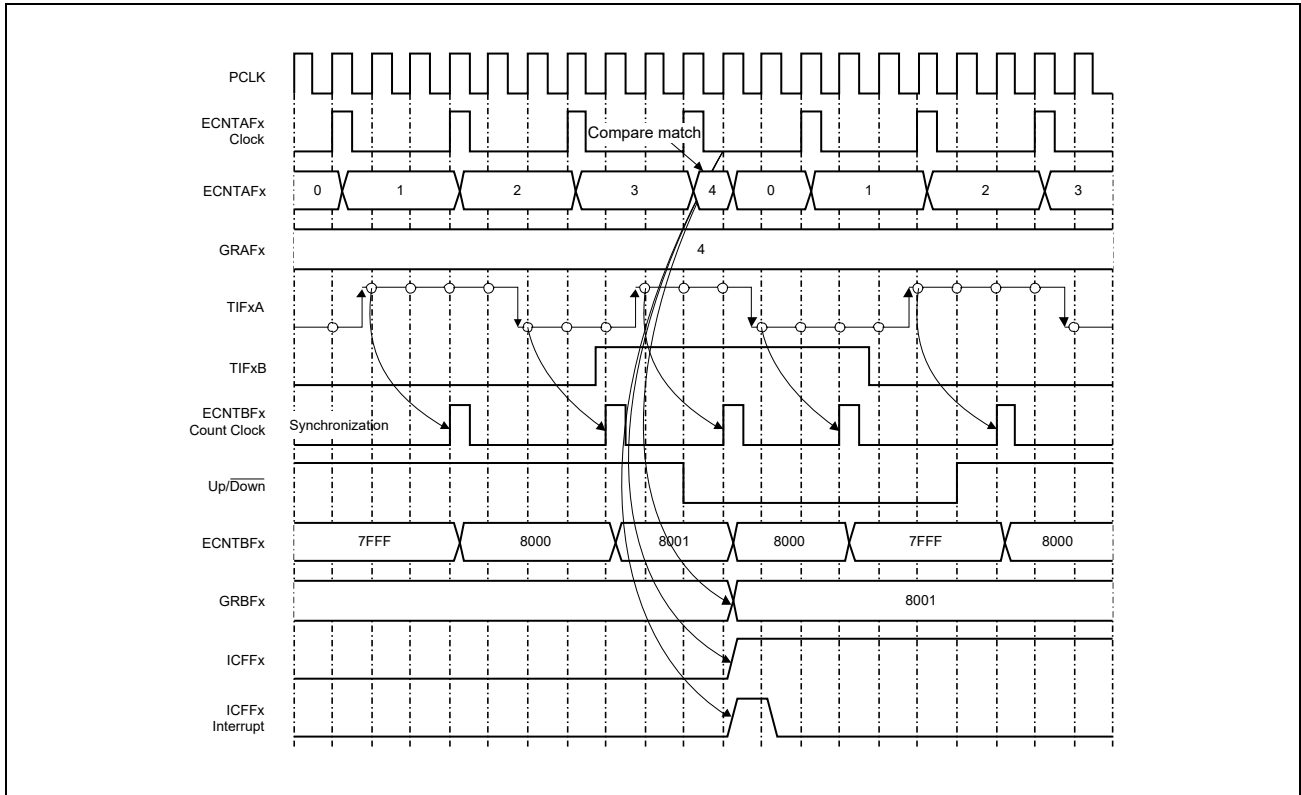


Figure 30.91 Operation Example of Up/Down Event Count

30.10.3.7 Four-time Multiplication Event Count

The count operation is executed using the external two input pins (TIFxA, TIFxB) as the count sources. Upcount or downcount is switched according to their input states. If a count period is designated to GRAF_x, the count number after designation can be obtained in GRBF_x. The counting period is the period of ECNTAF_x count source clock (GRAF_x value).

At this moment, the timer F operates as shown below. **Figure 30.91** shows an example of operation. The ECNTAF_x clock and ECNTBF_x clock show the timing that time counter ECNTAF_x and event counter ECNTBF_x execute the count operation or clearing, respectively.

- ECNTAF_x: Measures time using one of the clock buses 0 to 5. When a compare match is detected, the count value is cleared synchronized with the next PCLK.
- ECNTBF_x: Upcount/downcount operation is performed at both rising and falling edges of TIFxA and TIFxB respectively. Count direction is determined by the other signal input level. (See **Table 30.197.**) Because of synchronization processing, a delay of two cycles occurs in TIFxA and TIFxB.
- GRAF_x: Functions as the compare match register for ECNTAF_x. A compare match is detected when the count number in ECNTAF_x and GRAF_x agree.
- GRBF_x: Functions as the capture register for ECNTBF_x. When a compare match in ECNTAF_x is detected, this register captures the ECNTBF_x count number synchronizing with the next PCLK.
- Input capture interrupt request output: After a compare match detection of ECNTAF_x, an input capture interrupt request is issued to CPU at the next PCLK cycle.
- ICFF_x: After detecting a compare match in ECNTAF_x, sets the ICFF_x synchronized with the next PCLK.
- ECNTCF_x, GRCF_x, GRDF_x: Do not function.

Table 30.197 Count Direction in Four-time Multiplication Event Count Mode

Input	Count Direction							
	Upcount				Downcount			
TIFxA	High		Low		High		Low	
TIFxB		High		Low		High		High

NOTE

Operation when edge inputs in TIFxA and TIFxB are detected simultaneously is not guaranteed. The interval between edge inputs in TIFxA and TIFxB must be at least 1.5 cycles (PCLK).

Do not set this mode for the subblocks other than subblocks 0 to 2.

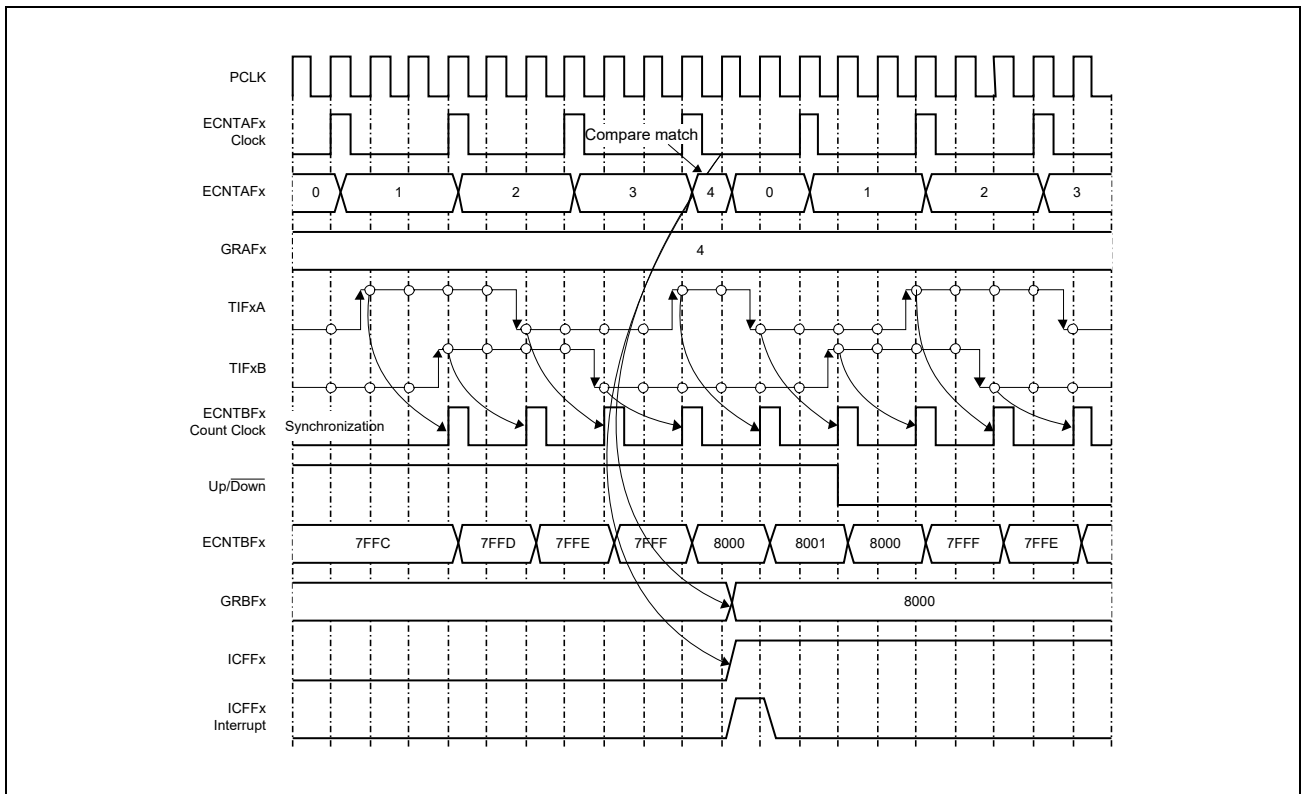


Figure 30.92 Operation Example of Four-time Multiplication Event Count Overflow/Underflow

30.10.3.8 Overflow and Underflow

Counter clearing of 0xFFFFFFFF (ECNTAFx and ECNTCFx) to 0x00000000 (ECNTAFx and ECNTCFx) or 0xFFFF (ECNTBFx) to 0x0000 (ECNTBFx) causes overflow detection. In this case, the overflow flag is reset and an overflow interrupt request is issued as soon as the counter value becomes 0x00000000 (or 0x0000). (A figure is omitted.) When an overflow of ECNTAFx is detected, OVFAFx is set and an overflow A interrupt request is issued. When an overflow of ECNTBFx is detected, OVFBFx is set and an overflow B interrupt request is issued. When an overflow of ECNTCFx is detected, OVFCFx is set and an overflow C interrupt request is issued.

An underflow is detected when the counter value changes from 0x0000 (ECNTBFx) to 0xFFFF (ECNTBFx). In this case, OVFBFx is reset and an overflow interrupt request is issued as soon as the counter value becomes 0xFFFF. (A figure is omitted.) OVFBFx is also set. An underflow occurs only with ECNTBFx, in which case an overflow B interrupt request is issued.

30.10.3.9 Simultaneous Access of Multiple Registers

In PWM input waveform measurement mode and rotation speed/pulse measurement mode, saving register values of measurement results to backup registers provides multiple register values of the same time point. When the CDRFx register is read, register values are saved from GRAFx to BGRAFx, from GRCFx to BGRCFx, and from GRDFx to BGRDFx.

For saving general register values to backup registers and for reading data from backup registers, the backup control register Fx (BKCRFx) must be set. For details, see the section above explaining backup control registers.

With the above method in PWM input waveform measurement mode, reading CDRFx and then BGRCFx provides the PWM cycle and the duty time of the same time point. In “rotation speed/pulse measurement” mode, reading CDRFx and then BGRAFx, BGRCFx, and BGRDFx provides data of the same time point.

30.11 Timer G

30.11.1 Overview of Operation

The timer G block consists of SB_G timer G subblocks.

A timer G subblock counts the input clock, and when a predetermined time period has elapsed, generates a pulse signal of one PCLK cycle period.

It is provided with external event synchronization mode in which counter operation starts when a synchronize with an AGCK timing signal from timer B.

An input clock can be selected for the counter from the six clocks on the clock bus.

Configuration

Timer G subblock consists of a 32-bit counter (TCNTGx), a compare match register (OCRGx), a reload register (RLDGx), and a controller.

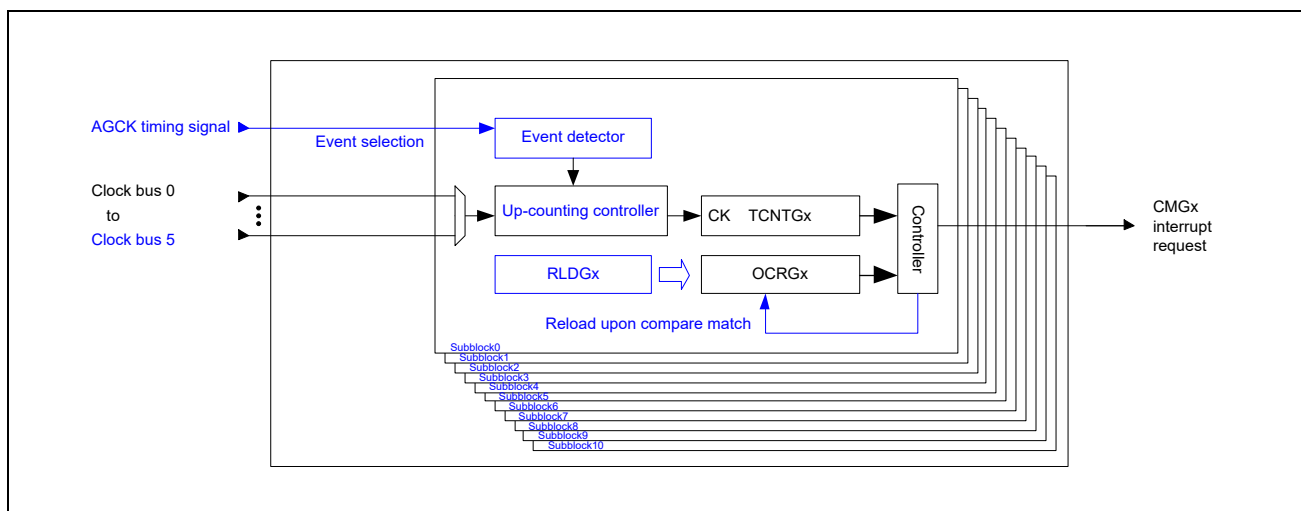


Figure 30.93 Block Diagram of Timer G Configuration

Interrupt Request

Timer G can issue SB_G interrupt requests as described below.

An interrupt request is issued upon compare match detection at timer subblock Gx.

30.11.2 Registers Related to Timer G

30.11.2.1 TSTRG — Timer Start Register G

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	STRG9	STRG8	STRG7	STRG6	STRG5	STRG4	STRG3	STRG2	STRG1	STRG0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.198 TSTRG Register Contents

Bit Position	Bit Name	Function
15 to SB _G	—	These bits are not used. Fix these bits to 0.
SB _G -1 to 0	STRGSB _G -1 to STRG0	Counter G start 0: TCNTG _x is disabled. 1: TCNTG _x is enabled.

Note: x indicates an integer from 0 to SB_G-1.

The timer start register G (TSTRG) is a 16-bit register. This register is an 8-bit/16-bit readable/writable register and specifies whether to start or stop each subblock (timer G0 to timer GSB_G-1) included in timer G. The counting is not performed if the TGE bit of the ATU master enable register (ATUENR) is not enabled even if the start bit of timer G permits counting.

TSTRG is initialized to 0000_H after reset.

Counter G Start (STRG_x)

These bits enable and disable timer counter (TCNTG_x) in the subblocks (G0 to GSB_G-1).

When these bits are cleared to “0”, TCNTG_x is stopped. TCNTG_x retains the value at that time while it is stopped.

When these bits are set to “1”, TCNTG_x resumes counting from the retained value. TCNTG_x runs when these bits and the TGE bit in ATU master enable register are both set to “1”.

CAUTION

The prescalers run independently of the setting of the counter G start bit and are not initialized when the TCNTG_x starts counting. Therefore, the time from when the STRG_x bit is set to when TCNTG_x starts counting may be less than the cycle of the selected count source clock (resolution), being undetermined due to hardware factors.

30.11.2.2 TIERG — Timer Interrupt Enable Register G

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CMPIE G9	CMPIE G8	CMPIE G7	CMPIE G6	CMPIE G5	CMPIE G4	CMPIE G3	CMPIE G2	CMPIE G1	CMPIE G0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.199 TIERG Register Contents

Bit Position	Bit Name	Function
15 to SB _G	—	These bits are not used. Fix these bits to 0.
SB _G -1 to 0	CMPIEGSB _G -1 to CMPIEG0	Compare match interrupt enable G _x 0: Disable interrupt request upon compare match 1: Enable interrupt request upon compare match

Note: x = 0 to SB_G-1: Corresponding to subblocks G0 to GSB_G-1

The timer interrupt enable register G (TIERG) is a 16-bit register. This register is an 8-bit/16-bit readable/writable register and specifies whether to enable or disable issuance of an interrupt request upon compare match output from each subblock (timer G0 to timer GSB_G-1). TIERG is initialized to 0000_H after reset.

30.11.2.3 RLDCRG — Reload Control Register G

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RLDEN G9	RLDEN G8	RLDEN G7	RLDEN G6	RLDEN G5	RLDEN G4	RLDEN G3	RLDEN G2	RLDEN G1	RLDEN G0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.200 RLDCRG Register Contents

Bit Position	Bit Name	Function
15 to SB _G	—	These bits are not used. When read, “0” is returned. When writing, write “0”.
SB _G -1 to 0	RLDENG _x	Reload Enable G _x 0: Disable the compare match register reload function upon compare match 1: Enable the compare match register reload function upon compare match

Note: x = 0 to SB_G-1: Corresponding to subblocks G0 to GSB_G-1

Reload control registers G_x (RLDCRG_x) are 8-bit/16-bit readable/writable registers. RLDENG_x enables and disables the reload function.

RLDCRG_x is initialized to 0000_H after reset.

Reload Enable G_x (RLDENG_x)

When this bit is set to “1” to enable the reload function, the value of the reload register (RLDG_x) is transferred to the compare match register (OCRG_x) upon compare match between the timer counter G_x (TCNTG_x) and compare match register (OCRG_x).

30.11.2.4 TCRGx — Timer Control Register Gx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	CKSELGx[2:0]			—	—	—	EVSIMGx
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R	R/W

Note: x = 0 to SB_G-1: Corresponding to subblocks G0 to GSB_G-1

Table 30.201 TCRGx Register Contents

Bit Position	Bit Name	Function
7	—	This bit is not used. When read, “0” is returned. When writing, write “0”.
6 to 4	CKSELGx[2:0]	Clock Select Gx These bits select the clock source of timer counter G (TCNTGx) of subblock. 000: Clock bus 0 001: Clock bus 1 010: Clock bus 2 011: Clock bus 3 100: Clock bus 4 101: Clock bus 5 110: Setting prohibited 111: Setting prohibited
3 to 1	—	These bits are not used. When read, “0” is returned. When writing, write “0”.
0	EVSIMGx	External event synchronization mode set Gx Specify the operation mode of timer subblock Gx 0: Normal count mode 1: External event synchronization mode

Note: x indicates an integer from 0 to SB_G-1.

Timer control registers Gx (TCRG0 to TCRGSB_G-1) are 8-bit readable/writable registers that set the operating mode of each subblock of timer G (G0 to GSB_G-1).

TCRG0 to TCRGSB_G-1 are initialized to 00_H after reset.

Clock Select Gx (CKSELGx)

These bits select the clock source of timer counter (TCNTGx) of subblocks (G0 to GSB_G-1).

Setting these bits to a value from “000” to “101” selects a clock source from clock bus 0 to clock bus 5. Do not set these bits to “111” or “111”. If “111” or “111” is set, the timer might not work properly.

External Event Synchronization Mode Set (EVSIMGx)

This bit specifies the operation mode for the timer counter Gx (TCNTGx) of timer subblocks G0 to GSB_G-1. If EVSIMGx is set to “1”, external event synchronization mode is set, an input AGCK timing signal is detected in timer B and a count is begun. When selecting the external event synchronous mode, select 101 (Clock bus 5 is selected in a clock source.) in CKSELGx [2:0] and set 1 as CB5SEL (Angle clock of timer B). Also, set CU4SEL of timer B as 1 (Until TCNTB3-1 counts TCNTB4 and, the clear value of TCNTB5 is set as 0.)

30.11.2.5 TSRGx — Timer Status Registers Gx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	EVDTFGx	OVFGx	CMFGx
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Note: x = 0 to SB_G-1: Corresponding to subblocks G0 to GSB_G-1

Table 30.202 TSRGx Register Contents

Bit Position	Bit Name	Function
7 to 2	—	These bits are not used. When read, “0” is returned. When writing, write “0”.
2	EVDTFGx	External event detection flag Gx 0: No external event has occurred. 1: An external event has occurred.
1	OVFGx	Overflow flag Gx 0: TCNTGx has not overflowed 1: TCNTGx has overflowed
0	CMFGx	Compare match flag Gx 0: Compare match has not occurred in subblocks Gx 1: Compare match has occurred in subblocks Gx

Note: x = 0 to SB_G-1: Corresponding to subblocks G0 to GSB_G-1

Timer status registers Gx (TSRG0 to TSRGSB_G-1) are 8-bit/16-bit read-only registers that measure the time and indicate occurrence of event counter overflow and compare match.

The overflow flag is the status flag that indicates the occurrence of overflow. It is not used to generate interrupts. The compare match flag is a status flag that indicates occurrence of an interrupt request. Setting a bit of the timer status clear register (TSCRGx) clears the corresponding flag. If an interrupt source takes effect when this flag is set, an interrupt request is issued again. An interrupt request is issued even if there is a conflict between clearing by the corresponding timer status clear register and setting by an interrupt source. When set and clear competed, clear is given priority to.

TSRG0 to TSRGSB_G-1 are initialized to 00_H after reset.

External Event Detection Flag Gx (EVDTFGx)

Indicates whether an external event is detected. This bit cannot be set to 0 or 1 by software.

- Setting (to 1) condition

When occurrence of the external event selected by input AGCKM timing signal is detected in timer B. (EVSIMGx = “1”)

- Clearing (to 0) condition

When the counter G start bit (STRGx) is set to “0”

Overflow Flag Gx (OVFGx)

Indicates whether timer counter Gx (TCNTGx) has overflowed. This bit cannot be set to 1 or 0 by software.

- Setting (to 1) condition

When TCNTGx has overflowed (from FFFF FFFF_H to 0000 0000_H)

- Clearing (to 0) condition

When “1” is written to OVFCGx in timer status clear register Gx (TSCRGx)

Compare Match Flag Gx (CMFGx)

Indicates whether a compare match has occurred in subblock Gx (G0 to GSB_G-1). This flag cannot be set to 1 or 0 by software.

- Setting (to 1) condition

When compare match is detected in subblock Gx

- Clearing (to 0) condition

When “1” is written to CMFCGx of the timer status clear register Gx (TSCRGx)

30.11.2.6 TSCRGx — Timer Status Clear Register Gx

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OVFCGx	CMFCGx
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W

Note: x = 0 to SB_G-1: Corresponding to subblocks G0 to GSB_G-1

Table 30.203 TSCRGx Register Contents

Bit Position	Bit Name	Function
7 to 2	—	These bits are not used. When read, “0” is returned. When writing, write “0”.
1	OVFCGx	Overflow flag clear Gx enable 0: Disabled (default) 1: Write 0 to OVFCGx
0	CMFCGx	Compare match flag clear Gx enable 0: Disabled (default) 1: Write 0 to CMFCGx

Note: x = 0 to SB_G-1: Corresponding to subblocks G0 to GSB_G-1

TSCRGx is an 8-bit readable/writable register that clears the overflow and compare match flags.

TSCRGx is readable/writable only in 8-bit units. When this register is read, “0” is always returned.

TSCRGx is initialized to 00_H after reset.

Overflow Flag Clear Gx Enable (OVFCGx)

When overflow flag Gx (OVFCGx) of timer status register Gx (TSRGx) is set to “1”, writing “1” to this register clears OVFCGx to “0”. When read, “0” is always returned.

Compare Match Flag Gx Enable (CMFCGx)

When compare match flag Gx (CMFCGx) of timer status register Gx (TSRGx) is set to “1”, writing “1” to this register clears CMFCGx to “0”. When read, “0” is always returned.

30.11.2.7 TCNTGx — Timer Counters Gx

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TCNTGx[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNTGx[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_{G-1}: Corresponding to subblocks G0 to GSB_{G-1}

Table 30.204 TCNTGx Register Contents

Bit Position	Bit Name	Function
31 to 0	TCNTGx[31:0]	Timer Counter Gx These bits store the up-counter value.

Note: x = 0 to SB_{G-1}: Corresponding to subblocks G0 to GSB_{G-1}

Timer counters Gx (TCNTGx) are 32-bit readable/writable registers. These registers are provided one for each subblock and are incremented by the clock selected in the corresponding control register. Clock bus 0 to 5 can be selected as the input clock.

When 1 is set as EVSYMGx, become the external event synchronous mode, and an input AGCK timing signal is detected in timer B and count start.

These counter values are compared with the value in compare match register Gx (OCRGx). When they match, the compare match flag (CMFGx bit in TSRGx register) is set and the counter value is cleared to 0000 0000_H in the next PCLK cycle. Note, however, that TCNTGx is initialized to 0000 0001_H if counter clearing by compare match and incrementation occur simultaneously. This occurs when TCNTGx is driven by the clock whose frequency is equal to the PCLK. An interrupt request can be issued to the CPU by detection of a compare match.

TCNTGx is initialized to 0000 0000_H after reset.

30.11.2.8 OCRGx — Compare Match Registers Gx

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OCRGx[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCRGx[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_G-1: Corresponding to subblocks G0 to GSB_G-1

Table 30.205 OCRGx Register Contents

Bit Position	Bit Name	Function
31 to 0	OCRGx[31:0]	Compare match Gx These bits set the compare match value.

Note: x = 0 to SB_G-1: Corresponding to subblocks G0 to GSB_G-1

Compare match registers Gx (OCRGx) are 32-bit readable/writable registers. These registers are provided one for each subblock, and function as the output compare register for timer counter Gx. A compare match can trigger an interrupt request.

Do not set OCRGx to 0000 0000_H. If set, compare match occurs at unintended timing.

OCRGx is initialized to FFFF FFFF_H after reset.

30.11.2.9 RLDGx — Reload Registers Gx

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RLDGx[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RLDGx[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: x = 0 to SB_G-1: Corresponding to subblocks G0 to GSB_G-1

Reload registers Gx (RLDGx) are 32-bit readable/writable registers.

When the reload function is enabled, the value in this register is transferred to the compare match register Gx (OCRGx) upon compare match in the compare register (OCRGx).

Don't set 0000_0000_H as RLDGx. When setting 0000_0000_H, a comparing match occurs by the timing which isn't intended.

RLDGx is initialized to FFFF FFFF_H after reset.

30.11.3 Details of Operation

Specifying a time period in OCRGx generates a positive logical pulse of one PCLK cycle when the specified period has elapsed. The initial value of the output signal is “0”. The counting period is specified based on the clock source of TCNTGx. The generated signals can be used as interrupt request triggers.

The compare match flag (CMFGx) of the timer status register Gx (TSRGx) is set when a compare match occurs. Compare match can be used as a trigger for issuing an interrupt request to the CPU.

Figure 30.94 illustrates the operation example. In this example, the TCNTGx clock is an ideal signal to show the counting and clearing timing of counter TCNTGx.

If EVSYMGx is set to “1”, external event synchronization mode is set, an input AGCK timing signal is detected in timer B and count start.

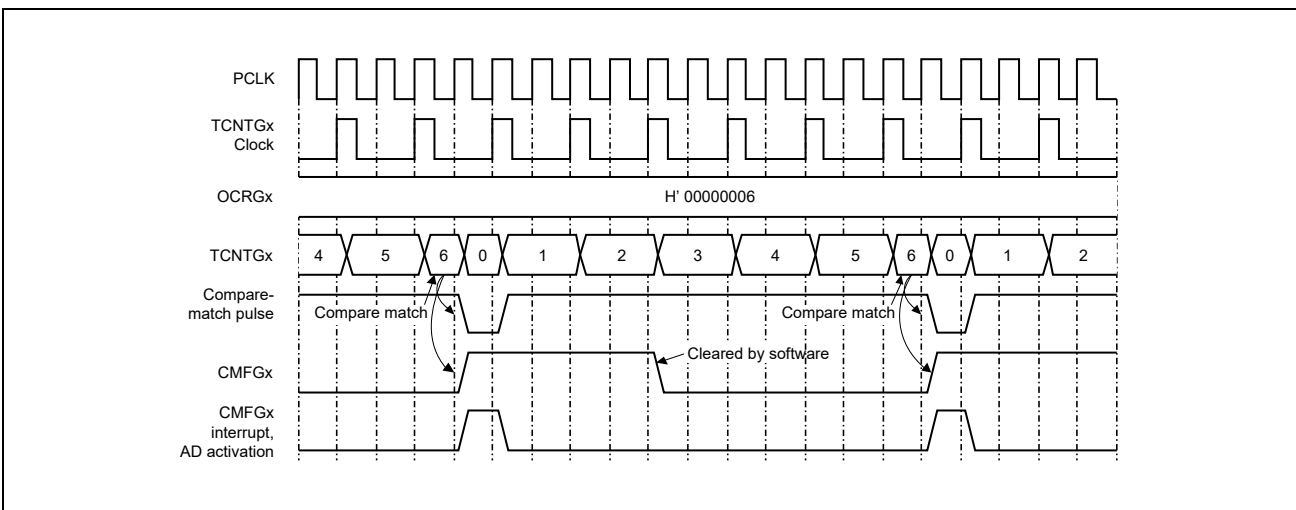


Figure 30.94 Operation Example of Counter and Compare Match

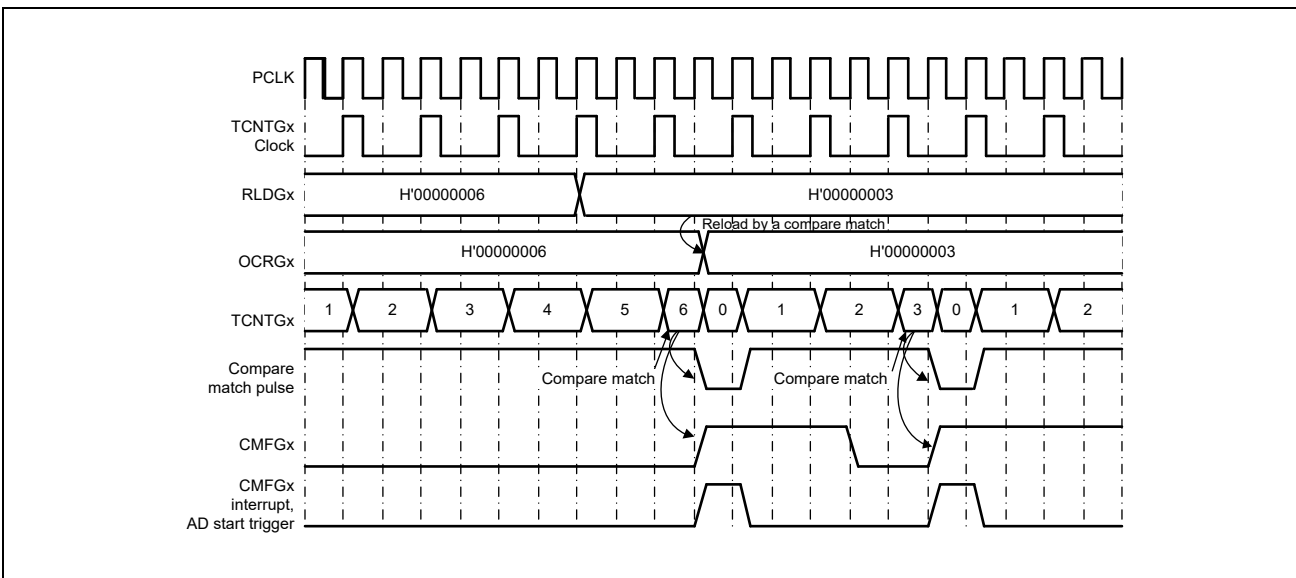


Figure 30.95 Reload Operation (RLDENGx = 1)

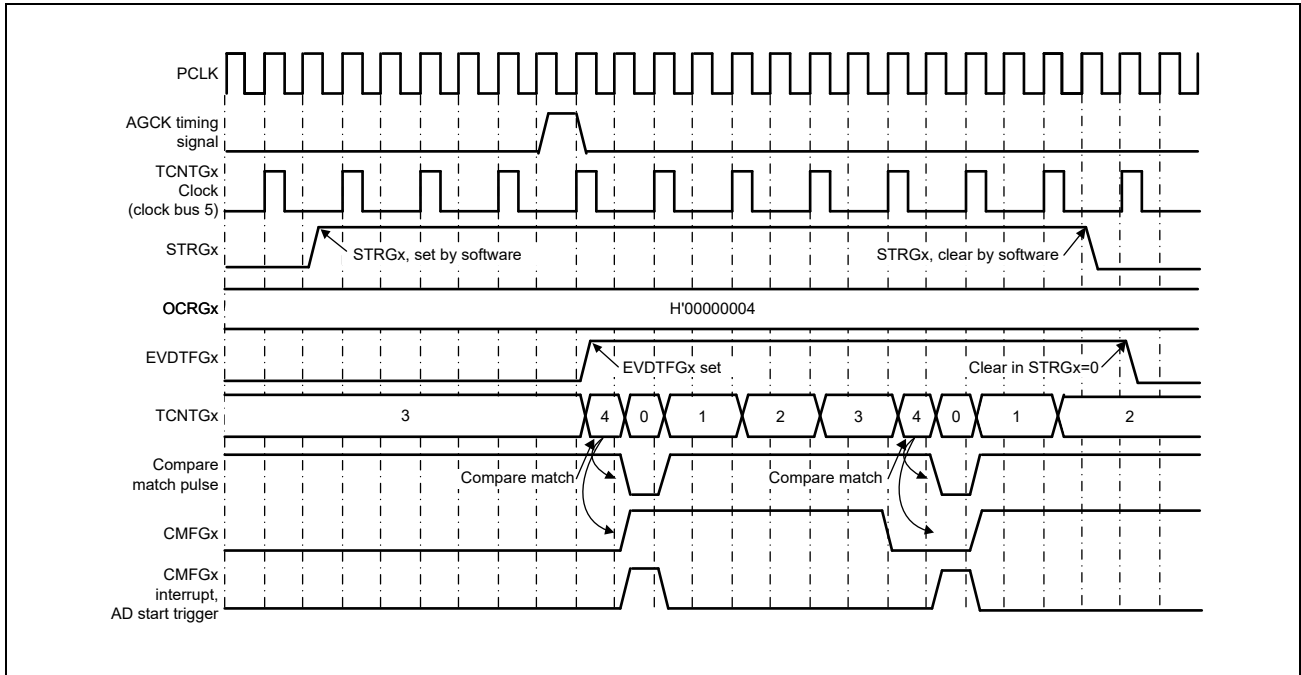


Figure 30.96 External Event Synchronization Mode (EVSIMGx = 1)

30.12 Automatic Switching of DMA and A/D Requests

30.12.1 Operation

This function controls trigger signals to be output to DMA and SAR-AD.

There are three trigger signals of data DMA trigger, count DMA trigger, and SAR-AD trigger signals described below

(1) Data DMA Trigger

DMA output signals AD skipping DMA transfer request 0 and AD skipping DMA transfer request 1 are provided as the data DMA trigger signals.

The data DMA trigger can be selected from AGCK1 from timer B, timer G sub-block G1, G2 compare interrupt output (timer G1 compare interrupt or timer G2 compare interrupt), DF filter completion signal, ADC scan group 0 to 4 end interrupts (ADC0 scan group 0 end interrupt, ADC0 scan group 1 end interrupt, ADC0 scan group 2 end interrupt, ADC0 scan group 3 end interrupt, or ADC0 scan group 4 end interrupt), and output stop.

Either of two data DMA trigger signals is output, and the DMA output to output the trigger is switched at the comparison timing between TCNT1D0 and OCR1D00 of timer D.

(2) Count DMA Trigger

DMA output signals AD skipping DMA transfer request 2 and AD skipping DMA transfer request 3 are provided as the count DMA trigger signals.

The count DMA trigger can be selected from AGCK of timer A, timer G sub-block G3 compare interrupt output (timer G3 compare interrupt), and output stop.

Either of two count DMA trigger signals is output, and the DMA output to output the trigger is switched at the comparison timing between TCNT1D0 and OCR1D00 of timer D.

(3) SAR-AD Trigger

SAR-AD output signal is provided as the SAR-AD trigger signal.

The SAR-AD trigger can be selected from AGCK1 of timer B, timer G sub-block G1, G2 compare interrupt outputs (timer G1 compare interrupt or timer G2 compare interrupt), and output stop.

30.12.2 Registers Related to Automatic Switching of DMA and AD Requests

30.12.2.1 TRGSRDMA0 — Trigger Status Register DMA0

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OUTSRDMA0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 30.206 TRGSRDMA0 Register Contents

Bit Position	Bit Name	Function
7 to 1	—	These bits are not used. Fix these bits to 0.
0	OUTSRDMA0	0: AD skipping DMA transfer request 0 is selected 1: AD skipping DMA transfer request 1 is selected.

Trigger status register DMA0 is an 8-bit read-only register that indicates the selected data DMA output: AD skipping DMA transfer request 0 or AD skipping DMA transfer request 1.

30.12.2.2 TRGSELDMA00 — Trigger Select Register DMA00

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	INSELDMA00			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 30.207 TRGSELDMA00 Register Contents

Bit Position	Bit Name	Function
7 to 4	—	These bits are not used. Fix these bits to 0.
3 to 0	INSELDMA00	0000: Output stop (output is fixed to “0”.) 0001: AGCK1 is selected. 0010: Timer G subblock G1 compare interrupt signal (CMFG1 interrupt) is selected. 0011: Timer G subblock G2 compare interrupt signal (CMFG2 interrupt) is selected. 0100: DF filter completion signal is selected. 0101: Setting prohibited 0110: Setting prohibited 0111: Setting prohibited 1000: ADC0 scan group 0 end interrupt is selected. 1001: ADC0 scan group 1 end interrupt is selected. 1010: ADC0 scan group 2 end interrupt is selected. 1011: ADC0 scan group 3 end interrupt is selected. 1100: ADC0 scan group 4 end interrupt is selected. 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited

This register is an 8-bit readable/writable register that selects the signal to be output to the data DMA output AD skipping DMA transfer request 0. This register can be rewritten during operation. When this register is rewritten, the changed setting becomes valid immediately.

30.12.2.3 TRGSELDMA01 — Trigger Select Register DMA01

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	INSELDMA01			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 30.208 TRGSELDMA01 Register Contents

Bit Position	Bit Name	Function
7 to 4	—	These bits are not used. Fix these bits to 0.
3 to 0	INSELDMA01	0000: Output stop (output is fixed to “0”.) 0001: AGCK1 is selected. 0010: Timer G subblock G1 compare interrupt signal (CMFG1 interrupt) is selected. 0011: Timer G subblock G2 compare interrupt signal (CMFG2 interrupt) is selected. 0100: DF filter completion signal is selected. 0101: Setting prohibited 0110: Setting prohibited 0111: Setting prohibited 1000: ADC0 scan group 0 end interrupt is selected. 1001: ADC0 scan group 1 end interrupt is selected. 1010: ADC0 scan group 2 end interrupt is selected. 1011: ADC0 scan group 3 end interrupt is selected. 1100: ADC0 scan group 4 end interrupt is selected. 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited

This register is an 8-bit readable/writable register that selects the signal to be output to the data DMA output AD skipping DMA transfer request 1. This register can be rewritten during operation. When this register is rewritten, the changed setting becomes valid immediately.

30.12.2.4 TRGSELAD — Trigger Select Register AD

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	INSELAD	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.209 TRGSELAD Register Contents

Bit Position	Bit Name	Function
7 to 2	—	These bits are not used. Fix these bits to 0.
1, 0	INSELAD	00: Output stop (output is fixed to "0".) 01: AGCK1 is selected. 10: Timer G subblock G1 compare interrupt signal (CMFG1 interrupt) is selected. 11: Timer G subblock G2 compare interrupt signal (CMFG2 interrupt) is selected.

This register is an 8-bit readable/writable register that selects the signal to be output to the SAR-AD trigger. This register can be rewritten during operation. When this register is rewritten, the changed setting becomes valid immediately.

30.12.2.5 TRGSRDMA1 — Trigger Status Register DMA1

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OUTSRDMA1
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 30.210 TRGSRDMA1 Register Contents

Bit Position	Bit Name	Function
7 to 1	—	These bits are not used. Fix these bits to 0.
0	OUTSRDMA1	0: AD skipping DMA transfer request 2 is selected 1: AD skipping DMA transfer request 3 is selected

This register is an 8-bit read-only register that indicates the selected count DMA output: AD skipping DMA transfer request 2 or AD skipping DMA transfer request 3.

30.12.2.6 TRGSELDMA10 — Trigger Select Register DMA10

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	INSELDMA10	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.211 TRGSELDMA10 Register Contents

Bit Position	Bit Name	Function
7 to 2	—	These bits are not used. Fix these bits to 0.
1, 0	INSELDMA10	00: Output stop (output is fixed to "0".) 01: AGCK is selected 10: Timer G subblock G3 compare interrupt signal (CMFG3 interrupt) is selected. 11: Setting prohibited

This register is an 8-bit readable/writable register that selects the signal to be output to the count DMA output AD skipping DMA transfer request 2. This register can be rewritten during operation. When this register is rewritten, the changed setting becomes valid immediately.

30.12.2.7 TRGSELDMA11 — Trigger Select Register DMA11

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	INSELDMA11	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.212 TRGSELDMA11 Register Contents

Bit Position	Bit Name	Function
7 to 2	—	These bits are not used. Fix these bits to 0.
1, 0	INSELDMA11	00: Output stop (output is fixed to "0".) 01: AGCK is selected 10: Timer G subblock G3 compare interrupt signal (CMFG3 interrupt) is selected. 11: Setting prohibited

This register is an 8-bit readable/writable register that selects the signal to be output to the count DMA output AD skipping DMA transfer request 3. This register can be rewritten during operation. When this register is rewritten, the changed setting becomes valid immediately.

30.12.3 Details of Operation

The following describes the operation and block diagram of each trigger.

30.12.3.1 Data DMA Trigger

The signal to be output to the data DMA trigger output pin (AD skipping DMA transfer request 0) is selected by the INSELDMA00[3:0] bits in TRGSELDMA00.

The signal to be output to the data DMA trigger output pin (AD skipping DMA transfer request 1) is selected by the INSELDMA01[3:0] bits in TRGSELDMA00.

The data DMA trigger output pin outputs only AD skipping DMA transfer request 0 when the DMA select signal is low, and only AD skipping DMA transfer request 1 when the DMA select signal is high.

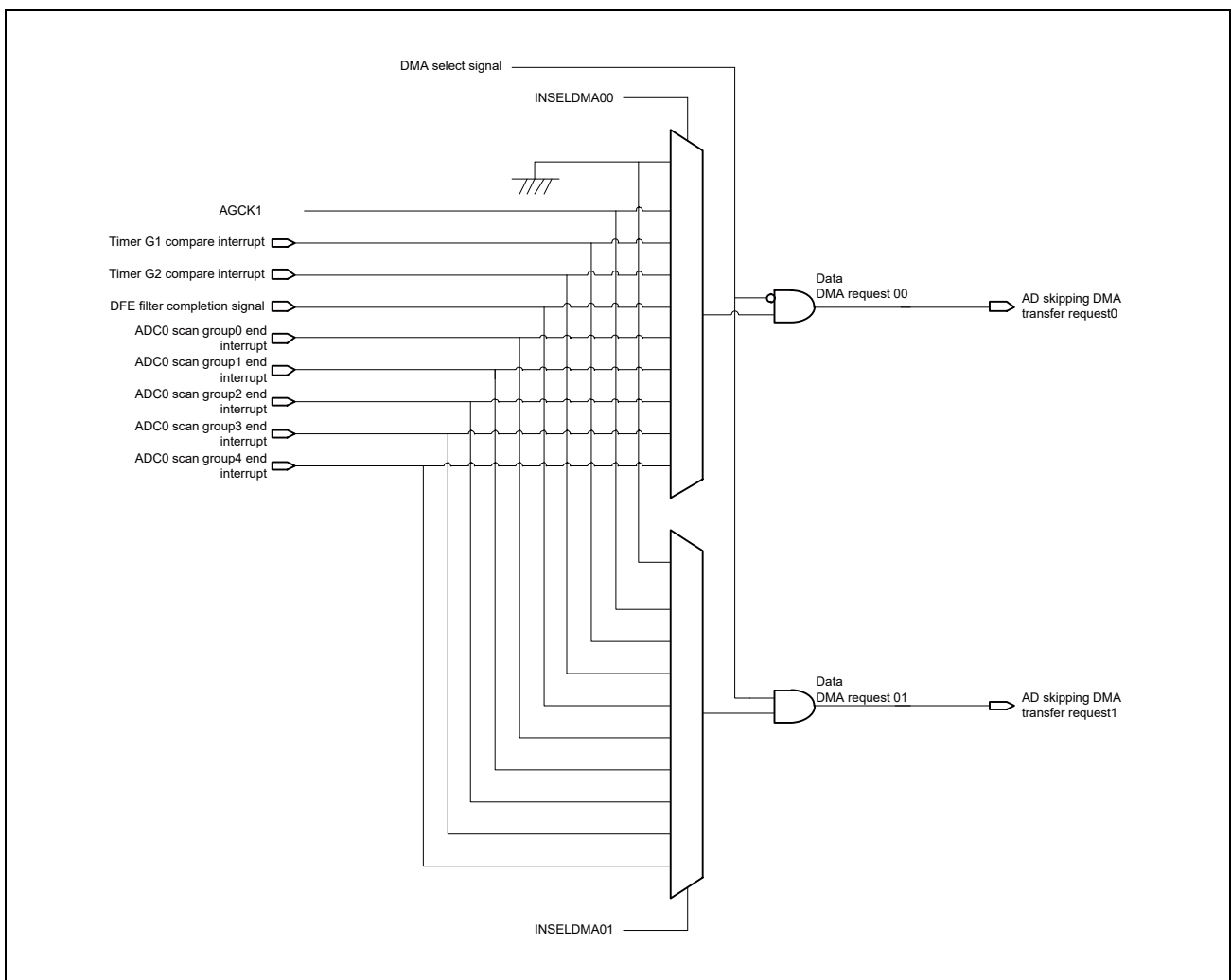


Figure 30.97 Data DMA Trigger Selection Circuit

30.12.3.2 Count DMA Trigger

The signal to be output to the count DMA trigger output pin (AD skipping DMA transfer request 2) is selected by the INSELDMA10[1:0] bits in TRGSELDMA10.

The signal to be output to the data DMA trigger output pin (AD skipping DMA transfer request 3) is selected by the INSELDMA11[1:0] bits in TRGSELDMA11.

The count DMA trigger output pin outputs only AD skipping DMA transfer request 2 when the DMA select signal is low, and only AD skipping DMA transfer request 3 when the DMA select signal is high.

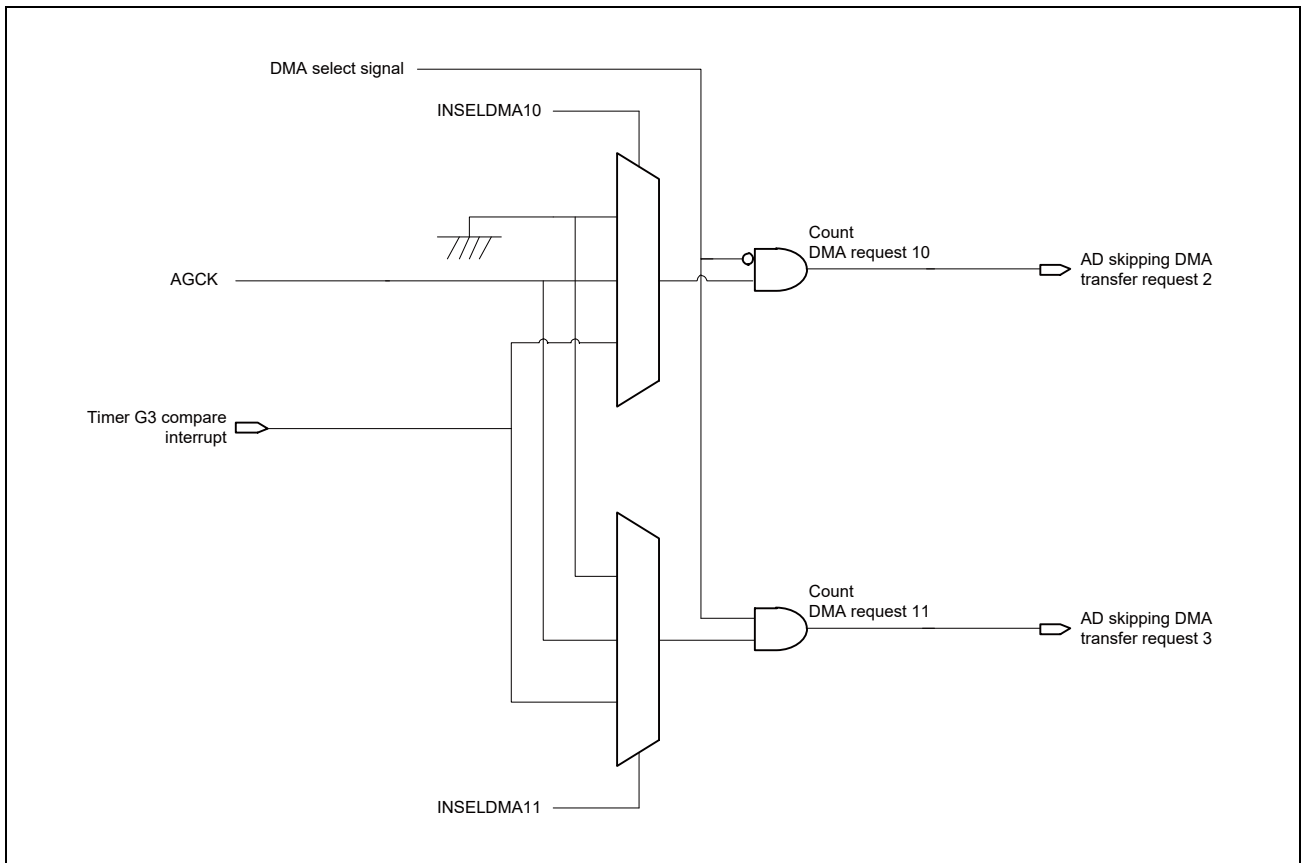


Figure 30.98 Count DMA Trigger Selection Circuit

30.12.3.3 SAR-AD Trigger

The signal to be output to the SAR-AD trigger output pin is selected by the INSELAD[1:0] bits in TRGSELAD.

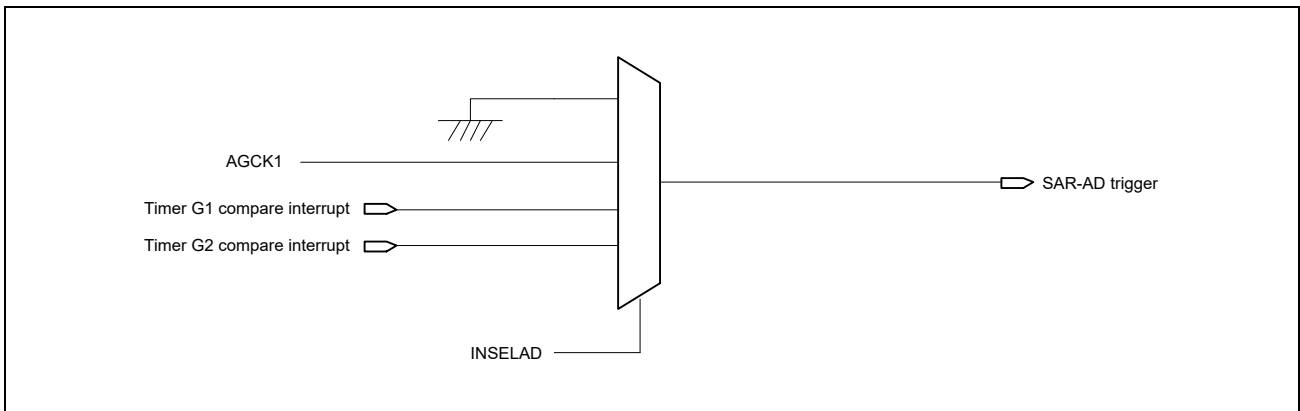


Figure 30.99 SAR-AD Trigger Selection Circuit

Figure 30.100 shows the trigger output timing chart.

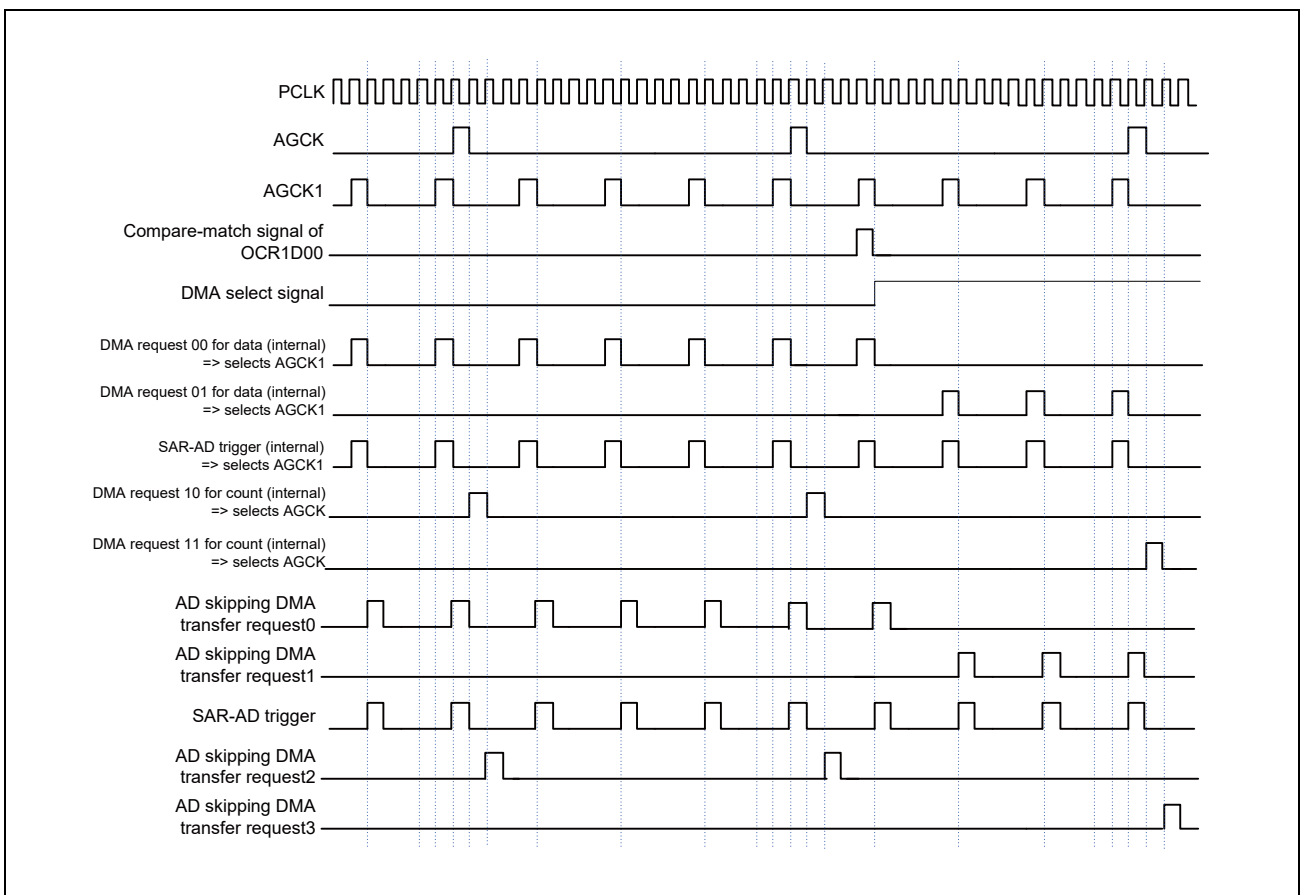


Figure 30.100 DMA/AD Trigger Timing Chart

30.13 Trigger Select Function

30.13.1 Overview of Operation

The function for which each trigger indicated on the following is selected.

- Interrupt signal communicated to INTC is selected.
- Interrupt signal communicated to sDMA/DTS is selected.
- “Timer DFE Capture trigger i ($i = 0$ to 2)” communicated to DFE selected.
- “Timer FIFO Capture trigger” communicated to DFE is selected.
- “read gate” communicated to DSADC is selected.
- “read gate” communicated to C-ADC is selected.
- “time stamp” and “sync pulse” communicated to PSI5S is selected.

30.13.2 Registers Related to Trigger Select Function

30.13.2.1 ATUINTELA — ATU TimerA Interrupt Selection Control Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATU_INTSEL_A3[2:0]			—	ATU_INTSEL_A2[2:0]			—	ATU_INTSEL_A1[2:0]			—	ATU_INTSEL_A0[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.213 ATUINTELA Register Contents

Bit Position	Bit Name	Function
31 to 15 11, 7, 3	—	Reserved. When writing to these bits, write 0.
14 to 12 ($j=3$) 10 to 8 ($j=2$) 6 to 4 ($j=1$) 2 to 0 ($j=0$)	ATU_INTSEL_Aj[2:0]	Select as interrupt factor no. $10+j$ from the following signals. ($j=3$ to 0) 000: ICRA0 input capture interrupt 001: ICRA1 input capture interrupt 010: ICRA2 input capture interrupt 011: ICRA3 input capture interrupt 100: ICRA4 input capture interrupt 101: ICRA5 input capture interrupt 110: ICRA6 input capture interrupt 111: ICRA7 input capture interrupt

The factor communicated to a CPU system is selected from the interrupt signal output from Timer A.

30.13.2.2 ATUINSEL_D0 — ATU TimerD Interrupt Selection Control Register 0

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ATU_INTSEL_D13[2:0]			—	ATU_INTSEL_D12[2:0]			—	ATU_INTSEL_D11[2:0]			—	ATU_INTSEL_D10[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATU_INTSEL_D03[2:0]			—	ATU_INTSEL_D02[2:0]			—	ATU_INTSEL_D01[2:0]			—	ATU_INTSEL_D00[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.214 ATUINSEL_D0 Register Contents

Bit Position	Bit Name	Function
31, 27, 23, 19, 15, 11, 7, 3	—	Reserved. When writing to these bits, write 0.
30 to 28 (xy = 13)	ATU_INTSEL_Dxy [2:0]	Select as interrupt request no.(140+4*x+y) from the following signals. 000: OCR1Dxy compare match interrupt
26 to 24 (xy = 12)		001: OCR2Dxy compare match interrupt
22 to 20 (xy = 11)		010: 1shot pulse ON Dxy occurrence interrupt
18 to 16 (xy = 10)		011: 1shot pulse OFF Dxy occurrence interrupt
14 to 12 (xy = 03)		100: UDIDxy down-counter underflow interrupt
10 to 8 (xy = 02)		101: All interrupts combined by OR
6 to 4 (xy = 01)		Others: inhibit:
2 to 0 (xy = 00)		

Note: x: Number of sub-block, y: Number of channel

The factor communicated to a CPU system is selected from the interrupt signal output from Timer D.

30.13.2.3 ATUINTELD1 — ATU TimerD Interrupt Selection Control Register 1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ATU_INTSEL_D33[2:0]			—	ATU_INTSEL_D32[2:0]			—	ATU_INTSEL_D31[2:0]			—	ATU_INTSEL_D30[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATU_INTSEL_D23[2:0]			—	ATU_INTSEL_D22[2:0]			—	ATU_INTSEL_D21[2:0]			—	ATU_INTSEL_D20[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.215 ATUINTELD1 Register Contents

Bit Position	Bit Name	Function
31, 27, 23, 19, 15, 11, 7, 3	—	Reserved. When writing to these bits, write 0.
30 to 28 (xy = 33)	ATU_INTSEL_Dxy [2:0]	Select as interrupt request no.(140+4*x+y) from the following signals.
26 to 24 (xy = 32)		000: OCR1Dxy compare match interrupt
22 to 20 (xy = 31)		001: OCR2Dxy compare match interrupt
18 to 16 (xy = 30)		010: 1shot pulse ON Dxy occurrence interrupt
14 to 12 (xy = 23)		011: 1shot pulse OFF Dxy occurrence interrupt
10 to 8 (xy = 22)		100: UDIDxy down-counter underflow interrupt
6 to 4 (xy = 21)		101: All interrupts combined by OR
2 to 0 (xy = 20)		Others inhibit:

Note: x: Number of sub-block, y: Number of channel

The factor communicated to a CPU system is selected from the interrupt signal output from Timer D.

30.13.2.4 ATUINTELD2 — ATU TimerD Interrupt Selection Control Register 2

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ATU_INTSEL_D53[2:0]			—	ATU_INTSEL_D52[2:0]			—	ATU_INTSEL_D51[2:0]			—	ATU_INTSEL_D50[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATU_INTSEL_D43[2:0]			—	ATU_INTSEL_D42[2:0]			—	ATU_INTSEL_D41[2:0]			—	ATU_INTSEL_D40[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.216 ATUINTELD2 Register Contents

Bit Position	Bit Name	Function
31, 27, 23, 19, 15, 11, 7, 3	—	Reserved. When writing to these bits, write 0.
30 to 28 (xy = 53)	ATU_INTSEL_Dxy [2:0]	Select as interrupt request no.(140+4*x+y) from the following signals. 000: OCR1Dxy compare match interrupt 001: OCR2Dxy compare match interrupt 010: 1shot pulse ON Dxy occurrence interrupt 011: 1shot pulse OFF Dxy occurrence interrupt 100: UDIDxy down-counter underflow interrupt 101: All interrupts combined by OR Others inhibit:
26 to 24 (xy = 52)		
22 to 20 (xy = 51)		
18 to 16 (xy = 50)		
14 to 12 (xy = 43)		
10 to 8 (xy = 42)		
6 to 4 (xy = 41)		
2 to 0 (xy = 40)		

Note: x: Number of sub-block, y: Number of channel

The factor communicated to a CPU system is selected from the interrupt signal output from Timer D.

30.13.2.5 ATUINTELD3 — ATU TimerD Interrupt Selection Control Register 3

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ATU_INTSEL_D73[2:0]			—	ATU_INTSEL_D72[2:0]			—	ATU_INTSEL_D71[2:0]			—	ATU_INTSEL_D70[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATU_INTSEL_D63[2:0]			—	ATU_INTSEL_D62[2:0]			—	ATU_INTSEL_D61[2:0]			—	ATU_INTSEL_D60[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.217 ATUINTELD3 Register Contents

Bit Position	Bit Name	Function
31, 27, 23, 19, 15, 11, 7, 3	—	Reserved. When writing to these bits, write 0.
30 to 28 (xy = 73)	ATU_INTSEL_Dxy [2:0]	Select as interrupt request no.(140+4*x+y) from the following signals.
26 to 24 (xy = 72)		000: OCR1Dxy compare match interrupt
22 to 20 (xy = 71)		001: OCR2Dxy compare match interrupt
18 to 16 (xy = 70)		010: 1shot pulse ON Dxy occurrence interrupt
14 to 12 (xy = 63)		011: 1shot pulse OFF Dxy occurrence interrupt
10 to 8 (xy = 62)		100: UDIDxy down-counter underflow interrupt
6 to 4 (xy = 61)		101: All interrupts combined by OR
2 to 0 (xy = 60)		Others inhibit:

Note: x: Number of sub-block, y: Number of channel

The factor communicated to a CPU system is selected from the interrupt signal output from Timer D.

30.13.2.6 ATUINSEL4 — ATU TimerD Interrupt Selection Control Register 4

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATU_INTSEL_D83[2:0]		—	ATU_INTSEL_D82[2:0]		—	ATU_INTSEL_D81[2:0]		—	ATU_INTSEL_D80[2:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.218 ATUINSEL4 Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved. When writing to these bits, write 0.
15, 11, 7, 3		
14 to 12 (xy = 83)	ATU_INTSEL_Dxy [2:0]	Select as interrupt request no.(140+4*x+y) from the following signals.
10 to 8 (xy = 82)		000: OCR1Dxy compare match interrupt
6 to 4 (xy = 81)		001: OCR2Dxy compare match interrupt
2 to 0 (xy = 80)		010: 1shot pulse ON Dxy occurrence interrupt
		011: 1shot pulse OFF Dxy occurrence interrupt
		100: UDIDxy down-counter underflow interrupt
		101: All interrupts combined by OR
		Others inhibit:

Note: x: Number of sub-block, y: Number of channel

The factor communicated to a CPU system is selected from the interrupt signal output from Timer D.

30.13.2.7 ATUDMASELB — ATU TimerB DMA Selection Control Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ATU_DMASEL_B3[3:0]				ATU_DMASEL_B2[3:0]				ATU_DMASEL_B1[3:0]				ATU_DMASEL_B0[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.219 ATUDMASELB Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved. When writing to these bits, write 0.
15 to 12 (j=3)	ATU_DMASEL_Bj	Select as DMA request 46+j or 174+j (group 1) or DTS request no.46+j or 110+j (group 2) from the following signals. (j=3 to 0)
11 to 8 (j=2)	[3:0]	
7 to 4 (j = 1)	(j=3 to 0)	
3 to 0 (j = 0)		
		0000: OCRB0 compare match interrupt
		0001: OCRB1 compare match interrupt
		0010: OCRB6 compare match interrupt
		0011: OCRB10 compare match interrupt
		0100: OCRB11 compare match interrupt
		0101: OCRB12 compare match interrupt
		0110: Comparison between TCNTB6M and ICRB6 condition match interrupt
		0111: AND/OR condition of CMFB6 and CMFB6M condition match interrupt
		1000: ICRB0 input capture interrupt
		others: inhibited

The factor communicated to a DMA system is selected from the interrupt signal output from Timer B.

30.13.2.8 ATUDMASELCD0 — ATU TimerC/TimerD DMA Selection Control Register 0

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ATU_DMASEL_CD07 [2:0]			—	ATU_DMASEL_CD06 [2:0]			—	ATU_DMASEL_CD05 [2:0]			—	ATU_DMASEL_CD04 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATU_DMASEL_CD03 [2:0]			—	ATU_DMASEL_CD02 [2:0]			—	ATU_DMASEL_CD01 [2:0]			—	ATU_DMASEL_CD00 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.220 ATUDMASELCD0 Register Contents (1/3)

Bit Position	Bit Name	Function
31	—	Reserved. When writing to these bits, write 0.
30 to 28	ATU_DMASEL_CD07 [2:0]	Select as DMA request no.71 or 199 (group 1), DTS request no.7 or 71 (group 3) from the following signals. 000: GRC13 input capture/compare match interrupt 001: OCRC13 input capture/compare match interrupt others: inhibit
27	—	Reserved. When writing to these bits, write 0.
26 to 24	ATU_DMASEL_CD06 [2:0]	Select as DMA request no.70 or 198 (group 1), DTS request no.6 or 70 (group 3) from the following signals. 000: GRC12 input capture/compare match interrupt 001: OCRC12 input capture/compare match interrupt others: inhibit
23	—	Reserved. When writing to these bits, write 0.
22 to 20	ATU_DMASEL_CD05 [2:0]	Select as DMA request no.69 or 197 (group 1), DTS request no.5 or 69 (group 3) from the following signals. 000: GRC11 input capture/compare match interrupt 001: OCRC11 input capture/compare match interrupt others: inhibit

Table 30.220 ATUDMASELCD0 Register Contents (2/3)

Bit Position	Bit Name	Function
19	—	Reserved. When writing to these bits, write 0.
18 to 16	ATU_DMASEL_CD04 [2:0]	Select as DMA request no.68 or 196 (group 1), DTS request no.4 or 68 (group 3) from the following signals. 000: GRC10 input capture/compare match interrupt 001: OCRC10 input capture/compare match interrupt others: inhibit
15	—	Reserved. When writing to these bits, write 0.
14 to 12	ATU_DMASEL_CD03 [2:0]	Select as DMA request no.67 or 195 (group 1) DTS request no.3 or 67 (group 3) from the following signals. 000: GRC03 input capture/compare match interrupt 001: OCRC03 input capture/compare match interrupt 010: OCR1D83 compare match interrupt 011: OCR2D83 compare match interrupt 100: 1shot pulse ON D83 occurrence interrupt 101: 1shot pulse OFF D83 occurrence interrupt 110: UDID83 down-counter underflow interrupt others: inhibit
11	—	Reserved. When writing to these bits, write 0.
10 to 8	ATU_DMASEL_CD02 [2:0]	Select as DMA request no.66 or 194 (group 1), DTS request no.2 or 66(group 3) from the following signals. 000: GRC02 input capture/compare match interrupt 001: OCRC02 input capture/compare match interrupt 010: OCR1D82 compare match interrupt 011: OCR2D82 compare match interrupt 100: 1shot pulse ON D82 occurrence interrupt 101: 1shot pulse OFF D82 occurrence interrupt 110: UDID82 down-counter underflow interrupt others: inhibit
7	—	Reserved. When writing to these bits, write 0.
6 to 4	ATU_DMASEL_CD01 [2:0]	Select as DMA request no.65 or 193 (group 1), DTS request no.1 or 65 (group 3) from the following signals. 000: GRC01 input capture/compare match interrupt 001: OCRC01 input capture/compare match interrupt 010: OCR1D81 compare match interrupt 011: OCR2D81 compare match interrupt 100: 1shot pulse ON D81 occurrence interrupt 101: 1shot pulse OFF D81 occurrence interrupt 110: UDID81 down-counter underflow interrupt others: inhibit

Table 30.220 ATUDMASELCD0 Register Contents (3/3)

Bit Position	Bit Name	Function
3	—	Reserved. When writing to these bits, write 0.
2 to 0	ATU_DMASEL_CD00 [2:0]	Select as DMA request no.64 or 192 (group 1), DTS request no.0 or 64 (group 3) from the following signals. 000: GRC00 input capture/compare match interrupt 001: OCRC00 input capture/compare match interrupt 010: OCR1D80 compare match interrupt 011: OCR2D80 compare match interrupt 100: 1shot pulse ON D80 occurrence interrupt 101: 1shot pulse OFF D80 occurrence interrupt 110: UDID80 down-counter underflow interrupt others: inhibit

The factor communicated to a DMA system is selected from the interrupt signal output from Timer C/D.

30.13.2.9 ATUDMASELCD1 — ATU TimerC/TimerD DMA Selection Control Register 1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ATU_DMASEL_CD15 [2:0]			—	ATU_DMASEL_CD14 [2:0]			—	ATU_DMASEL_CD13 [2:0]			—	ATU_DMASEL_CD12 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATU_DMASEL_CD11 [2:0]			—	ATU_DMASEL_CD10 [2:0]			—	ATU_DMASEL_CD09 [2:0]			—	ATU_DMASEL_CD08 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.221 ATUDMASELCD1 Register Contents (1/3)

Bit Position	Bit Name	Function
31	—	Reserved. When writing to these bits, write 0.
30 to 28	ATU_DMASEL_CD15 [2:0]	Select as DMA request no.79 or 207 (group 1). DTS request no.15 or 79 (group 3) from the following signals. 000: GRC33 input capture/compare match interrupt 001: OCRC33 input capture/compare match interrupt others: inhibit
27	—	Reserved. When writing to these bits, write 0.
26 to 24	ATU_DMASEL_CD14 [2:0]	Select as DMA request no.78 or 206 (group 1), DTS request no.14 or 78 (group 3) from the following signals. 000: GRC32 input capture/compare match interrupt 001: OCRC32 input capture/compare match interrupt others: inhibit
23	—	Reserved. When writing to these bits, write 0.
22 to 20	ATU_DMASEL_CD13 [2:0]	Select as DMA request no.77 or 205 (group 1), DTS request no.13 or 77(group 3) from the following signals. 000: GRC31 input capture/compare match interrupt 001: OCRC31 input capture/compare match interrupt others: inhibit

Table 30.221 ATUDMASELCD1 Register Contents (2/3)

Bit Position	Bit Name	Function
19	—	Reserved. When writing to these bits, write 0.
18 to 16	ATU_DMASEL_CD12 [2:0]	Select as DMA request no.76 or 204 (group 1), DTS request no.12or 76 (group 3) from the following signals. 000: GRC30 input capture/compare match interrupt 001: OCRC30 input capture/compare match interrupt others: inhibit
15	—	Reserved. When writing to these bits, write 0.
14 to 12	ATU_DMASEL_CD11 [2:0]	Select as DMA request no.75 or 203 (group 1), DTS request no.11or 75(group 3) from the following signals. 000: GRC23 input capture/compare match interrupt 001: OCRC23 input capture/compare match interrupt others: inhibit
11	—	Reserved. When writing to these bits, write 0.
10 to 8	ATU_DMASEL_CD10 [2:0]	Select as DMA request no.74 or 202 (group 1), DTS request no.10 or 74 (group 3) from the following signals. 000: GRC22 input capture/compare match interrupt 001: OCRC22 input capture/compare match interrupt others: inhibit
7	—	Reserved. When writing to these bits, write 0.
6 to 4	ATU_DMASEL_CD09 [2:0]	Select as DMA request no.73 or 201 (group 1), DTS request no.9 or 73(group 3) from the following signals. 000: GRC21 input capture/compare match interrupt 001: OCRC21 input capture/compare match interrupt others: inhibit

Table 30.221 ATUDMASELCD1 Register Contents (3/3)

Bit Position	Bit Name	Function
3	—	Reserved. When writing to these bits, write 0.
2 to 0	ATU_DMASEL_CD08 [2:0]	Select as DMA request no.72 or 200 (group 1), DTS request no.8 or 72 (group 3) from the following signals. 000: GRC20 input capture/compare match interrupt 001: OCRC20 input capture/compare match interrupt others: inhibit

The factor communicated to a DMA system is selected from the interrupt signal output from Timer C/D.

30.13.2.10 ATUDMASELCD2 — ATU TimerC/TimerD DMA Selection Control Register 2

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ATU_DMASEL_CD23 [2:0]			—	ATU_DMASEL_CD22 [2:0]			—	ATU_DMASEL_CD21 [2:0]			—	ATU_DMASEL_CD20 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATU_DMASEL_CD19 [2:0]			—	ATU_DMASEL_CD18 [2:0]			—	ATU_DMASEL_CD17 [2:0]			—	ATU_DMASEL_CD16 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.222 ATUDMASELCD2 Register Contents (1/3)

Bit Position	Bit Name	Function
31	—	Reserved. When writing to these bits, write 0.
30 to 28	ATU_DMASEL_CD23 [2:0]	Select as DMA request no.87 or 215 (group 1), DTS request no.23 or 87 (group 3) from the following signals. 000: GRC53 input capture/compare match interrupt 001: OCRC53 input capture/compare match interrupt others: inhibit
27	—	Reserved. When writing to these bits, write 0.
26 to 24	ATU_DMASEL_CD22 [2:0]	Select as DMA request no.86 or 214 (group 1), DTS request no.22 or 86 (group 3) from the following signals. 000: GRC52 input capture/compare match interrupt 001: OCRC52 input capture/compare match interrupt others: inhibit
23	—	Reserved. When writing to these bits, write 0.
22 to 20	ATU_DMASEL_CD21 [2:0]	Select as DMA request no.85 or 213 (group 1), DTS request no.21 or 85 (group 3) from the following signals. 000: GRC51 input capture/compare match interrupt 001: OCRC51 input capture/compare match interrupt others: inhibit

Table 30.222 ATUDMASELCD2 Register Contents (2/3)

Bit Position	Bit Name	Function
19	—	Reserved. When writing to these bits, write 0.
18 to 16	ATU_DMASEL_CD20 [2:0]	Select as DMA request no.84 or 212 (group 1), DTS request no.20 or 84 (group 3) from the following signals. 000: GRC50 input capture/compare match interrupt 001: OCRC50 input capture/compare match interrupt others: inhibit
15	—	Reserved. When writing to these bits, write 0.
14 to 12	ATU_DMASEL_CD19 [2:0]	Select as DMA request no.83 or 211 (group 1), DTS request no.19 or 83 (group 3) from the following signals. 000: GRC43 input capture/compare match interrupt 001: OCRC43 input capture/compare match interrupt others: inhibit
11	—	Reserved. When writing to these bits, write 0.
10 to 8	ATU_DMASEL_CD18 [2:0]	Select as DMA request no.82 or 210 (group 1), DTS request no.18 or 82 (group 3) from the following signals. 000: GRC42 input capture/compare match interrupt 001: OCRC42 input capture/compare match interrupt others: inhibit
7	—	Reserved. When writing to these bits, write 0.
6 to 4	ATU_DMASEL_CD17 [2:0]	Select as DMA request no.81 or 209(group 1), DTS request no.17 or 81 (group 3) from the following signals. 000: GRC41 input capture/compare match interrupt 001: OCRC41 input capture/compare match interrupt others: inhibit

Table 30.222 ATUDMASELCD2 Register Contents (3/3)

Bit Position	Bit Name	Function
3	—	Reserved. When writing to these bits, write 0.
2 to 0	ATU_DMASEL_CD16 [2:0]	Select as DMA request no.80 or 208 (group 1), DTS request no.16 or 80 (group 3) from the following signals. 000: GRC40 input capture/compare match interrupt 001: OCRC40 input capture/compare match interrupt others: inhibit

The factor communicated to a DMA system is selected from the interrupt signal output from Timer C/D.

30.13.2.11 ATUDMASELCD3 — ATU TimerC/TimerD DMA Selection Control Register 3

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ATU_DMASEL_CD31 [2:0]			—	ATU_DMASEL_CD30 [2:0]			—	ATU_DMASEL_CD29 [2:0]			—	ATU_DMASEL_CD28 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATU_DMASEL_CD27 [2:0]			—	ATU_DMASEL_CD26 [2:0]			—	ATU_DMASEL_CD25 [2:0]			—	ATU_DMASEL_CD24 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.223 ATUDMASELCD3 Register Contents (1/2)

Bit Position	Bit Name	Function
31	—	Reserved. When writing to these bits, write 0.
30 to 28	ATU_DMASEL_CD31 [2:0]	Select as DMA request no.95 or 223 (group 1), DTS request no.31or 95 (group 3) from the following signals. 000: GRC73 input capture/compare match interrupt 001: OCRC73 input capture/compare match interrupt others: inhibit
27	—	Reserved. When writing to these bits, write 0.
26 to 24	ATU_DMASEL_CD30 [2:0]	Select as DMA request no.94 or 222 (group 1), DTS request no.30 or 94 (group 3) from the following signals. 000: GRC72 input capture/compare match interrupt 001: OCRC72 input capture/compare match interrupt others: inhibit
23	—	Reserved. When writing to these bits, write 0.
22 to 20	ATU_DMASEL_CD29 [2:0]	Select as DMA request no.93 or 221 (group 1), DTS request no.29 or 93 (group 3) from the following signals. 000: GRC71 input capture/compare match interrupt 001: OCRC71 input capture/compare match interrupt others: inhibit
19	—	Reserved. When writing to these bits, write 0.
18 to 16	ATU_DMASEL_CD28 [2:0]	Select as DMA request no.92 or 220 (group 1), DTS request no.28 or 92 (group 3) from the following signals. 000: GRC70 input capture/compare match interrupt 001: OCRC70 input capture/compare match interrupt others: inhibit
15	—	Reserved. When writing to these bits, write 0.

Table 30.223 ATUDMASELCD3 Register Contents (2/2)

Bit Position	Bit Name	Function
14 to 12	ATU_DMASEL_CD27 [2:0]	Select as DMA request no.91 or 219 (group 1), DTS request no.27 or 91 (group 3) from the following signals. 000: GRC63 input capture/compare match interrupt 001: OCRC63 input capture/compare match interrupt others: inhibit
11	—	Reserved. When writing to these bits, write 0.
10 to 8	ATU_DMASEL_CD26 [2:0]	Select as DMA request no.90 or 218 (group 1), DTS request no.26 or 90 (group 3) from the following signals. 000: GRC62 input capture/compare match interrupt 001: OCRC62 input capture/compare match interrupt others: inhibit
7	—	Reserved. When writing to these bits, write 0.
6 to 4	ATU_DMASEL_CD25 [2:0]	Select as DMA request no.89 or 217 (group 1), DTS request no.25 or 89 (group 3) from the following signals. 000: GRC61 input capture/compare match interrupt 001: OCRC61 input capture/compare match interrupt others: inhibit
3	—	Reserved. When writing to these bits, write 0.
2 to 0	ATU_DMASEL_CD24 [2:0]	Select as DMA request no.88 or 216 (group 1), DTS request no.24 or 88 (group 3) from the following signals. 000: GRC60 input capture/compare match interrupt 001: OCRC60 input capture/compare match interrupt others: inhibit

The factor communicated to a DMA system is selected from the interrupt signal output from Timer C/D.

30.13.2.12 ATUDMASELCD4 — ATU TimerC/TimerD DMA Selection Control Register 4

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ATU_DMASEL_CD39 [2:0]			—	ATU_DMASEL_CD38 [2:0]			—	ATU_DMASEL_CD37 [2:0]			—	ATU_DMASEL_CD36 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATU_DMASEL_CD35 [2:0]			—	ATU_DMASEL_CD34 [2:0]			—	ATU_DMASEL_CD33 [2:0]			—	ATU_DMASEL_CD32 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.224 ATUDMASELCD4 Register Contents (1/3)

Bit Position	Bit Name	Function
31	—	Reserved. When writing to these bits, write 0.
30 to 28	ATU_DMASEL_CD39 [2:0]	Select as DMA request no.103 or 231 (group 1), DTS request no.39 or 103 (group 3) from the following signals. 000: GRC93 input capture/compare match interrupt 001: OCRC93 input capture/compare match interrupt 010: OCR1D13 compare match interrupt 011: OCR2D13 compare match interrupt 100: 1shot pulse ON D13 occurrence interrupt 101: 1shot pulse OFF D13 occurrence interrupt 110: UDID13 down-counter underflow interrupt others: inhibit
27	—	Reserved. When writing to these bits, write 0.
26 to 24	ATU_DMASEL_CD38 [2:0]	Select as DMA request no.102 or 230 (group 1), DTS request no.38 or 102 group 3) from the following signals. 000: GRC92 input capture/compare match interrupt 001: OCRC92 input capture/compare match interrupt 010: OCR1D12 compare match interrupt 011: OCR2D12 compare match interrupt 100: 1shot pulse ON D12 occurrence interrupt 101: 1shot pulse OFF D12 occurrence interrupt 110: UDID12 down-counter underflow interrupt others: inhibit
23	—	Reserved. When writing to these bits, write 0.

Table 30.224 ATUDMASELCD4 Register Contents (2/3)

Bit Position	Bit Name	Function
22 to 20	ATU_DMASEL_CD37 [2:0]	Select as DMA request no.101 or 229 (group 1), DTS request no.37 or 101 (group 3) from the following signals. 000: GRC91 input capture/compare match interrupt 001: OCRC91 input capture/compare match interrupt 010: OCR1D11 compare match interrupt 011: OCR2D11 compare match interrupt 100: 1shot pulse ON D11 occurrence interrupt 101: 1shot pulse OFF D11 occurrence interrupt 110: UDID11 down-counter underflow interrupt others: inhibit
19	—	Reserved. When writing to these bits, write 0.
18 to 16	ATU_DMASEL_CD36 [2:0]	Select as DMA request no.100 or 228 (group 1), DTS request no.36 or 100 (group 3) from the following signals. 000: GRC90 input capture/compare match interrupt 001: OCRC90 input capture/compare match interrupt 010: OCR1D10 compare match interrupt 011: OCR2D10 compare match interrupt 100: 1shot pulse ON D10 occurrence interrupt 101: 1shot pulse OFF D10 occurrence interrupt 110: UDID10 down-counter underflow interrupt others: inhibit
15	—	Reserved. When writing to these bits, write 0.
14 to 12	ATU_DMASEL_CD35 [2:0]	Select as DMA request no.99 or 227 (group 1), DTS request no.35 or 99 (group 3) from the following signals. 000: GRC83 input capture/compare match interrupt 001: OCRC83 input capture/compare match interrupt 010: OCR1D03 compare match interrupt 011: OCR2D03 compare match interrupt 100: 1shot pulse ON D03 occurrence interrupt 101: 1shot pulse OFF D03 occurrence interrupt 110: UDID03 down-counter underflow interrupt others: inhibit
11	—	Reserved. When writing to these bits, write 0.
10 to 8	ATU_DMASEL_CD34 [2:0]	Select as DMA request no.98 or 226 (group 1), DTS request no.34 or 98 (group 3) from the following signals. 000: GRC82 input capture/compare match interrupt 001: OCRC82 input capture/compare match interrupt 010: OCR1D02 compare match interrupt 011: OCR2D02 compare match interrupt 100: 1shot pulse ON D02 occurrence interrupt 101: 1shot pulse OFF D02 occurrence interrupt 110: UDID02 down-counter underflow interrupt others: inhibit
7	—	Reserved. When writing to these bits, write 0.

Table 30.224 ATUDMASELCD4 Register Contents (3/3)

Bit Position	Bit Name	Function
6 to 4	ATU_DMASEL_CD33 [2:0]	Select as DMA request no.97 or 225 (group 1), DTS request no.33 or 97 (group 3) from the following signals. 000: GRC81 input capture/compare match interrupt 001: OCRC81 input capture/compare match interrupt 010: OCR1D01 compare match interrupt 011: OCR2D01 compare match interrupt 100: 1shot pulse ON D01 occurrence interrupt 101: 1shot pulse OFF D01 occurrence interrupt 110: UDID01 down-counter underflow interrupt others: inhibit
3	—	Reserved. When writing to these bits, write 0.
2 to 0	ATU_DMASEL_CD32 [2:0]	Select as DMA request no.96 or 224 (group 1), DTS request no.32 or 96 (group 3) from the following signals. 000: GRC80 input capture/compare match interrupt 001: OCRC80 input capture/compare match interrupt 010: OCR1D00 compare match interrupt 011: OCR2D00 compare match interrupt 100: 1shot pulse ON D00 occurrence interrupt 101: 1shot pulse OFF D00 occurrence interrupt 110: UDID00 down-counter underflow interrupt others: inhibit

The factor communicated to a DMA system is selected from the interrupt signal output from Timer C/D.

30.13.2.13 ATUDMASELCD5 — ATU TimerC/TimerD DMA Selection Control Register 5

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ATU_DMASEL_CD47 [2:0]			—	ATU_DMASEL_CD46 [2:0]			—	ATU_DMASEL_CD45 [2:0]			—	ATU_DMASEL_CD44 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATU_DMASEL_CD43 [2:0]			—	ATU_DMASEL_CD42 [2:0]			—	ATU_DMASEL_CD41 [2:0]			—	ATU_DMASEL_CD40 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.225 ATUDMASELCD5 Register Contents (1/3)

Bit Position	Bit Name	Function
31	—	Reserved. When writing to these bits, write 0.
30 to 28	ATU_DMASEL_CD47 [2:0]	Select as DMA request no.111 or 239 (group 1), DTS request no.47 or 111 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D33 compare match interrupt 011: OCR2D33 compare match interrupt 100: 1shot pulse ON D33 occurrence interrupt 101: 1shot pulse OFF D33 occurrence interrupt 110: UDID33 down-counter underflow interrupt others: inhibit
27	—	Reserved. When writing to these bits, write 0.
26 to 24	ATU_DMASEL_CD46 [2:0]	Select as DMA request no.110 or 238 (group 1), DTS request no.46 or 110 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D32 compare match interrupt 011: OCR2D32 compare match interrupt 100: 1shot pulse ON D32 occurrence interrupt 101: 1shot pulse OFF D32 occurrence interrupt 110: UDID32 down-counter underflow interrupt others: inhibit
23	—	Reserved. When writing to these bits, write 0.
22 to 20	ATU_DMASEL_CD45 [2:0]	Select as DMA request no.109 or 237 (group 1), DTS request no.45 or 109 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D31 compare match interrupt 011: OCR2D31 compare match interrupt 100: 1shot pulse ON D31 occurrence interrupt 101: 1shot pulse OFF D31 occurrence interrupt 110: UDID31 down-counter underflow interrupt others: inhibit

Table 30.225 ATUDMASELCD5 Register Contents (2/3)

Bit Position	Bit Name	Function
19	—	Reserved. When writing to these bits, write 0.
18 to 16	ATU_DMASEL_CD44 [2:0]	Select as DMA request no.108 or 236 (group 1), DTS request no.44 or 108 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D30 compare match interrupt 011: OCR2D30 compare match interrupt 100: 1shot pulse ON D30 occurrence interrupt 101: 1shot pulse OFF D30 occurrence interrupt 110: UDID30 down-counter underflow interrupt others: inhibit
15	—	Reserved. When writing to these bits, write 0.
14 to 12	ATU_DMASEL_CD43 [2:0]	Select as DMA request no.107 or 235 (group 1), DTS request no.43 or 107 (group 3) from the following signals. 000: GRC103 input capture/compare match interrupt 001: OCRC103 input capture/compare match interrupt 010: OCR1D23 compare match interrupt 011: OCR2D23 compare match interrupt 100: 1shot pulse ON D23 occurrence interrupt 101: 1shot pulse OFF D23 occurrence interrupt 110: UDID23 down-counter underflow interrupt others: inhibit
11	—	Reserved. When writing to these bits, write 0.
10 to 8	ATU_DMASEL_CD42 [2:0]	Select as DMA request no.106 or 234 (group 1), DTS request no.42 or 106 (group 3) from the following signals. 000: GRC102 input capture/compare match interrupt 001: OCRC102 input capture/compare match interrupt 010: OCR1D22 compare match interrupt 011: OCR2D22 compare match interrupt 100: 1shot pulse ON D22 occurrence interrupt 101: 1shot pulse OFF D22 occurrence interrupt 110: UDID22 down-counter underflow interrupt others: inhibit
7	—	Reserved. When writing to these bits, write 0.
6 to 4	ATU_DMASEL_CD41 [2:0]	Select as DMA request no.105 or 233 (group 1), DTS request no.41 or 105 (group 3) from the following signals. 000: GRC101 input capture/compare match interrupt 001: OCRC101 input capture/compare match interrupt 010: OCR1D21 compare match interrupt 011: OCR2D21 compare match interrupt 100: 1shot pulse ON D21 occurrence interrupt 101: 1shot pulse OFF D21 occurrence interrupt 110: UDID21 down-counter underflow interrupt others: inhibit

Table 30.225 ATUDMASELCD5 Register Contents (3/3)

Bit Position	Bit Name	Function
3	—	Reserved. When writing to these bits, write 0.
2 to 0	ATU_DMASEL_CD40 [2:0]	Select as DMA request no.104 or 232 (group 1), DTS request no.40 or 104 (group 3) from the following signals. 000: GRC100 input capture/compare match interrupt 001: OCRC100 input capture/compare match interrupt 010: OCR1D20 compare match interrupt 011: OCR2D20 compare match interrupt 100: 1shot pulse ON D20 occurrence interrupt 101: 1shot pulse OFF D20 occurrence interrupt 110: UDID20 down-counter underflow interrupt others: inhibit

The factor communicated to a DMA system is selected from the interrupt signal output from Timer C/D.

30.13.2.14 ATUDMASELCD6 — ATU TimerC/TimerD DMA Selection Control Register 6

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ATU_DMASEL_CD55 [2:0]			—	ATU_DMASEL_CD54 [2:0]			—	ATU_DMASEL_CD53 [2:0]			—	ATU_DMASEL_CD52 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATU_DMASEL_CD51 [2:0]			—	ATU_DMASEL_CD50 [2:0]			—	ATU_DMASEL_CD49 [2:0]			—	ATU_DMASEL_CD48 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.226 ATUDMASELCD6 Register Contents (1/3)

Bit Position	Bit Name	Function
31	—	Reserved. When writing to these bits, write 0.
30 to 28	ATU_DMASEL_CD55 [2:0]	Select as DMA request no.119 or 247 (group 1), DTS request no.55 or 119 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D53 compare match interrupt 011: OCR2D53 compare match interrupt 100: 1shot pulse ON D53 occurrence interrupt 101: 1shot pulse OFF D53 occurrence interrupt 110: UDID53 down-counter underflow interrupt others: inhibit
27	—	Reserved. When writing to these bits, write 0.
26 to 24	ATU_DMASEL_CD54 [2:0]	Select as DMA request no.118 or 246 (group 1), DTS request no.54 or 118 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D52 compare match interrupt 011: OCR2D52 compare match interrupt 100: 1shot pulse ON D52 occurrence interrupt 101: 1shot pulse OFF D52 occurrence interrupt 110: UDID52 down-counter underflow interrupt others: inhibit
23	—	Reserved. When writing to these bits, write 0.
22 to 20	ATU_DMASEL_CD53 [2:0]	Select as DMA request no.117 or 245 (group 1), DTS request no.53 or 117 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D51 compare match interrupt 011: OCR2D51 compare match interrupt 100: 1shot pulse ON D51 occurrence interrupt 101: 1shot pulse OFF D51 occurrence interrupt 110: UDID51 down-counter underflow interrupt others: inhibit

Table 30.226 ATUDMASELCD6 Register Contents (2/3)

Bit Position	Bit Name	Function
19	—	Reserved. When writing to these bits, write 0.
18 to 16	ATU_DMASEL_CD52 [2:0]	Select as DMA request no.116 or 244 (group 1), DTS request no.52 or 116 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D50 compare match interrupt 011: OCR2D50 compare match interrupt 100: 1shot pulse ON D50 occurrence interrupt 101: 1shot pulse OFF D50 occurrence interrupt 110: UDID50 down-counter underflow interrupt others: inhibit
15	—	Reserved. When writing to these bits, write 0.
14 to 12	ATU_DMASEL_CD51 [2:0]	Select as DMA request no.115 or 243 (group 1), DTS request no.51 or 115 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D43 compare match interrupt 011: OCR2D43 compare match interrupt 100: 1shot pulse ON D43 occurrence interrupt 101: 1shot pulse OFF D43 occurrence interrupt 110: UDID43 down-counter underflow interrupt others: inhibit
11	—	Reserved. When writing to these bits, write 0.
10 to 8	ATU_DMASEL_CD50 [2:0]	Select as DMA request no.114 or 242 (group 1), DTS request no.50 or 114 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D42 compare match interrupt 011: OCR2D42 compare match interrupt 100: 1shot pulse ON D42 occurrence interrupt 101: 1shot pulse OFF D42 occurrence interrupt 110: UDID42 down-counter underflow interrupt others: inhibit
7	—	Reserved. When writing to these bits, write 0.
6 to 4	ATU_DMASEL_CD49 [2:0]	Select as DMA request no.113 or 241 (group 1), DTS request no.49 or 113 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D41 compare match interrupt 011: OCR2D41 compare match interrupt 100: 1shot pulse ON D41 occurrence interrupt 101: 1shot pulse OFF D41 occurrence interrupt 110: UDID41 down-counter underflow interrupt others: inhibit

Table 30.226 ATUDMASELCD6 Register Contents (3/3)

Bit Position	Bit Name	Function
3	—	Reserved. When writing to these bits, write 0.
2 to 0	ATU_DMASEL_CD48 [2:0]	Select as DMA request no.112 or 240 (group 1), DTS request no.48 or 112 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D40 compare match interrupt 011: OCR2D40 compare match interrupt 100: 1shot pulse ON D40 occurrence interrupt 101: 1shot pulse OFF D40 occurrence interrupt 110: UDID40 down-counter underflow interrupt others: inhibit

The factor communicated to a DMA system is selected from the interrupt signal output from Timer C/D.

30.13.2.15 ATUDMASELCD7 — ATU TimerC/TimerD DMA Selection Control Register 7

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ATU_DMASEL_CD63 [2:0]			—	ATU_DMASEL_CD62 [2:0]			—	ATU_DMASEL_CD61 [2:0]			—	ATU_DMASEL_CD60 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATU_DMASEL_CD59 [2:0]			—	ATU_DMASEL_CD58 [2:0]			—	ATU_DMASEL_CD57 [2:0]			—	ATU_DMASEL_CD56 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.227 ATUDMASELCD7 Register Contents (1/2)

Bit Position	Bit Name	Function
31	—	Reserved. When writing to these bits, write 0.
30 to 28	ATU_DMASEL_CD63 [2:0]	Select as DMA request no.127 or 255 (group 1), DTS request no.63 or 127 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D73 compare match interrupt 011: OCR2D73 compare match interrupt 100: 1shot pulse ON D73 occurrence interrupt 101: 1shot pulse OFF D73 occurrence interrupt 110: UDID73 down-counter underflow interrupt others: inhibit
27	—	Reserved. When writing to these bits, write 0.
26 to 24	ATU_DMASEL_CD62 [2:0]	Select as DMA request no.126 or 254 (group 1), DTS request no.62 or 126 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D72 compare match interrupt 011: OCR2D72 compare match interrupt 100: 1shot pulse ON D72 occurrence interrupt 101: 1shot pulse OFF D72 occurrence interrupt 110: UDID72 down-counter underflow interrupt others: inhibit
23	—	Reserved. When writing to these bits, write 0.
22 to 20	ATU_DMASEL_CD61 [2:0]	Select as DMA request no.125 or 253 (group 1), DTS request no.61 or 125 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D71 compare match interrupt 011: OCR2D71 compare match interrupt 100: 1shot pulse ON D71 occurrence interrupt 101: 1shot pulse OFF D71 occurrence interrupt 110: UDID71 down-counter underflow interrupt others: inhibit
19	—	Reserved. When writing to these bits, write 0.

Table 30.227 ATUDMASELCD7 Register Contents (2/2)

Bit Position	Bit Name	Function
18 to 16	ATU_DMASEL_CD60 [2:0]	Select as DMA request no.124 or 252 (group 1), DTS request no.60 or 124 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D70 compare match interrupt 011: OCR2D70 compare match interrupt 100: 1shot pulse ON D70 occurrence interrupt 101: 1shot pulse OFF D70 occurrence interrupt 110: UDID70 down-counter underflow interrupt others: inhibit
15	—	Reserved. When writing to these bits, write 0.
14 to 12	ATU_DMASEL_CD59 [2:0]	Select as DMA request no.123 or 251 (group 1), DTS request no.59 or 123 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D63 compare match interrupt 011: OCR2D63 compare match interrupt 100: 1shot pulse ON D63 occurrence interrupt 101: 1shot pulse OFF D63 occurrence interrupt 110: UDID63 down-counter underflow interrupt others: inhibit
11	—	Reserved. When writing to these bits, write 0.
10 to 8	ATU_DMASEL_CD58 [2:0]	Select as DMA request no.122 or 250 (group 1), DTS request no.58 or 122 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D62 compare match interrupt 011: OCR2D62 compare match interrupt 100: 1shot pulse ON D62 occurrence interrupt 101: 1shot pulse OFF D62 occurrence interrupt 110: UDID62 down-counter underflow interrupt others: inhibit
7	—	Reserved. When writing to these bits, write 0.
6 to 4	ATU_DMASEL_CD57 [2:0]	Select as DMA request no.121 or 249 (group 1), DTS request no.57 or 121 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D61 compare match interrupt 011: OCR2D61 compare match interrupt 100: 1shot pulse ON D61 occurrence interrupt 101: 1shot pulse OFF D61 occurrence interrupt 110: UDID61 down-counter underflow interrupt others: inhibit
3	—	Reserved. When writing to these bits, write 0.
2 to 0	ATU_DMASEL_CD56 [2:0]	Select as DMA request no.120 or 248 (group 1), DTS request no.56 or 120 (group 3) from the following signals. 000: Output stop (output is fixed to "0".) 010: OCR1D60 compare match interrupt 011: OCR2D60 compare match interrupt 100: 1shot pulse ON D60 occurrence interrupt 101: 1shot pulse OFF D60 occurrence interrupt 110: UDID60 down-counter underflow interrupt others: inhibit

The factor communicated to a DMA system is selected from the interrupt signal output from Timer C/D.

30.13.2.16 ATUDMASELE — ATU TimerE DMA Selection Control Register

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATU_DMASEL_ E8[1:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ATU_DMASEL_ E7[1:0]	ATU_DMASEL_ E6[1:0]	ATU_DMASEL_ E5[1:0]	ATU_DMASEL_ E4[1:0]	ATU_DMASEL_ E3[1:0]	ATU_DMASEL_ E2[1:0]	ATU_DMASEL_ E1[1:0]	ATU_DMASEL_ E0[1:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.228 ATUDMASELE Register Contents

Bit Position	Bit Name	Function
31 to 18	—	Reserved. When writing to these bits, write 0.
17 to 16 (x=8)	ATU_DMASEL_ Ex	Select as DMA request 0+x or 128+x (group 1) or DTS request no.0+x or 64+x (group 2) from the following signals. 00: CYLREx0 cycle compare match interrupt / DTREx0 duty compare match interrupt 01: CYLREx1 cycle compare match interrupt / DTREx1 duty compare match interrupt 10: CYLREx2 cycle compare match interrupt / DTREx2 duty compare match interrupt 11: CYLREx3 cycle compare match interrupt / DTREx3 duty compare match interrupt
15 to 14 (x=7)	[1:0]	
13 to 12 (x=6)		
11 to 10 (x=5)		
9 to 8 (x=4)		
7 to 6 (x=3)		
5 to 4 (x=2)		
3 to 2 (x = 1)		
1 to 0 (x = 0)		

Note: x: Number of sub-block

The factor communicated to a DMA system is selected from the interrupt signal output from Timer E

30.13.2.17 ATUDFEENTQi — ATU to DFE Capture Trigger i Input Enable

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	ATU_DFEEN_TQi[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.229 ATUDFEENTQi Register Contents

Bit Position	Bit Name	Function
31 to 12	—	Reserved. When writing to these bits, write 0.
11	ATU_DFEEN_TQi[11]	Enable/disable Timer DFE Capture trigger i input factor C91 (TimerC sub-block 9, channel 1). 0: disabled, 1: enabled
10	ATU_DFEEN_TQi[10]	Enable/disable Timer DFE Capture trigger i input factor C90 (TimerC sub-block 9, channel 0). 0: disabled, 1: enabled
9	ATU_DFEEN_TQi[9]	Enable/disable Timer DFE Capture trigger i input factor C83 (TimerC sub-block 8, channel 3). 0: disabled, 1: enabled
8	ATU_DFEEN_TQi[8]	Enable/disable Timer DFE Capture trigger i input factor C82 (TimerC sub-block 8, channel 2). 0: disabled, 1: enabled
7	ATU_DFEEN_TQi[7]	Enable/disable Timer DFE Capture trigger i input factor C81 (TimerC sub-block 8, channel 1). 0: disabled, 1: enabled
6	ATU_DFEEN_TQi[6]	Enable/disable Timer DFE Capture trigger i input factor C80 (TimerC sub-block 8, channel 0). 0: disabled, 1: enabled
5	ATU_DFEEN_TQi[5]	Enable/disable Timer DFE Capture trigger i input factor D11 (TimerD sub-block 1, channel 1). 0: disabled, 1: enabled
4	ATU_DFEEN_TQi[4]	Enable/disable Timer DFE Capture trigger i input factor D10 (TimerD sub-block 1, channel 0). 0: disabled, 1: enabled
3	ATU_DFEEN_TQi[3]	Enable/disable Timer DFE Capture trigger i input factor D03 (TimerD sub-block 0, channel 3). 0: disabled, 1: enabled
2	ATU_DFEEN_TQi[2]	Enable/disable Timer DFE Capture trigger i input factor D02 (TimerD sub-block 0, channel 2). 0: disabled, 1: enabled
1	ATU_DFEEN_TQi[1]	Enable/disable Timer DFE Capture trigger i input factor D01 (TimerD sub-block 0, channel 1). 0: disabled, 1: enabled
0	ATU_DFEEN_TQi[0]	Enable/disable Timer DFE Capture trigger i input factor D00 (TimerD sub-block 0, channel 0). 0: disabled, 1: enabled

The factor communicated to a DFE (as Timer DFE Capture trigger i input: i = 0 to 2) is selected from the interrupt signal output from Timer C/D. ATUDFEENTQi can control the enable/disable.

30.13.2.18 ATUDFESELi — ATU to DFE Capture Trigger i Input Select

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	ATU_D FESEL C91_i	ATU_D FESEL C90_i	ATU_D FESEL C83_i	ATU_D FESEL C82_i	ATU_D FESEL C81_i	ATU_D FESEL C80_i	—	ATU_DFESEL_D11_i [2:0]	—	ATU_DFESEL_D10_i [2:0]	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATU_DFESEL_D03_i [2:0]	—	—	—	ATU_DFESEL_D02_i [2:0]	—	—	—	ATU_DFESEL_D01_i [2:0]	—	—	—	—	ATU_DFESEL_D00_i [2:0]	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.230 ATUDFESELi Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 30	—	Reserved. When writing to these bits, write 0.
29	ATU_DFESEL_C91_i	Select as DFE Timer DFE Capture trigger i input factor C91 from the following signals. 0: GRC91 input capture/compare match interrupt 1: OCRC91 input capture/compare match interrupt
28	ATU_DFESEL_C90_i	Select as DFE Timer DFE Capture trigger i input factor C90 from the following signals. 0: GRC90 input capture/compare match interrupt 1: OCRC90 input capture/compare match interrupt
27	ATU_DFESEL_C83_i	Select as DFE Timer DFE Capture trigger i input factor C83 from the following signals. 0: GRC83 input capture/compare match interrupt 1: OCRC83 input capture/compare match interrupt
26	ATU_DFESEL_C82_i	Select as DFE Timer DFE Capture trigger i input factor C82 from the following signals. 0: GRC82 input capture/compare match interrupt 1: OCRC82 input capture/compare match interrupt
25	ATU_DFESEL_C81_i	Select as DFE Timer DFE Capture trigger i input factor C81 from the following signals. 0: GRC81 input capture/compare match interrupt 1: OCRC81 input capture/compare match interrupt
24	ATU_DFESEL_C80_i	Select as DFE Timer DFE Capture trigger i input factor C80 from the following signals. 0: GRC80 input capture/compare match interrupt 1: OCRC80 input capture/compare match interrupt
23	—	Reserved. When writing to these bits, write 0.
22 to 20	ATU_DFESEL_D11_i [2:0]	Select as DFE Timer DFE Capture trigger i input factor D11 from the following signals. 000: OCR1D11 compare match interrupt 001: OCR2D11 compare match interrupt 010: 1shot pulse ON D11 occurrence interrupt 011: 1shot pulse OFF D11 occurrence interrupt 100: UDID11 down-counter underflow interrupt Others inhibit
19	—	Reserved. When writing to these bits, write 0.

Table 30.230 ATUDFESEL_i Register Contents (2/2)

Bit Position	Bit Name	Function
18 to 16	ATU_DFESEL_D10 _i [2:0]	Select as DFE Timer DFE Capture trigger <i>i</i> input factor D10 from the following signals. 000: OCR1D10 compare match interrupt 001: OCR2D10 compare match interrupt 010: 1shot pulse ON D10 occurrence interrupt 011: 1shot pulse OFF D10 occurrence interrupt 100: UDID10 down-counter underflow interrupt Others inhibit
15	—	Reserved. When writing to these bits, write 0.
14 to 12	ATU_DFESEL_D03 _i [2:0]	Select as DFE Timer DFE Capture trigger <i>i</i> input factor D03 from the following signals. 000: OCR1D03 compare match interrupt 001: OCR2D03 compare match interrupt 010: 1shot pulse ON D03 occurrence interrupt 011: 1shot pulse OFF D03 occurrence interrupt 100: UDID03 down-counter underflow interrupt Others inhibit
11	—	Reserved. When writing to these bits, write 0.
10 to 8	ATU_DFESEL_D02 _i [2:0]	Select as DFE Timer DFE Capture trigger <i>i</i> input factor D02 from the following signals. 000: OCR1D02 compare match interrupt 001: OCR2D02 compare match interrupt 010: 1shot pulse ON D02 occurrence interrupt 011: 1shot pulse OFF D02 occurrence interrupt 100: UDID02 down-counter underflow interrupt Others inhibit
7	—	Reserved. When writing to these bits, write 0.
6 to 4	ATU_DFESEL_D01 _i [2:0]	Select as DFE Timer DFE Capture trigger <i>i</i> input factor D01 from the following signals. 000: OCR1D01 compare match interrupt 001: OCR2D01 compare match interrupt 010: 1shot pulse ON D01 occurrence interrupt 011: 1shot pulse OFF D01 occurrence interrupt 100: UDID01 down-counter underflow interrupt Others inhibit
3	—	Reserved. When writing to these bits, write 0.
2 to 0	ATU_DFESEL_D00 _i [2:0]	Select as DFE Timer DFE Capture trigger <i>i</i> input factor D00 from the following signals. 000: OCR1D00 compare match interrupt 001: OCR2D00 compare match interrupt 010: 1shot pulse ON D00 occurrence interrupt 011: 1shot pulse OFF D00 occurrence interrupt 100: UDID00 down-counter underflow interrupt Others inhibit

The factor communicated to a DFE (as Timer DFE Capture trigger *i* input: *i* = 0 to 2) is selected from the interrupt signal output from Timer C/D.

30.13.2.19 ATUDFESELD1T — ATU to DFE Timer FIFO Capture Trigger Select

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATU_DFESEL_D1T [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 30.231 ATUDFESELD1T Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved. When writing to these bits, write 0.
2 to 0	ATU_DFESEL_D1T [2:0]	Select as DFE Timer FIFO Capture trigger input from the following signals. 000: OCRG0 compare-match interrupt 001: OCRG1 compare-match interrupt 010: OCRG2 compare-match interrupt 011: OCRG3 compare-match interrupt 100: OCRG4 compare-match interrupt 101: OCRG5 compare-match interrupt 110: OCRG6 compare-match interrupt 111: OCRG7 compare-match interrupt

The factor communicated to a DFE (as Timer FIFO Capture trigger input) is selected from the interrupt signal output from Timer C/D.

30.13.2.20 ATUDSSELDSTS — ATU to DSADC Read Gate Trigger Select

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	ATU_DSSEL_D STS9[1:0]	ATU_DSSEL_D STS8[1:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ATU_DSSEL_D STS7[1:0]	ATU_DSSEL_D STS6[1:0]	ATU_DSSEL_D STS5[1:0]	ATU_DSSEL_D STS4[1:0]	ATU_DSSEL_D STS3[1:0]	ATU_DSSEL_D STS2[1:0]	ATU_DSSEL_D STS1[1:0]	ATU_DSSEL_D STS0[1:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.232 ATUDSSELDSTS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 20	—	Reserved. When writing to these bits, write 0.
19 to 18	ATU_DSSEL_DSTS9 [1:0]	Select as DS-ADC read gate trigger 9 from the following signals. 00: TIOC21 Timer C (sub-block 2, channel 1) compare-match output 01: TOD21A Timer D (sub-block 2, channel 1) compare-match A output 10: TOD21B Timer D (sub-block 2, channel 1) compare-match B output 11: TOE21 Timer E (sub-block 2, channel 1) PWM output
17 to 16	ATU_DSSEL_DSTS8 [1:0]	Select as DS-ADC read gate trigger 8 from the following signals. 00: TIOC20 Timer C (sub-block 2, channel 0) compare-match output 01: TOD20A Timer D (sub-block 2, channel 0) compare-match A output 10: TOD20B Timer D (sub-block 2, channel 0) compare-match B output 11: TOE20 Timer E (sub-block 2, channel 0) PWM output
15 to 14	ATU_DSSEL_DSTS7 [1:0]	Select as DS-ADC read gate trigger 7 from the following signals. 00: TIOC13 Timer C (sub-block 1, channel 3) compare-match output 01: TOD13A Timer D (sub-block 1, channel 3) compare-match A output 10: TOD13B Timer D (sub-block 1, channel 3) compare-match B output 11: TOE13 Timer E (sub-block 1, channel 3) PWM output
13 to 12	ATU_DSSEL_DSTS6 [1:0]	Select as DS-ADC read gate trigger 6 from the following signals. 00: TIOC12 Timer C (sub-block 1, channel 2) compare-match output 01: TOD12A Timer D (sub-block 1, channel 2) compare-match A output 10: TOD12B Timer D (sub-block 1, channel 2) compare-match B output 11: TOE12 Timer E (sub-block 1, channel 2) PWM output
11 to 10	ATU_DSSEL_DSTS5 [1:0]	Select as DS-ADC read gate trigger 5 from the following signals. 00: TIOC11 Timer C (sub-block 1, channel 1) compare-match output 01: TOD11A Timer D (sub-block 1, channel 1) compare-match A output 10: TOD11B Timer D (sub-block 1, channel 1) compare-match B output 11: TOE11 Timer E (sub-block 1, channel 1) PWM output

Table 30.232 ATUDSSELDSTS Register Contents (2/2)

Bit Position	Bit Name	Function
9 to 8	ATU_DSSEL_DSTS4 [1:0]	Select as DS-ADC read gate trigger 4 from the following signals. 00: TIOC10 Timer C (sub-block 1, channel 0) compare-match output 01: TOD10A Timer D (sub-block 1, channel 0) compare-match A output 10: TOD10B Timer D (sub-block 1, channel 0) compare-match B output 11: TOE10 Timer E (sub-block 1, channel 0) PWM output
7 to 6	ATU_DSSEL_DSTS3 [1:0]	Select as DS-ADC read gate trigger 3 from the following signals. 00: TIOC03 Timer C (sub-block 0, channel 3) compare-match output 01: TOD03A Timer D (sub-block 0, channel 3) compare-match A output 10: TOD03B Timer D (sub-block 0, channel 3) compare-match B output 11: TOE03 Timer E (sub-block 0, channel 3) PWM output
5 to 4	ATU_DSSEL_DSTS2 [1:0]	Select as DS-ADC read gate trigger 2 from the following signals. 00: TIOC02 Timer C (sub-block 0, channel 2) compare-match output 01: TOD02A Timer D (sub-block 0, channel 2) compare-match A output 10: TOD02B Timer D (sub-block 0, channel 2) compare-match B output 11: TOE02 Timer E (sub-block 0, channel 2) PWM output
3 to 2	ATU_DSSEL_DSTS1 [1:0]	Select as DS-ADC read gate trigger 1 from the following signals. 00: TIOC01 Timer C (sub-block 0, channel 1) compare-match output 01: TOD01A Timer D (sub-block 0, channel 1) compare-match A output 10: TOD01B Timer D (sub-block 0, channel 1) compare-match B output 11: TOE01 Timer E (sub-block 0, channel 1) PWM output
1 to 0	ATU_DSSEL_DSTS0 [1:0]	Select as DS-ADC read gate trigger 0 from the following signals. 00: TIOC00 Timer C (sub-block 0, channel 0) compare-match output 01: TOD00A Timer D (sub-block 0, channel 0) compare-match A output 10: TOD00B Timer D (sub-block 0, channel 0) compare-match B output 11: TOE00 Timer E (sub-block 0, channel 0) PWM output

The factor communicated to a DS-ADC (as read gate input) is selected from the external-output signal from Timer C/D/E.

30.13.2.21 ATUCASELCATS — ATU to CADC Read Gate Trigger Select

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATU_CASEL_C ATS0[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 30.233 ATUCASELCATS Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved. When writing to these bits, write 0.
1 to 0	ATU_CASEL_CATS0 [1:0]	Select as C-ADC read gate trigger 0 from the following signals. 00: TIOC22 Timer C (sub-block 2, channel 2) compare-match output 01: TOD22A Timer D (sub-block 2, channel 2) compare-match A output 10: TOD22B Timer D (sub-block 2, channel 2) compare-match B output 11: TOE22 Timer E (sub-block 2, channel 2) PWM output

The factor communicated to a C-ADC (as read gate input) is selected from the external-output signal from Timer C/D/E.

30.13.2.22 ATUP5SSEL — ATU to PSI5S Time Stamp Trigger Select

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	ATU_P5SSEL_SSTS _B [1:0]	ATU_P5SSEL_SSTS _A [1:0]	ATU_P5SSEL_CRTS _B [1:0]	ATU_P5SSEL_CRTS _A [1:0]	ATU_P5SSEL_CKTS _B [1:0]	ATU_P5SSEL_CKTS _A [1:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ATU_P5SSEL_GC7[1:0]	ATU_P5SSEL_GC6[1:0]	ATU_P5SSEL_GC5[1:0]	ATU_P5SSEL_GC4[1:0]	ATU_P5SSEL_GC3[1:0]	ATU_P5SSEL_GC2[1:0]	ATU_P5SSEL_GC1[1:0]	—	—							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 30.234 ATUP5SSEL Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	—	Reserved. When writing to these bits, write 0.
27 to 26	ATU_P5SSEL_SSTS _B [1:0]	Select as PSI5-S Run-stop time stamp B from the following signals. 00: TIOC31 Timer C (sub-block 3, channel 1) compare-match output 01: TOD31A Timer D (sub-block 3, channel 1) compare-match A output 10: TOD31B Timer D (sub-block 3, channel 1) compare-match B output 11: TOE31 Timer E (sub-block 3, channel 1) PWM output
25 to 24	ATU_P5SSEL_SSTS _A [1:0]	Select as PSI5-S Run-stop time stamp A from the following signals. 00: TIOC30 Timer C (sub-block 3, channel 0) compare-match output 01: TOD30A Timer D (sub-block 3, channel 0) compare-match A output 10: TOD30B Timer D (sub-block 3, channel 0) compare-match B output 11: TOE30 Timer E (sub-block 3, channel 0) PWM output
23 to 22	ATU_P5SSEL_CRTS _B [1:0]	Select as PSI5-S clear time stamp B from the following signals. 00: TIOC23 Timer C (sub-block 2, channel 3) compare-match output 01: TOD23A Timer D (sub-block 2, channel 3) compare-match A output 10: TOD23B Timer D (sub-block 2, channel 3) compare-match B output 11: TOE23 Timer E (sub-block 2, channel 3) PWM output
21 to 20	ATU_P5SSEL_CRTS _A [1:0]	Select as PSI5-S clear time stamp A from the following signals. 00: TIOC22 Timer C (sub-block 2, channel 2) compare-match output 01: TOD22A Timer D (sub-block 2, channel 2) compare-match A output 10: TOD22B Timer D (sub-block 2, channel 2) compare-match B output 11: TOE22 Timer E (sub-block 2, channel 2) PWM output
19 to 18	ATU_P5SSEL_CKTS _B [1:0]	Select as PSI5-S clock time stamp B from the following signals. 00: TIOC21 Timer C (sub-block 2, channel 1) compare-match output 01: TOD21A Timer D (sub-block 2, channel 1) compare-match A output 10: TOD21B Timer D (sub-block 2, channel 1) compare-match B output 11: TOE21 Timer E (sub-block 2, channel 1) PWM output

Table 30.234 ATUP5SSEL Register Contents (2/2)

Bit Position	Bit Name	Function
17 to 16	ATU_P5SSEL_CKTS A[1:0]	Select as PSI5-S clock time stamp A from the following signals. 00: TIOC20 Timer C (sub-block 2, channel 0) compare-match output 01: TOD20A Timer D (sub-block 2, channel 0) compare-match A output 10: TOD20B Timer D (sub-block 2, channel 0) compare-match B output 11: TOE20 Timer E (sub-block 2, channel 0) PWM output
15 to 14	ATU_P5SSEL_TGC7[1:0]	Select as PSI5-S Trigger sync channel 7 from the following signals. 00: TIOC13 Timer C (sub-block 1, channel 3) compare-match output 01: TOD13A Timer D (sub-block 1, channel 3) compare-match A output 10: TOD13B Timer D (sub-block 1, channel 3) compare-match B output 11: TOE13 Timer E (sub-block 1, channel 3) PWM output
13 to 12	ATU_P5SSEL_TGC6[1:0]	Select as PSI5-S Trigger sync channel 6 from the following signals. 00: TIOC12 Timer C (sub-block 1, channel 2) compare-match output 01: TOD12A Timer D (sub-block 1, channel 2) compare-match A output 10: TOD12B Timer D (sub-block 1, channel 2) compare-match B output 11: TOE12 Timer E (sub-block 1, channel 2) PWM output
11 to 10	ATU_P5SSEL_TGC5[1:0]	Select as PSI5-S Trigger sync channel 5 from the following signals. 00: TIOC11 Timer C (sub-block 1, channel 1) compare-match output 01: TOD11A Timer D (sub-block 1, channel 1) compare-match A output 10: TOD11B Timer D (sub-block 1, channel 1) compare-match B output 11: TOE11 Timer E (sub-block 1, channel 1) PWM output
9 to 8	ATU_P5SSEL_TGC4[1:0]	Select as PSI5-S Trigger sync channel 4 from the following signals. 00: TIOC10 Timer C (sub-block 1, channel 0) compare-match output 01: TOD10A Timer D (sub-block 1, channel 0) compare-match A output 10: TOD10B Timer D (sub-block 1, channel 0) compare-match B output 11: TOE10 Timer E (sub-block 1, channel 0) PWM output
7 to 6	ATU_P5SSEL_TGC3[1:0]	Select as PSI5-S Trigger sync channel 3 from the following signals. 00: TIOC03 Timer C (sub-block 0, channel 3) compare-match output 01: TOD03A Timer D (sub-block 0, channel 3) compare-match A output 10: TOD03B Timer D (sub-block 0, channel 3) compare-match B output 11: TOE03 Timer E (sub-block 0, channel 3) PWM output
5 to 4	ATU_P5SSEL_TGC2[1:0]	Select as PSI5-S Trigger sync channel 2 from the following signals. 00: TIOC02 Timer C (sub-block 0, channel 2) compare-match output 01: TOD02A Timer D (sub-block 0, channel 2) compare-match A output 10: TOD02B Timer D (sub-block 0, channel 2) compare-match B output 11: TOE02 Timer E (sub-block 0, channel 2) PWM output
3 to 2	ATU_P5SSEL_TGC1[1:0]	Select as PSI5-S Trigger sync channel 1 from the following signals. 00: TIOC01 Timer C (sub-block 0, channel 1) compare-match output 01: TOD01A Timer D (sub-block 0, channel 1) compare-match A output 10: TOD01B Timer D (sub-block 0, channel 1) compare-match B output 11: TOE01 Timer E (sub-block 0, channel 1) PWM output
1 to 0	—	Reserved. When writing to these bits, write 0.

The factor communicated to a PSI5-S (as Time Stamp input) is selected from the external-output signal from Timer C/D/E.

30.13.3 Details of Operation

30.13.3.1 Interrupt signal communicated to INTC is selected.

The function for which interrupt number allocation of an interrupt factor of ATU is selected.

- Timer A

4 of allocation is possible from “ICRA0-7 input capture interrupt” of timer A. One is selected from ICRA0-7 and a factor is assigned by setting of ATUINTELSA register. It can be allocated to interrupt number 10-13.

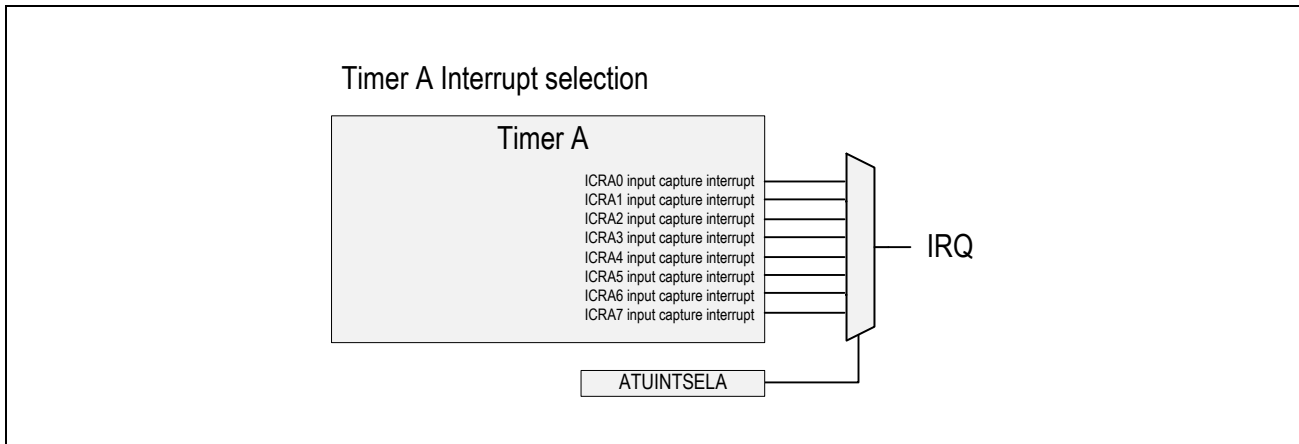


Figure 30.101 Block Diagram of Timer A Interrupt Selection.

- Timer D

One factor is selected from “OCR1Dxy compare match interrupt”, “OCR2Dxy compare match interrupt”, “1shot pulse ON Dxy occurrence interrupt”, “1shot pulse OFF Dxy occurrence interrupt” and “UDIDxy down-counter underflow interrupt” of timer D and it’s assigned to interrupt. The interrupt number and correspondence with subblock x and channel y are shown to **Table 30.235**.

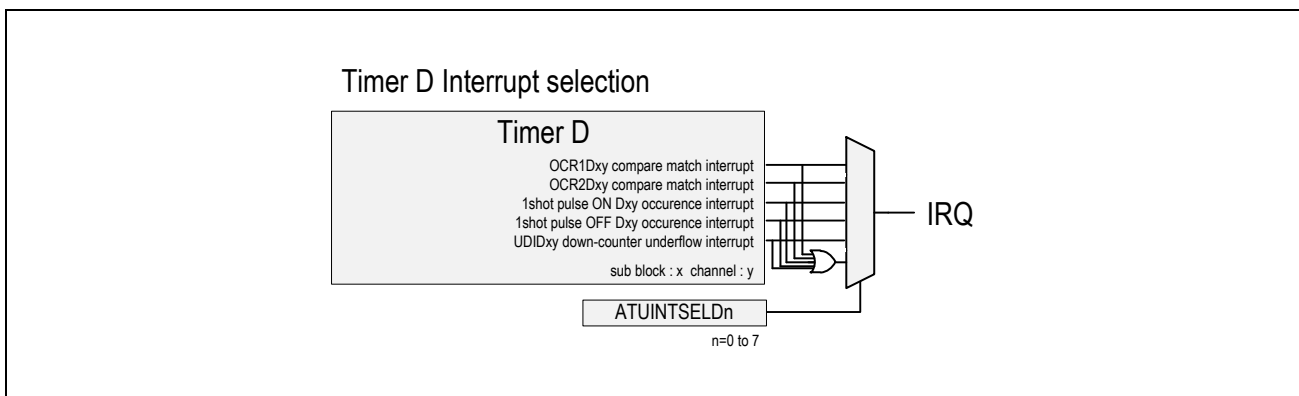


Figure 30.102 Block Diagram of Timer D Interrupt Selection.

Table 30.235 Interrupt Number and Contrast in a Timer D Subblock Channel

Interrupt Number	Timer D	
	Sub Block x	Channel y
140	0	0
141	0	1
142	0	2
143	0	3
144	1	0
145	1	1
146	1	2
147	1	3
148	2	0
149	2	1
150	2	2
151	2	3
152	3	0
153	3	1
154	3	2
155	3	3
156	4	0
157	4	1
158	4	2
159	4	3
160	5	0
161	5	1
162	5	2
163	5	3
164	6	0
165	6	1
166	6	2
167	6	3
168	7	0
169	7	1
170	7	2
171	7	3
172	8	0
173	8	1
174	8	2
175	8	3

30.13.3.2 Interrupt signal communicated to sDMA/DTS is selected.

- Timer B

One factor is selected from “OCRB0, OCRB1, OCRB6, OCRB10, OCRB11, OCRB12 compare match interrupt”, “Comparison between TCNTB6M and ICRB6 condition match interrupt”, “AND/OR condition of CMFB6 and CMFB6M condition match interrupt” and “ICRB0 input capture interrupt” of timer B and it’s assigned to sDMA / DTS.

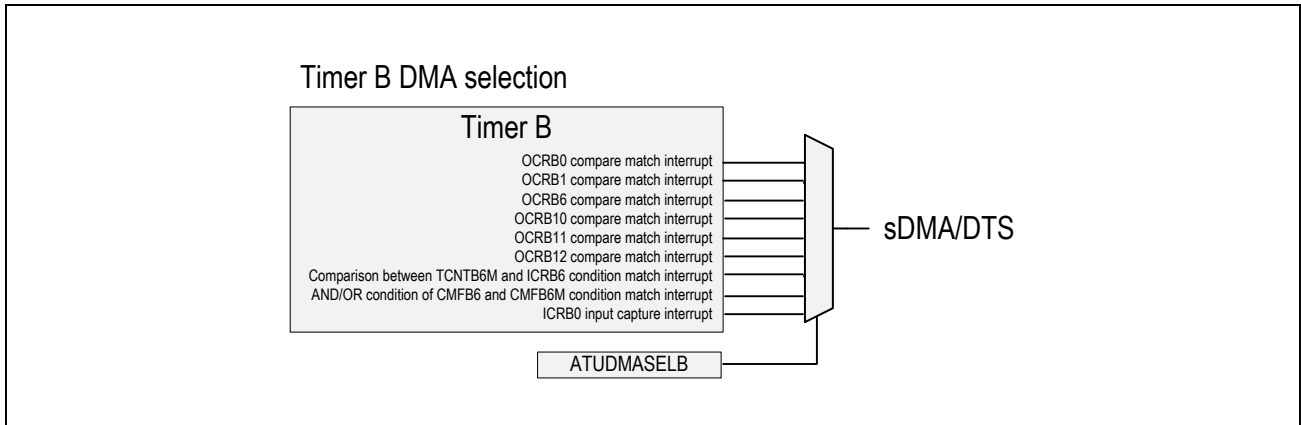


Figure 30.103 Block Diagram of Timer B DMA Selection.

• Timer C / Timer D

One factor is selected from “GRCxy input capture/compare match interrupt” and “OCRCxy input capture/compare match interrupt” of timer C and “OCR1Dxy, OCR2Dxy compare match interrupt”, “1shot pulse ON Dxy occurrence interrupt”, “1shot pulse OFF Dxy occurrence interrupt” and “UDIDxy down-counter underflow interrupt” of timer D and it’s assigned sDMA / DTS. The sDMA/DTS number and correspondence with subblock x and channel y are shown to **Table 30.236**.

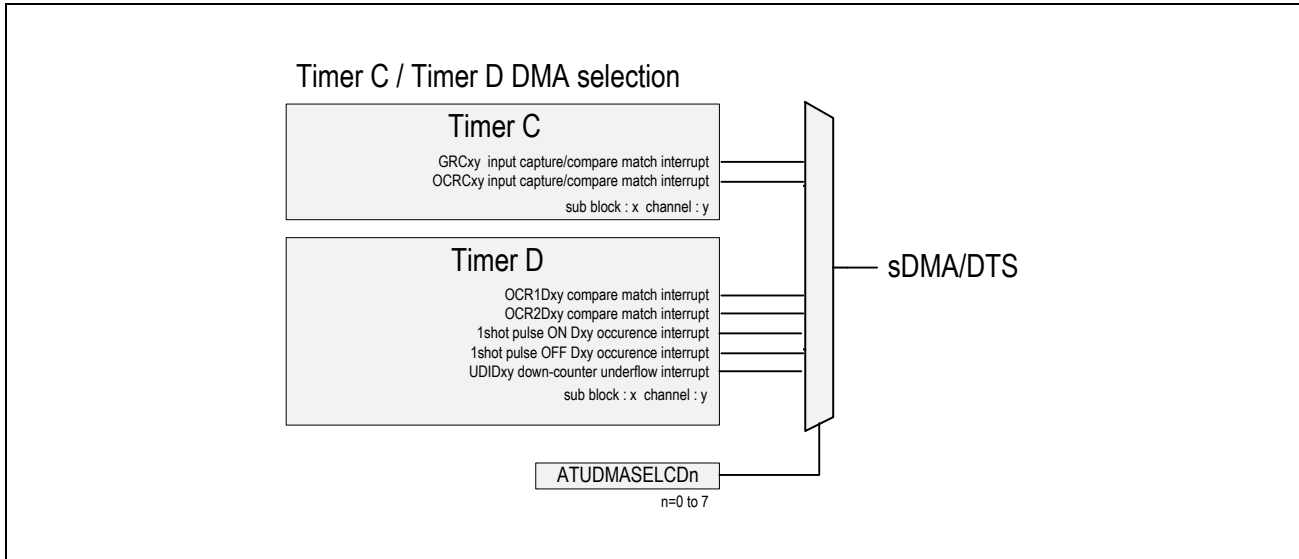


Figure 30.104 Block Diagram of Timer C / Timer D DMA Selection.

Table 30.236 sDMA/DTS Trigger Number and Contrast in a Timer C and D Subblock Channel (1/2)

sDMA Trigger		DTS Trigger		Timer C		Timer D	
Number (Group1)		Number (Group3)		Sub Block x	Channel y	Sub Block x	Channel y
64	192	0	64	0	0	8	0
65	193	1	65	0	1	8	1
66	194	2	66	0	2	8	2
67	195	3	67	0	3	8	3
68	196	4	68	1	0	—	—
69	197	5	69	1	1	—	—
70	198	6	70	1	2	—	—
71	199	7	71	1	3	—	—
72	200	8	72	2	0	—	—
73	201	9	73	2	1	—	—
74	202	10	74	2	2	—	—
75	203	11	75	2	3	—	—
76	204	12	76	3	0	—	—
77	205	13	77	3	1	—	—
78	206	14	78	3	2	—	—
79	207	15	79	3	3	—	—
80	208	16	80	4	0	—	—

Table 30.236 sDMA/DTS Trigger Number and Contrast in a Timer C and D Subblock Channel (2/2)

sDMA Trigger		DTS Trigger		Timer C		Timer D	
Number (Group1)		Number (Group3)		Sub Block x	Channel y	Sub Block x	Channel y
81	209	17	81	4	1	—	—
82	210	18	82	4	2	—	—
83	211	19	83	4	3	—	—
84	212	20	84	5	0	—	—
85	213	21	85	5	1	—	—
86	214	22	86	5	2	—	—
87	215	23	87	5	3	—	—
88	216	24	88	6	0	—	—
89	217	25	89	6	1	—	—
90	218	26	90	6	2	—	—
91	219	27	91	6	3	—	—
92	220	28	92	7	0	—	—
93	221	29	93	7	1	—	—
94	222	30	94	7	2	—	—
95	223	31	95	7	3	—	—
96	224	32	96	8	0	0	0
97	225	33	97	8	1	0	1
98	226	34	98	8	2	0	2
99	227	35	99	8	3	0	3
100	228	36	100	9	0	1	0
101	229	37	101	9	1	1	1
102	230	38	102	9	2	1	2
103	231	39	103	9	3	1	3
104	232	40	104	10	0	2	0
105	233	41	105	10	1	2	1
106	234	42	106	10	2	2	2
107	235	43	107	10	3	2	3
108	236	44	108	—	—	3	0
109	237	45	109	—	—	3	1
110	238	46	110	—	—	3	2
111	239	47	111	—	—	3	3
112	240	48	112	—	—	4	0
113	241	49	113	—	—	4	1
114	242	50	114	—	—	4	2
115	243	51	115	—	—	4	3
116	244	52	116	—	—	5	0
117	245	53	117	—	—	5	1
118	246	54	118	—	—	5	2
119	247	55	119	—	—	5	3
120	248	56	120	—	—	6	0
121	249	57	121	—	—	6	1
122	250	58	122	—	—	6	2
123	251	59	123	—	—	6	3
124	252	60	124	—	—	7	0
125	253	61	125	—	—	7	1
126	254	62	126	—	—	7	2
127	255	63	127	—	—	7	3

• Timer E

One factor is selected from “CYLREx0-3 cycle compare match interrupt / DTREx0-3 duty compare match interrupt” of timer E and it’s assigned to sDMA/DTS. The sDMA/DTS number and correspondence with subblock x are shown to **Table 30.237**.

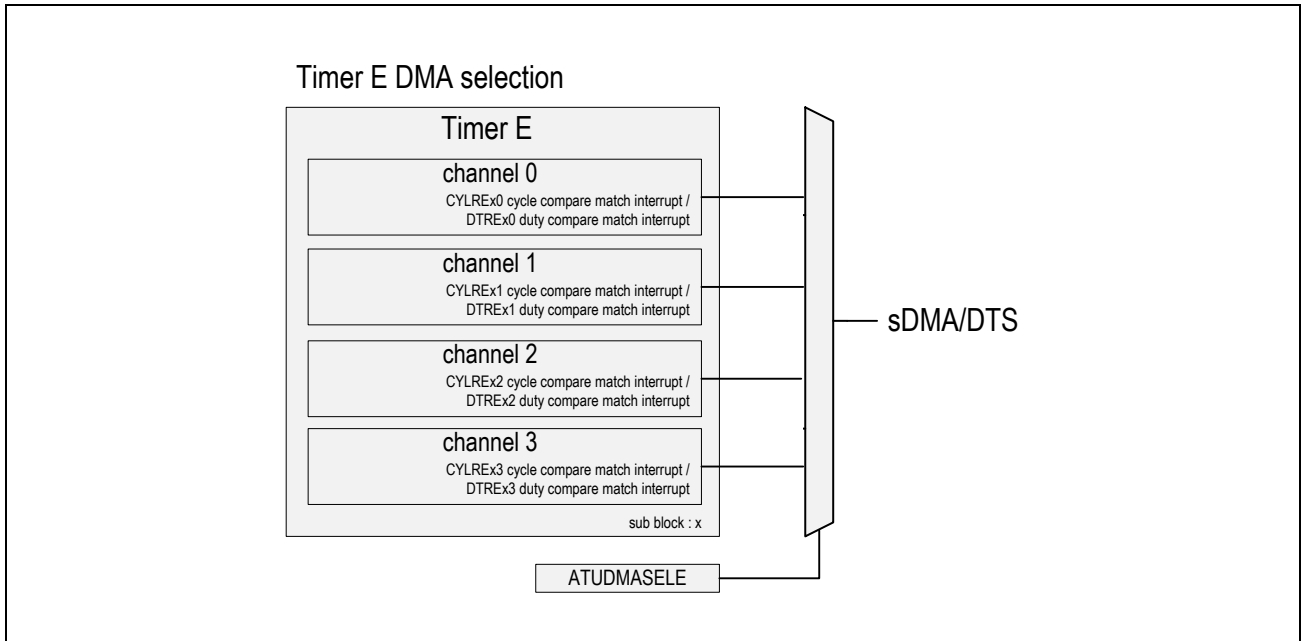


Figure 30.105 Block Diagram of Timer E DMA Selection.

Table 30.237 sDMA/DTS Trigger Number and Contrast in a Timer E Subblock

sDMA Trigger Number (Group1)		DTS Trigger Number (Group2)		Timer E Sub Block x
0	128	0	64	0
1	129	1	65	1
2	130	2	66	2
3	131	3	67	3
4	132	4	68	4
5	133	5	69	5
6	134	6	70	6
7	135	7	71	7
8	136	8	72	8

30.13.3.3 “Timer DFE Capture trigger i (i = 0 to 2)” communicated to DFE selected.

The factor selected from “GRCxy input capture/compare match interrupt” or “OCRCxy input capture/compare match interrupt” in timer C and “OCR1Dxy, OCR2Dxy compare match interrupt”, “1shot pulse ON Dxy occurrence interrupt”, “1shot pulse OFF Dxy occurrence interrupt” and “UDIDxy down-counter underflow interrupt” of timer D is assigned to a trigger of DFE. A factor of each timer can change enable or disable. It can be chosen from several subblocks and a channel from timer C and timer D. The subblock which can be chosen and a channel are shown to **Table 30.238**.

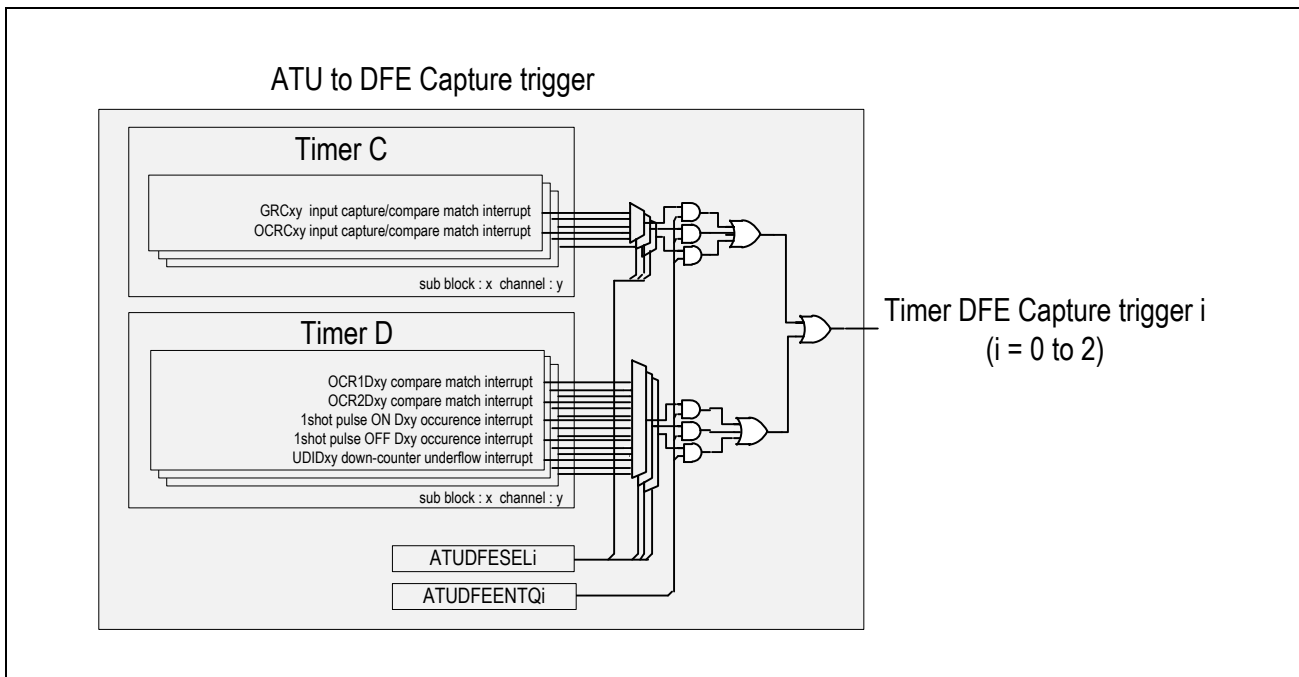


Figure 30.106 Block Diagram of ATU to DFE Capture Trigger Selection.

Table 30.238 DFE Trigger and Contrast in a Timer C and D Subblock (1/2)

Trigger Function	Timer	Sub Block x	Channel y
Timer DFE Capture trigger 0	Timer C	8	0
		8	1
		8	2
		8	3
		9	0
		9	1
	Timer D	0	0
		0	1
		0	2
		0	3
		1	0
	1	1	

Table 30.238 DFE Trigger and Contrast in a Timer C and D Subblock (2/2)

Trigger Function	Timer	Sub Block x	Channel y
Timer DFE Capture trigger 1	Timer C	8	0
		8	1
		8	2
		8	3
		9	0
		9	1
	Timer D	0	0
		0	1
		0	2
		0	3
		1	0
		1	1
Timer DFE Capture trigger 2	Timer C	8	0
		8	1
		8	2
		8	3
		9	0
		9	1
	Timer D	0	0
		0	1
		0	2
		0	3
		1	0
		1	1

30.13.3.4 “Timer FIFO Capture trigger” communicated to DFE is selected.

One factor is selected from “OCRG0-OCRG7 compare-match interrupt” of timer G and it’s assigned to interrupt.

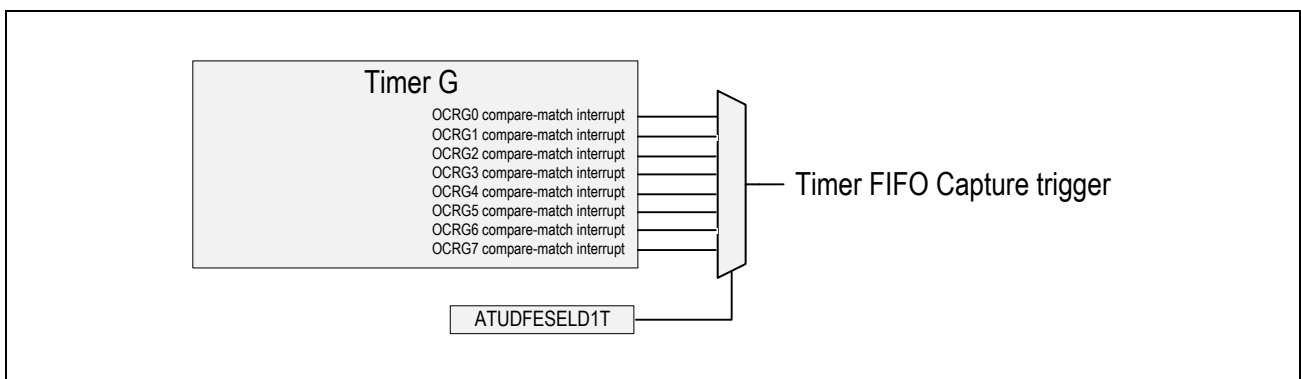


Figure 30.107 Block Diagram of ATU to DFE Capture Trigger Selection.

30.13.3.5 “read gate” communicated to DSADC is selected.

One factor is selected from TIOCxy, TODxyA, TODxyB and TOExy and it’s assigned to read gate in DSADC.

Correspondence with a subblock of the unit of DSADC and each timer and a channel is shown to **Table 30.239**.

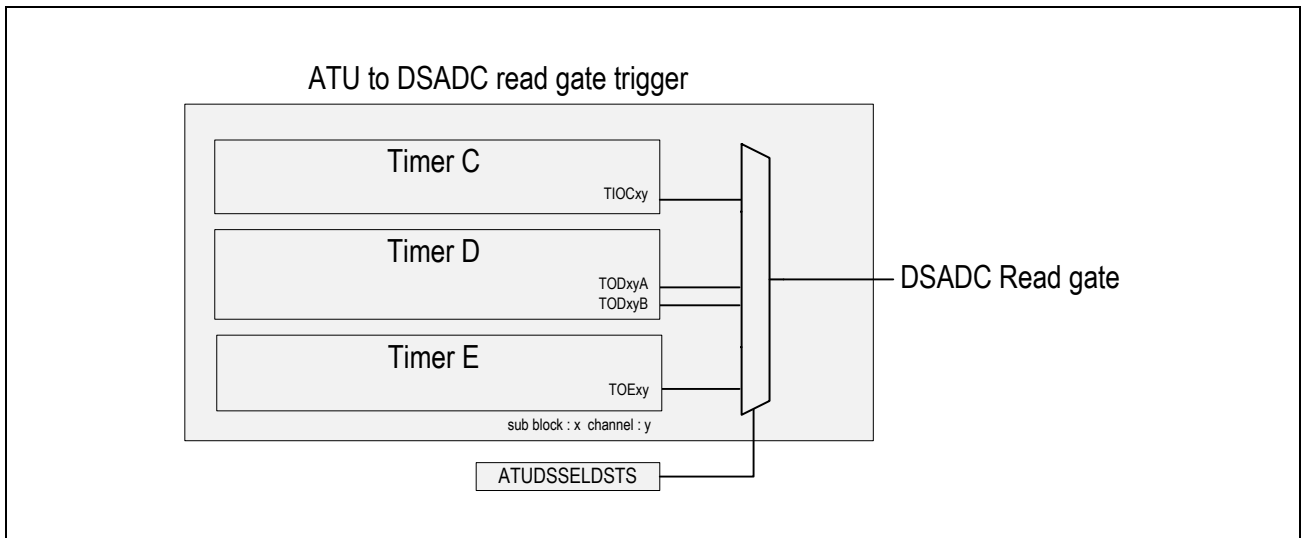


Figure 30.108 Block Diagram of ATU to DSADC Read Gate Trigger Selection.

Table 30.239 DSADC Channel Number and Contrast in a Timer C, D and E Subblock and Channel

Unit	Timer C		Timer D		Timer E	
	Sub Block x	Channel y	Sub Block x	Channel y	Sub Block x	Channel y
DSADC00	0	0	0	0	0	0
DSADC10	0	1	0	1	0	1
DSADC20	0	2	0	2	0	2
DSADC12	0	3	0	3	0	3
DSADC13	1	0	1	0	1	0
DSADC11	1	1	1	1	1	1
DSADC15	1	2	1	2	1	2
DSADC14	1	3	1	3	1	3
DSADC22	2	0	2	0	2	0
DSADC21	2	1	2	1	2	1

30.13.3.6 “read gate” communicated to CADC is selected.

One factor is selected from TIOCxy, TODxyA, TODxyB and TOExy and it’s assigned to read gate in CADC.

Correspondence with a subblock of the unit of CADC and each timer and a channel is shown to **Table 30.240**

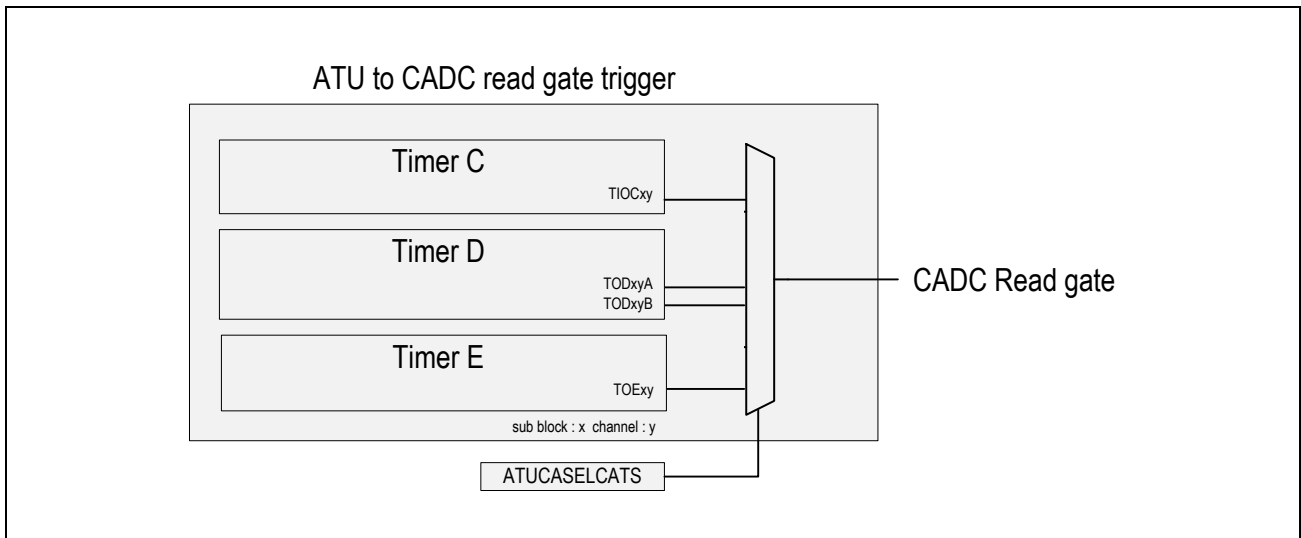


Figure 30.109 Block Diagram of ATU to CADC Read Gate Trigger Selection.

Table 30.240 CADC Channel Number and Contrast in a Timer C, D and E Subblock and Channel

Unit	Timer C		Timer D		Timer E	
	Sub Block x	Channel y	Sub Block x	Channel y	Sub Block x	Channel y
CADC00	2	2	2	2	2	2

30.13.3.7 “time stamp” and “sync pulse” communicated to PSI5S is selected.

One factor is selected from TIOCxy, TODxyA, TODxyB and TOExy and it's assigned to time stamp and sync pulse in PSI5S. Correspondence with a subblock of a trigger of PSI5S and each timer and a channel is shown to **Table 30.241**.

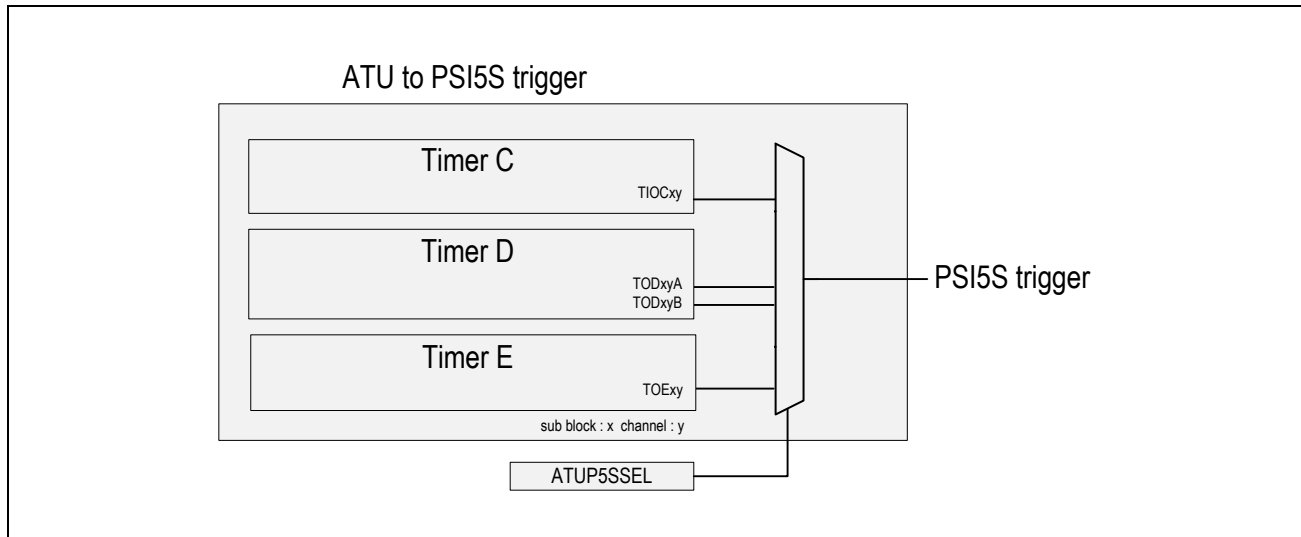


Figure 30.110 Block Diagram of ATU to PSI5S Time Stamp Trigger Selection.

Table 30.241 PSI5S Trigger Function and Contrast in a Timer C, D and E Subblock and Channel

Trigger Function	Timer C		Timer D		Timer E	
	Sub Block x	Channel y	Sub Block x	Channel y	Sub Block x	Channel y
trigger sync channel 0	0	0	0	0	0	0
trigger sync channel 1	0	1	0	1	0	1
trigger sync channel 2	0	2	0	2	0	2
trigger sync channel 3	0	3	0	3	0	3
trigger sync channel 4	1	0	1	0	1	0
trigger sync channel 5	1	1	1	1	1	1
trigger sync channel 6	1	2	1	2	1	2
trigger sync channel 7	1	3	1	3	1	3
clock timestamp A	2	0	2	0	2	0
clock timestamp B	2	1	2	1	2	1
clear timestamp A	2	2	2	2	2	2
clear timestamp B	2	3	2	3	2	3
Run-stop timestamp A	3	0	3	0	3	0
Run-stop timestamp B	3	1	3	1	3	1

30.14 Usage Notes

In ATU-V, status register (TSR) flags deployed in different timers are status flags indicating occurrences of compare match, overflow, underflow, and input capture. If compare match, overflow, underflow, or input capture occurs when the status flag is set to 1, the status flag retains the value “1”, and an interrupt request is issued to the CPU. To find out whether an interrupt is requested properly, check the corresponding flag on the INTC. If two interrupt conditions occur in succession in a period shorter than a PCLK cycle, an interrupt request is generated for the first interrupt condition, and not for the subsequent one.

This section describes the conflicts and operation that may occur during ATU-V operation.

30.14.1 Input Capture Conflict Operation

30.14.1.1 Conflict between Writing to General Register and Input Capture

When a write to a general register occurs simultaneously with input capture, writing takes priority (see the left portion of **Figure 30.111** below). However, an input capture interrupt request is issued, and if input capture status is provided, the input capture flag is set. The waveforms in the right portion of **Figure 30.111** show a case in which writing occurs one PCLK cycle prior to input capture.

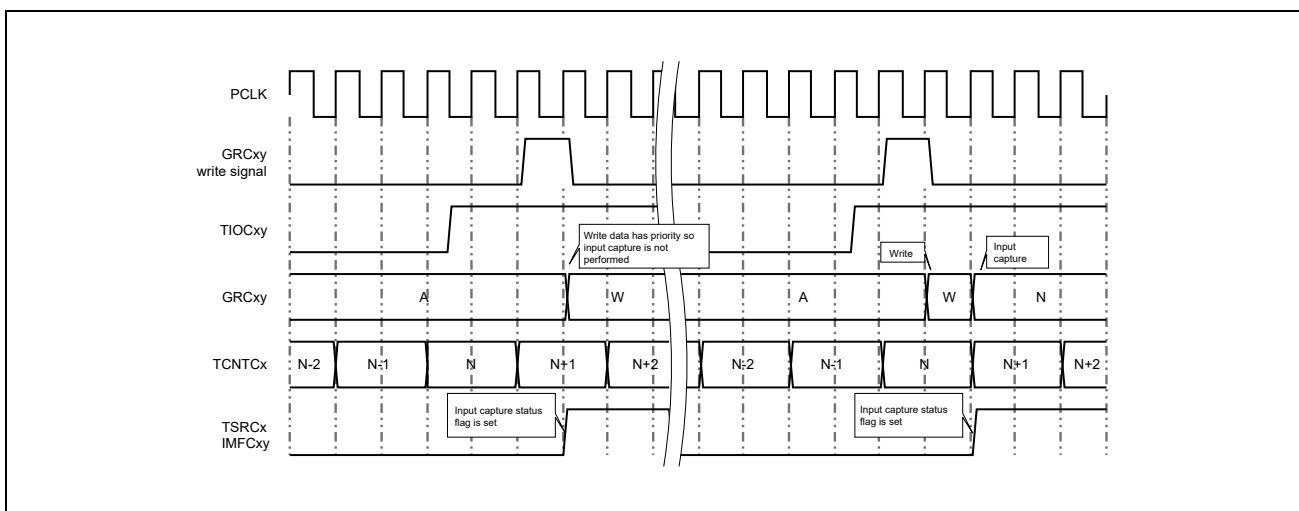


Figure 30.111 Conflict between Writing to GRCxy and Input Capture

Timer	Counter (Whose Value is Captured)	Capture Register	Status
Timer C	TCNTCx	GRCxy	IMFCxy
Timer F	ECNTAFx	GRAFx	ICFFx
	ECNTBFx	GRBFx	
	ECNTCFx	GRCFx	
	ECNTCFx + GRDFx	GRDFx	

30.14.1.2 Conflict between Writing to Counter and Input Capture

When a write to a counter occurs simultaneously with input capture, the value immediately before writing is captured (see the left portion of **Figure 30.112**). The waveforms in the right portion of **Figure 30.112** show a case in which writing occurs one PCLK cycle prior to input capture, so the written value is captured.

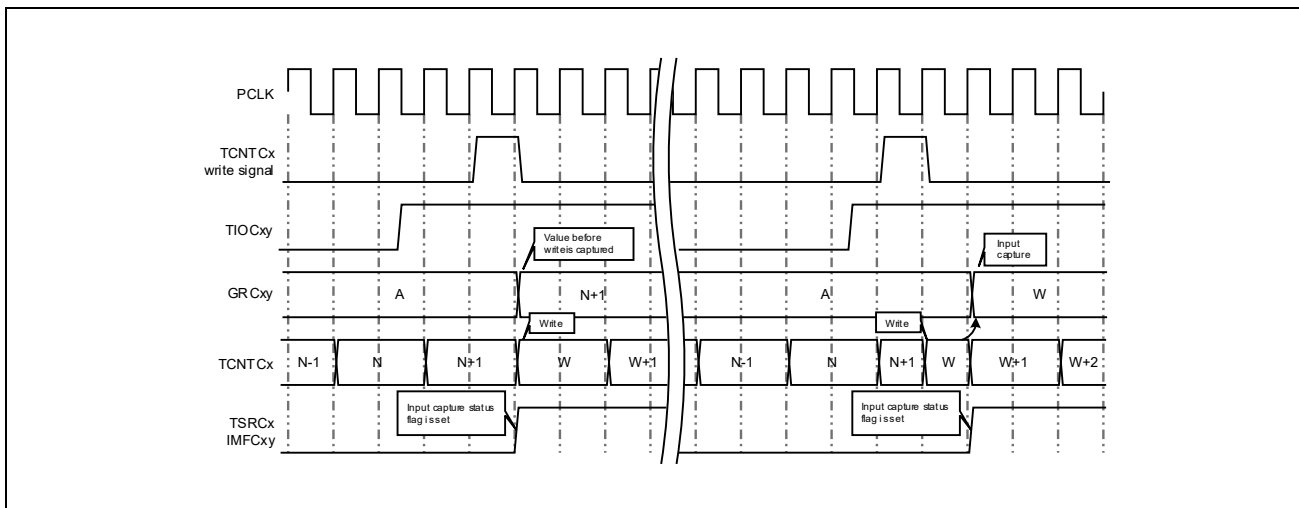


Figure 30.112 Conflict between Writing to TCNTCx and Input Capture

Timer	Counter (Whose Value is Captured)	Capture Register	Status
Timer A	TCNTAx	ICRAx	ICFAx
Timer B	TCNTB0	ICRB0	ICFB0
	TCNTB0 + ICRB1	ICRB1	
	TCNTB6	ICRB6	—
Timer C	TCNTCx	GRCxy	IMFCxy
Timer D	TCNT1Dx	OSBRDx	—
	TCNT1Dx	ICR1Dxy	
	TCNT2Dx	ICR2Dxy	—
Timer F	ECNTAFx	GRAFx	ICFFx
	ECNTBFx	GRBFx	
	ECNTCFx	GRCFx	
	ECNTCFx + GRDFx	GRDFx	

30.14.1.3 Conflict between Setting and Clearing of Input Capture Status Flag

Flag clearing by a timer status clear register takes priority over flag setting by an input capture. The left portion of **Figure 30.113** shows an example where an input capture and a flag clearing occur simultaneously, and the flag is cleared. In contrast with this, the waveforms in the right portion of **Figure 30.113** show an example in which input capture is performed immediately after the flag has been cleared.

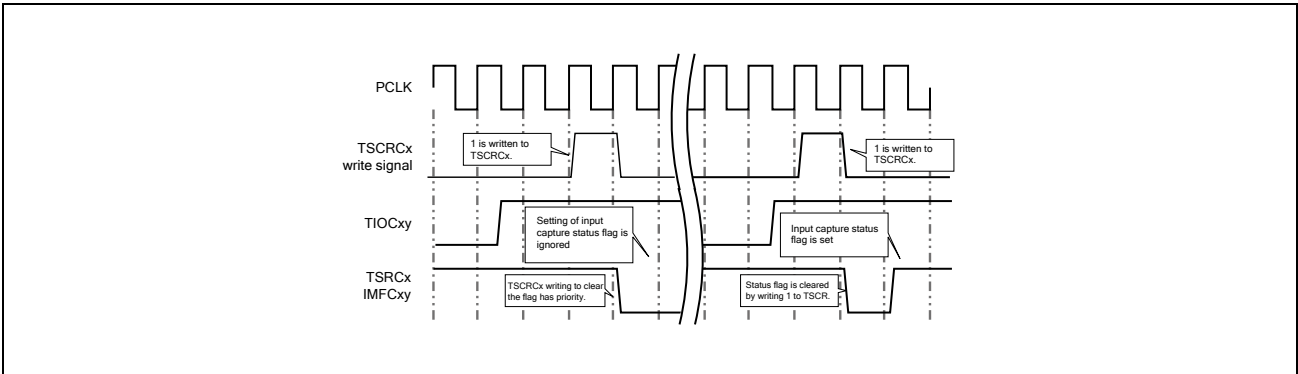


Figure 30.113 Conflict between Status Clearing by Writing 1 to Status Clear Register and Input Capture

Flag	
Timer A	ICFAx
Timer C	IMFCxy

Flag	
Timer B	ICFB0
Timer F	ICFFx

30.14.2 Compare Match Conflict Operation

30.14.2.1 Compare Match Signal in Cases of Consecutive Comparison

Matches between the counter and comparison register will take two or more PCLK cycles when the counter clock has a period of or more than two PCLK cycles. Detection of the rising edge of the compare match signal by the timers listed in the table below is handled as the compare match signal.

Timer	Counter	Compare Match Register
Timer C	TCNTCx	GRCxy, OCRCxy
Timer D	TCNT1Dx, TCNT2Dx	OCR1Dxy, RCR1Dx OCR2Dxy, RCR2Dx

30.14.2.2 Conflict between Writing to Compare Match General Register and Compare Match

A conflict between a write to a register provided with the compare match function and compare match is described below, using timer C as an example. If a write occurs after the values of GRCxy and TCNTCx match (see the left portion of **Figure 30.114**), a compare match interrupt request is issued, and the compare match status is set. If GRCxy and TCNTCx do not match in any cycles (see the right portion of **Figure 30.114**), no compare match is detected.

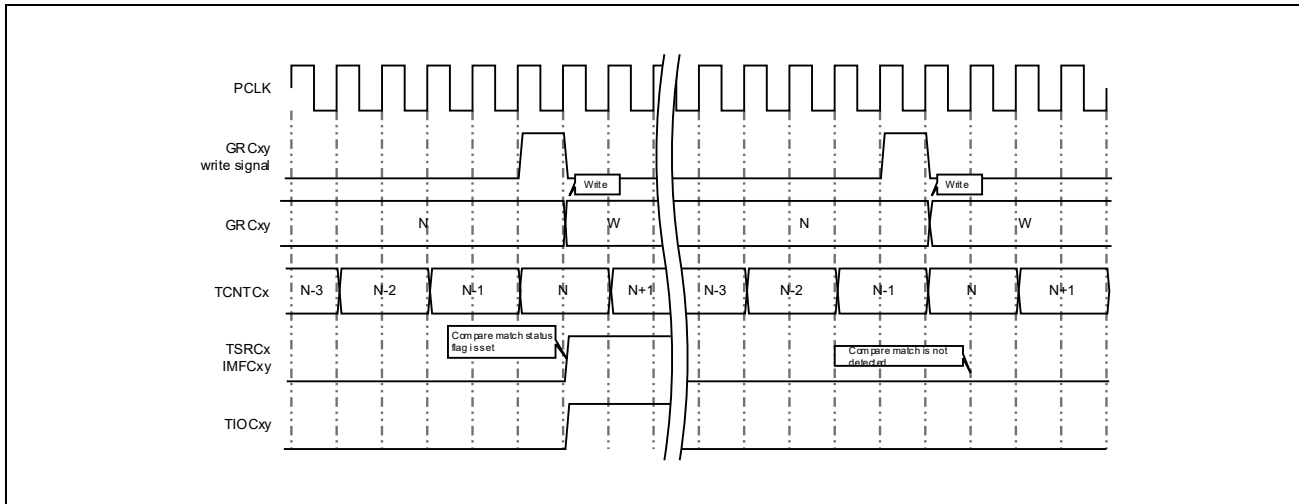


Figure 30.114 Conflict between Writing to GRCxy and Compare Match

Timer	Counter	Compare Match Register	Status
Timer B	TCNTB1	OCRB1	CMFB1
	TCNTB1	OCRB10	CMFB10
	TCNTB1	OCRB11	CMFB11
	TCNTB1	OCRB12	CMFB12
Timer C	TCNTCx	GRCxy	IMFCxy
Timer D	TCNT1Dx	OCR1Dxy	CMFADxy
	TCNT2Dx	OCR2Dxy	CMFBDxy
Timer F	ECNTAFx	GRAFx	—
	ECNTBFx	GRBFx	—
	ECNTCFx	GRBFx	OVFCFx
		GRDFx	OVFCFx
Timer G	TCNTGx	OCRGx	CMFGx

Timing of compare match B0 and compare match B6 of timer B and the cycle match of timer E differ from the timing of these compare matches. For details, see the next **Section 30.14.2.3, Conflict between CYLRExy Write and Cycle Match with TCNTExy**.

30.14.2.3 Conflict between CYLRExy Write and Cycle Match with TCNTExy

Operation when a write to CYLRExy occurs simultaneously with compare match (cycle match) with TCNTExy is shown below. As shown in the left portion of **Figure 30.115**, if a write to CYLRExy occurs at the same timing as cycle match counter clearing, TCNTExy is cleared as in the same manner as normal cycle match, a cycle match interrupt request is issued, and the cycle match status and PWM output are changed. The waveforms in the right portion of **Figure 30.115** show an example in which CYLRExy is written to before the counter is cleared. In this case, no cycle match is detected and TCNTExy continues counting up.

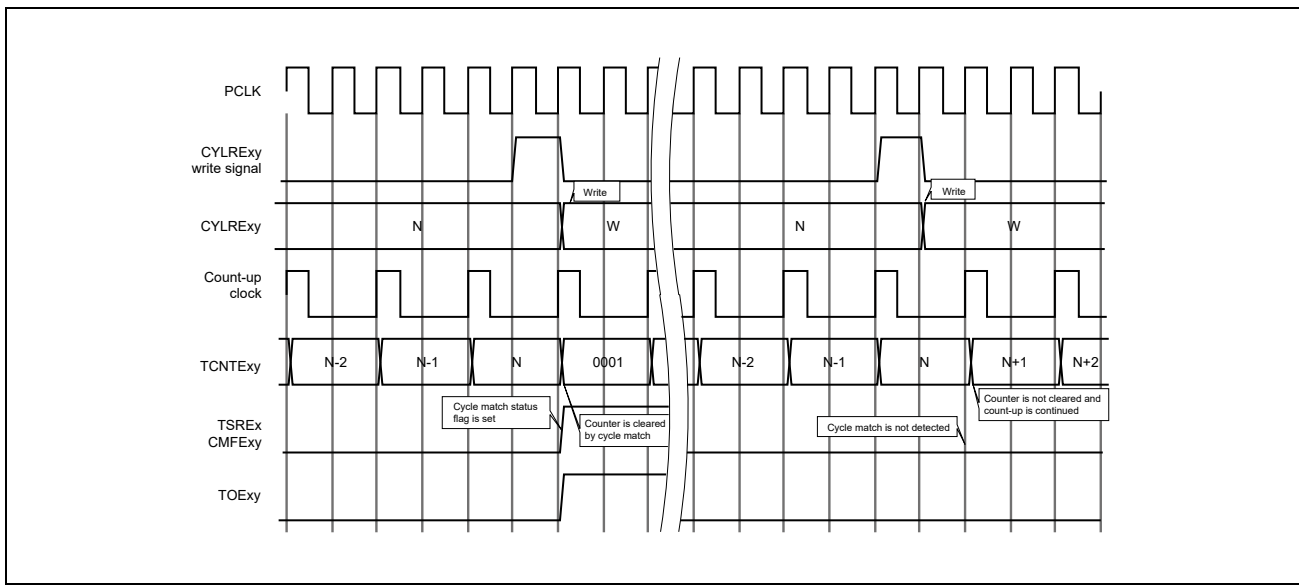


Figure 30.115 Conflict between Writing to CYLRExy and Cycle Match

Timer	Counter	Compare (Cycle) Match Register	Status
Timer B	TCNTB0	OCRB0	CMFB0
	TCNTB6	OCRB6	CMFB6
	TCNTB6M	ICRB6	CMFB6M
Timer E	TCNTExy	CYLRExy	CMFExy

30.14.2.4 Conflict between DTRExy Write and Cycle Match with TCNTExy

The operation is the same as the case of conflicts between CYLRExy writes and cycle matches with TCNTExy described above.

30.14.2.5 Conflict between Writing to Counter and Compare Match

A conflict between a write to a counter and compare match is described below. If a write occurs after the compare match register and the counter match (see the left portion of **Figure 30.116**), a compare match interrupt request is issued, and the compare match status is set. If the compare match register and the counter do not match in any cycles (see the right portion of **Figure 30.116**), a compare match interrupt request is not issued, and a compare match is not detected.

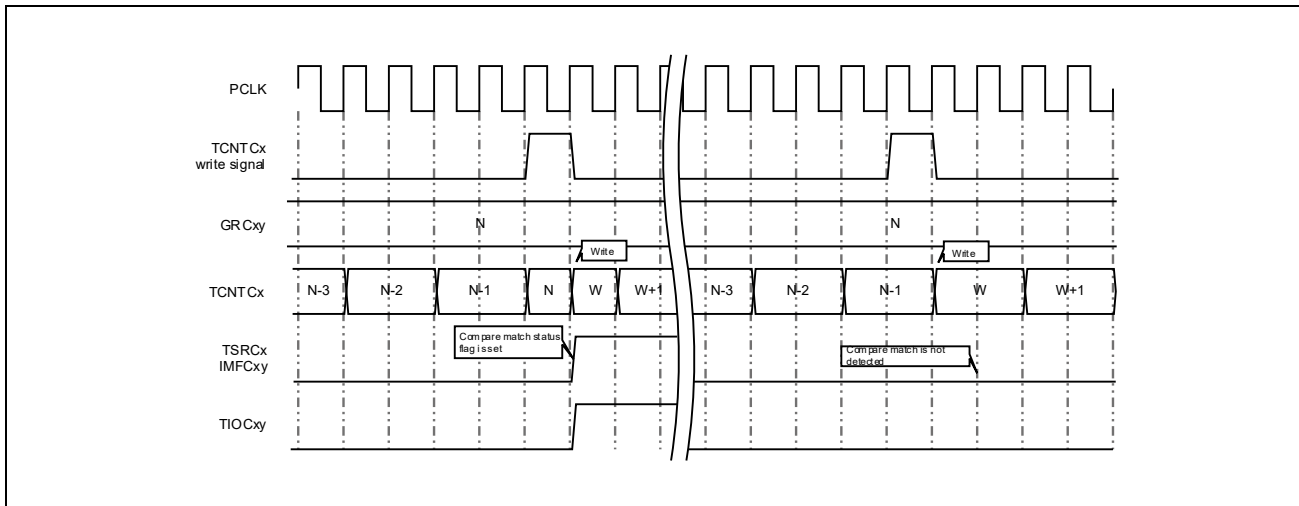


Figure 30.116 Conflict between Writing to TCNTCx and Compare Match

Timer	Counter	Compare Match Register	Status
Timer B	TCNTB1	OCRB1	CMFB1
Timer C	TCNTCx	GRCxy	IMFCxy
Timer D	TCNT1Dx	OCR1Dxy	CMFADxy
	TCNT2Dx	OCR2Dxy	CMFBDxy
Timer F	ECNTAFx	GRAFx	—
	ECNTBFx	GRBFx	—
	ECNTCFx	GRBFx	OVFCFx
		GRDFx	OVFCFx
Timer G	TCNTGx	OCRGx	CMFGx

Timing of compare match B0 and compare match B6 of timer B and the cycle match of timer E differ from the timing of these compare matches. For details, see the **Section 30.14.2.7, Conflict between Writing to TCNTExy and Counter Clearing by Cycle Match** and the **Section 30.14.2.3, Conflict between CYLRExy Write and Cycle Match with TCNTExy**, below.

30.14.2.6 Conflict between Writing to Counter and Counter Clearing by Compare Match

The waveforms shown below are for when the compare match counter clear function is enabled. If a write to a counter and counter clearing by compare match occur simultaneously, the counter is not cleared and the write takes priority (see the left portion of **Figure 30.117**). However, a compare match interrupt request is issued, and the compare match status is set. The waveforms in the right portion of **Figure 30.117** show a case in which writing to TCNTCx is delayed one PCLK cycle.

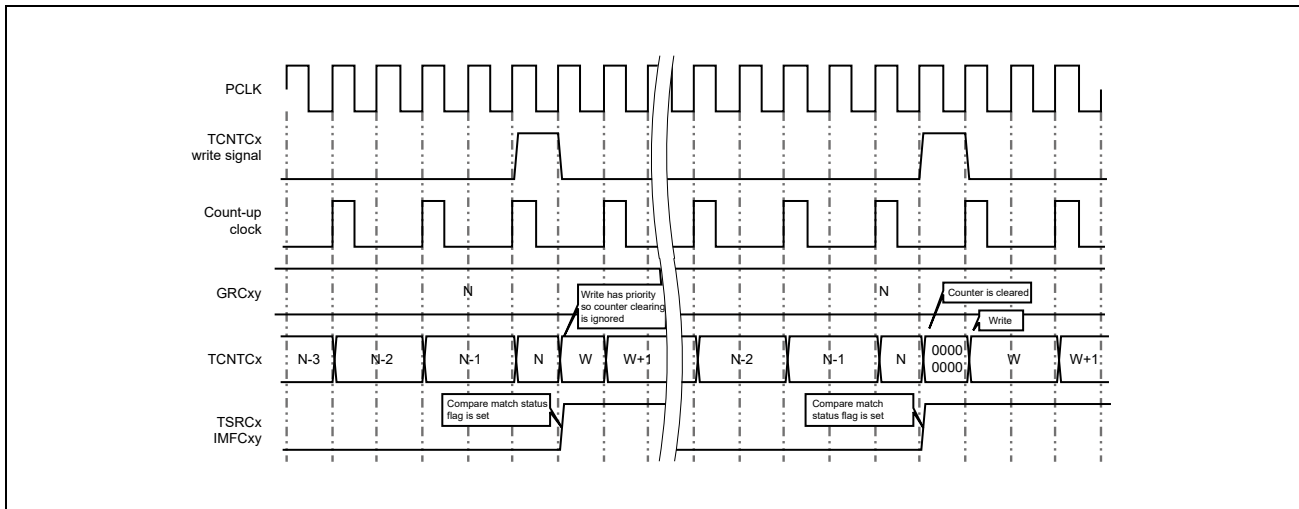


Figure 30.117 Conflict between Writing to TCNTCx and Counter Clearing by Compare Match

Timer	Counter	Compare Match Register	Status
Timer C	TCNTCx	GRCxy	IMFCxy
	TCNTCx	CUCRCx	OVFCx
Timer D	TCNT1Dx	CUCR1Dx	OVF1Dx
	TCNT2Dx	CUCR2Dx	OVF2Dx
Timer F	ECNTAFx	GRAFx	—
	ECNTBFx	GRBFx	—
Timer G	TCNTGx	OCRGx	CMFGx

30.14.2.7 Conflict between Writing to TCNTExy and Counter Clearing by Cycle Match

When a write to TCNTExy occurs simultaneously with counter clearing by cycle match, the counter is not cleared and TCNTExy is written to. However, a cycle match interrupt request is issued, the cycle match status is set, and reload of the cycle-setting register and the duty cycle setting register are performed (see the right portion of **Figure 30.118**). PWM waveform output at the time of cycle match is also performed.

The waveforms in the left portion of **Figure 30.118** show operation when writing is performed one PCLK cycle earlier than the count-up clock.

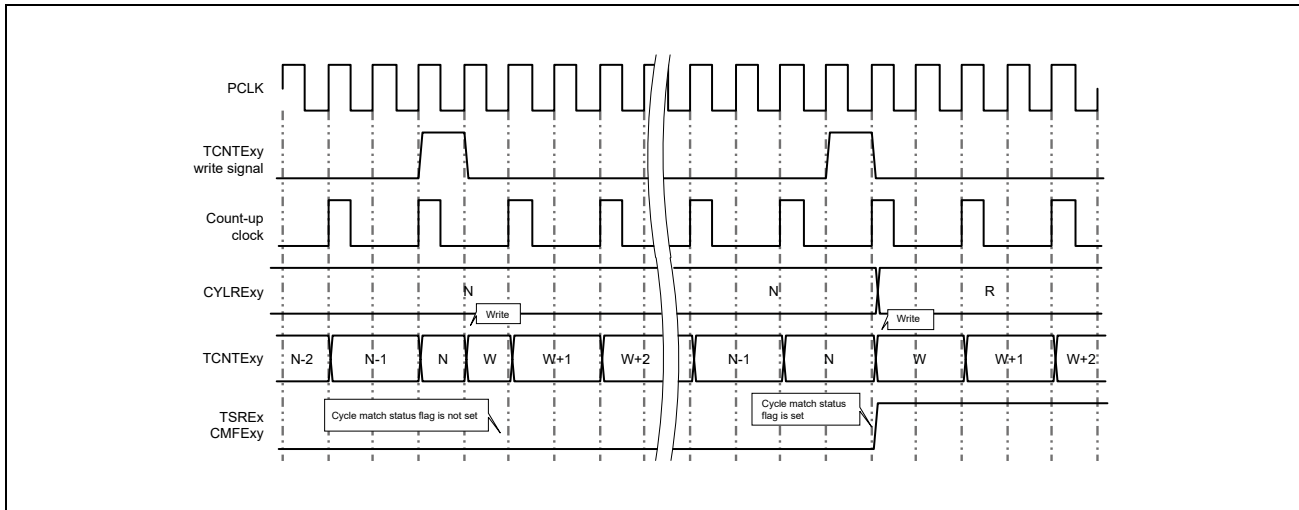


Figure 30.118 Conflict between TCNTExy Write and Counter Clearing by Cycle Match

Timer	Counter	Compare (Cycle) Match Register	Status
Timer B	TCNTB0	OCRB0	CMFB0
	TCNTB6	OCRB6	CMFB6
	TCNTB6M	ICRB6	CMFB6M
Timer E	TCNTExy	CYLRExy	CMFExy

30.14.2.8 Conflict between Setting and Clearing of Compare Match Status Flag

Flag clearing by a timer status clear register takes priority over flag setting by compare match (see the left portion of **Figure 30.119**). The right portion of **Figure 30.119** shows the case where a compare match occurs just after flag clearing.

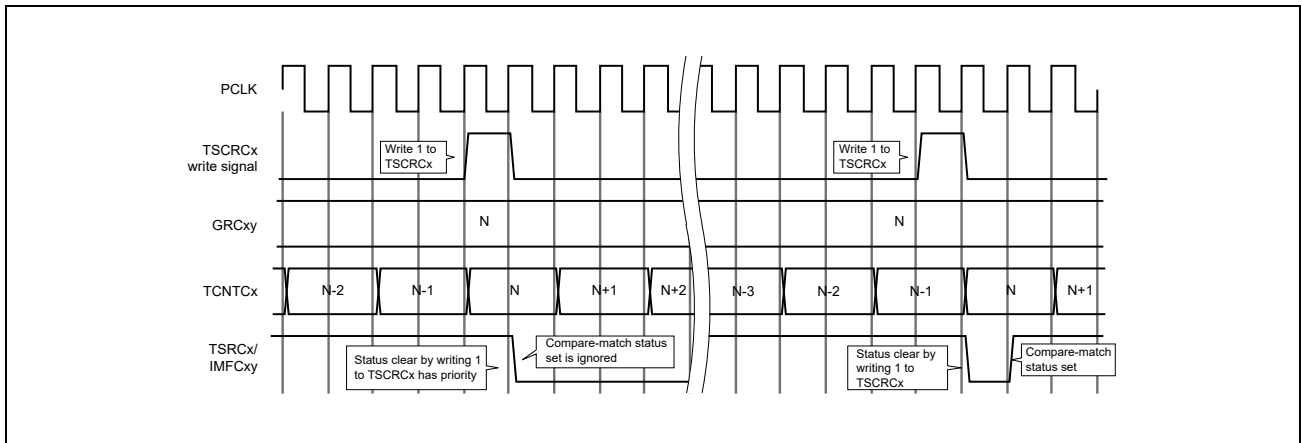


Figure 30.119 Conflict between Setting and Clearing of Compare Match Status Flag

Timer	Counter	Compare Match Register	Status
Timer B	TCNTB1	OCRB1	CMFB1
	TCNTB1	OCRB10	CMFB10
	TCNTB1	OCRB11	CMFB11
	TCNTB1	OCRB12	CMFB12
Timer C	TCNTCx	GRCxy	IMFCxy
Timer D	TCNT1Dx	OCR1Dxy	CMFADxy
	TCNT2Dx	OCR2Dxy	CMFBDxy
Timer F	TCNTCFx	GRDFx	OVFCFx
Timer G	TCNTGx	OCRGx	CMFGx

Timing of compare match B0 and compare match B6 of timer B and the cycle match of timer E differ from the timing of these compare matches. For details, see the next **Section 30.14.2.9, Conflict between Setting of Cycle Match Status Flag and Writing 1 to the Status Clear Register** and the **Section 30.14.2.3, Conflict between CYLRExy Write and Cycle Match with TCNTExy**.

30.14.2.9 Conflict between Setting of Cycle Match Status Flag and Writing 1 to the Status Clear Register

If setting of the cycle match flag (cycle match) and clearing of the flag by the timer status clear register occur simultaneously, clearing of the flag takes priority. The left portion of **Figure 30.120** shows an example where flag setting by cycle match and flag clearing by the timer status clear register occur simultaneously. The waveforms in the right portion of **Figure 30.120** show an example in which the flag is cleared one PCLK cycle earlier than flag setting.

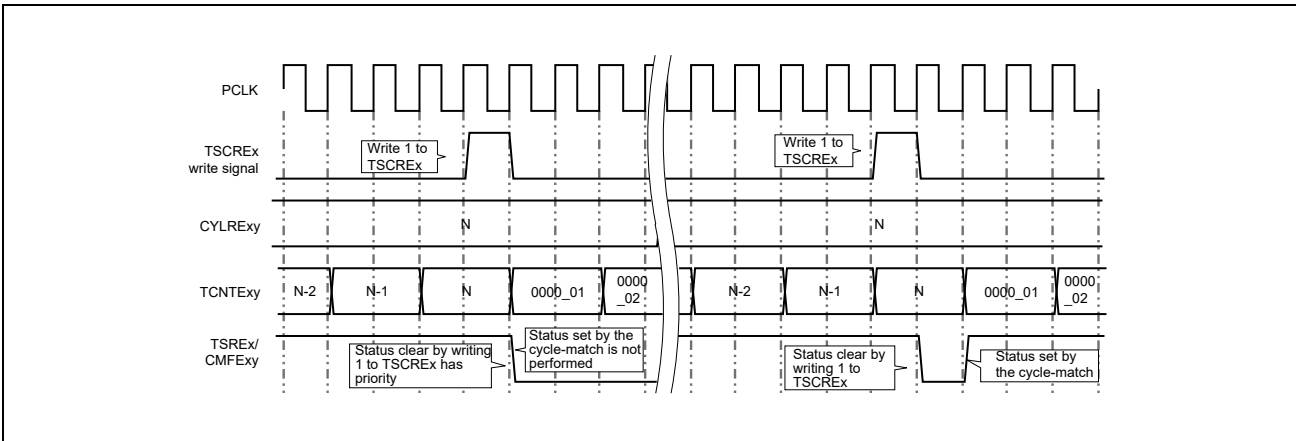


Figure 30.120 Conflict between Cycle Match and Cycle Match Status Clear by Writing 1 to the Status Clear Register

Timer	Counter	Compare (Cycle) Match Register	Status
Timer B	TCNTB0	OCRB0	CMFB0
	TCNTB6	OCRB6	CMFB6
	TCNTB6M	ICRB6	CMFB6M
Timer E	TCNTExy	CYLRExy	CMFExy

30.14.2.10 Conflict between Writing 0 to TCNTExy and Cycle Match

Operation when writing “0” to TCNTExy occurs simultaneously with cycle match is shown below. The waveforms in the left portion of **Figure 30.121** show a case in which 000000_H is written to TCNTExy at the same time TCNTExy is cleared to 000001_H due to cycle match. A cycle match interrupt request is issued and the cycle match status is set, but PWM output does not start because writing “0” takes priority. PWM output restarts when TCNTExy counts up to 000001_H. The waveforms in the middle of **Figure 30.121** show a case in which “0” is written to TCNTExy one PCLK cycle after the counter has been cleared by cycle match. Cycle match detection and PWM output are restarted at the timing when the TCNTExy counter value changes from “N” to “1”. In contrast with this, the waveforms in the right portion of **Figure 30.121** show an example in which “0” is written one PCLK before detection of cycle match. In this case, neither cycle match is detected nor PWM output restarted, and the previous state is retained.

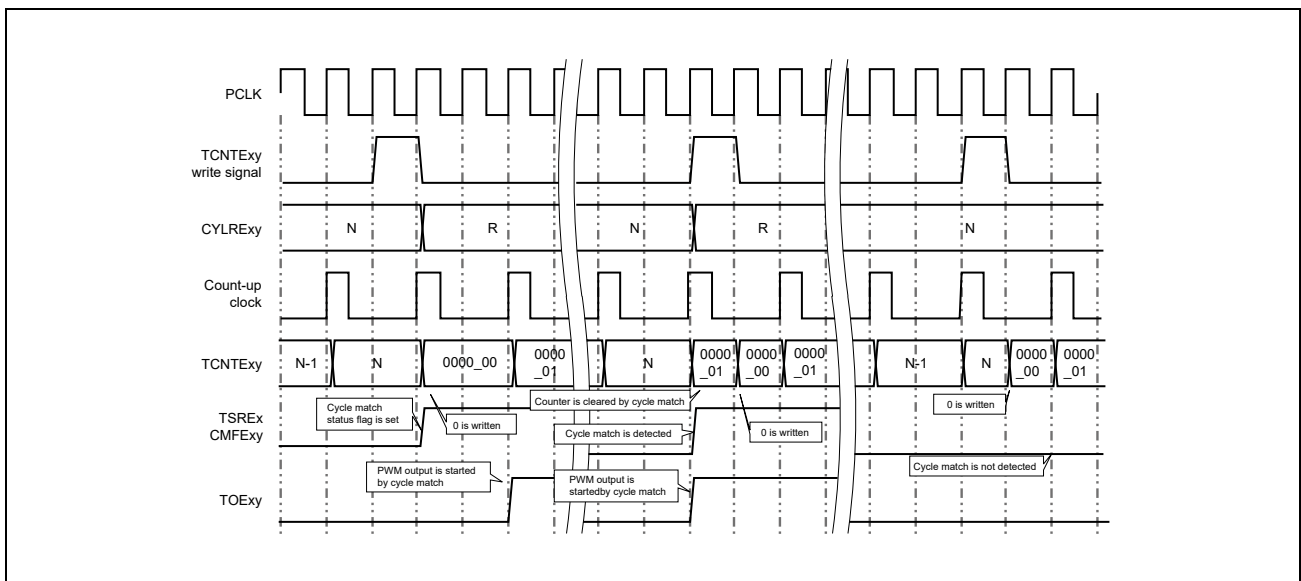


Figure 30.121 Conflict between Writing 0 to TCNTExy and Cycle Match

30.14.3 Load/Reload Conflict Operation

30.14.3.1 Conflict between Data Transfer and Writing to Transfer Destination Register

A conflict between data transfer between registers and a write to the transfer destination register is described below. When data transfer occurs simultaneously with a write to the transfer destination register, writing takes priority and the attempt of data transfer is ignored. **Figure 30.122** shows a conflict between reload to CYLRExy of timer E and a write to it. As shown by the waveforms in the left portion of the figure, if writing to CYLRExy occurs at the same timing as cycle reload, writing takes priority. The waveforms in the right portion of the figure indicate a case in which CYLRExy is written to immediately after cycle reload.

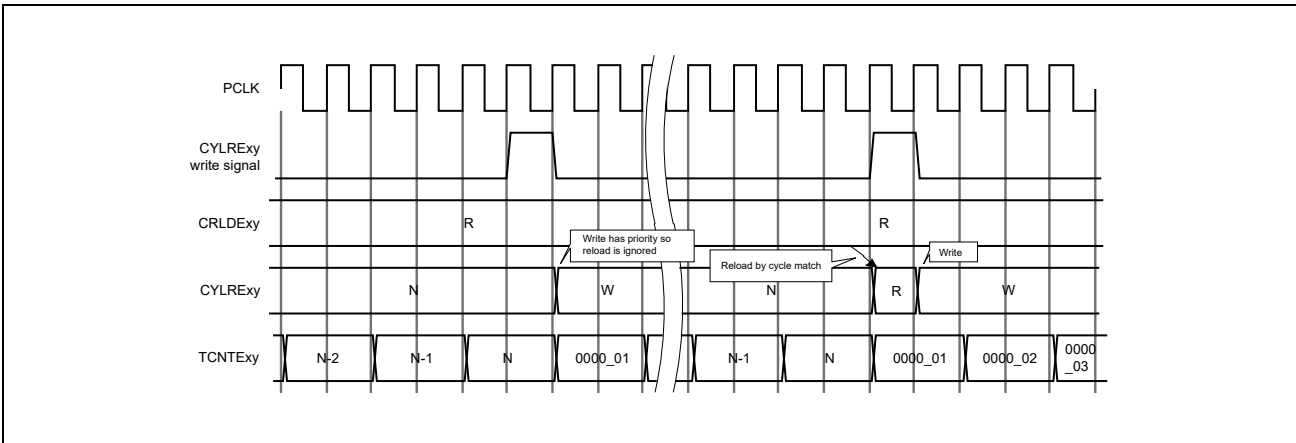


Figure 30.122 Conflict between Writing to CYLRExy and Cycle Reload

Timer	Transfer Data	Transfer Destination Register	Transfer Timing
Timer B	ICRB0	TCNTB2	External event
	LDB		
	TCNTB2 - PIMR		
	ICRB0 - PIMR	RLDB	External event
	LDB - PIMR		
Timer E	TCNTB3 + PIMR	TCNTB3	External event
	TCNTB3	TCNTB4	External event
	CRLDExy	CYLRExy	Cycle match
	DRLDExy	DTRExy	Cycle match

30.14.3.2 Conflict between Data Transfer and Writing to Transfer Source Register

A conflict between data transfer between registers and a write to the transfer source register is described below. When data transfer occurs simultaneously with a write to the transfer source register, the value prior to writing is transferred. At the same time, the value of the transfer source register is modified. **Figure 30.123** shows the operation when writing to CRLDExy occurs at the timing of cycle reload. If writing to CRLDExy occurs at the same timing as cycle reload (waveforms in the left portion of the figure), the value immediately before writing is reloaded. On the other hand, the waveforms in the right portion of the figure show an example in which CRLDExy is written to one cycle earlier than cycle reload.

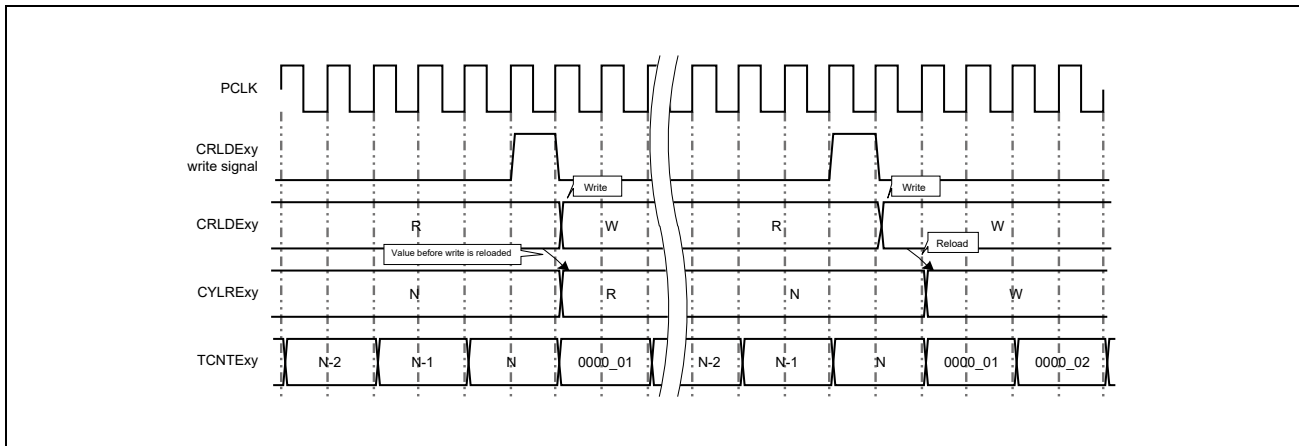


Figure 30.123 Conflict between Writing to CRLDExy and Cycle Reload

Timer	Transfer Source Register	Transferred Value	Transfer Destination Register	Transfer Timing
Timer B	LDB	LDB - PIMR	RLDB	External event
		LDB	TCNTB2	External event
	PIMR	ICRB0 - PIMR	RLDB	External event
		LDB - PIMR	TCNTB2	External event
	RLDB	TCNTB2 - PIMR	TCNTB2	External event
		TCNTB3 + PIMR	TCNTB3	External event
TCNTB3	TCNTB2 + RLDB	TCNTB2	External event	
	TCNTB3	TCNTB3	TCNTB4	External event
Timer E	CRLDExy	CRLDExy	CYLRExy	Cycle match
	DRLDExy	DRLDExy	DTRExy	Cycle match

30.14.4 Counter Conflict Operation

30.14.4.1 Conflict between Writing to Counter and Count-Up/Count-Down

When a write to a counter occurs simultaneously with incrementation/decrementation of the counter, the write operation takes priority. The attempt to increment/decrement the value is ignored and incrementation/decrementation resumes from the value written in the next counter clock.

30.14.4.2 Conflict between Count-Up and Counter Clearing

When incrementation of a counter occurs simultaneously with clearing of the counter, the counter is not cleared to “0” but cleared to “1”.

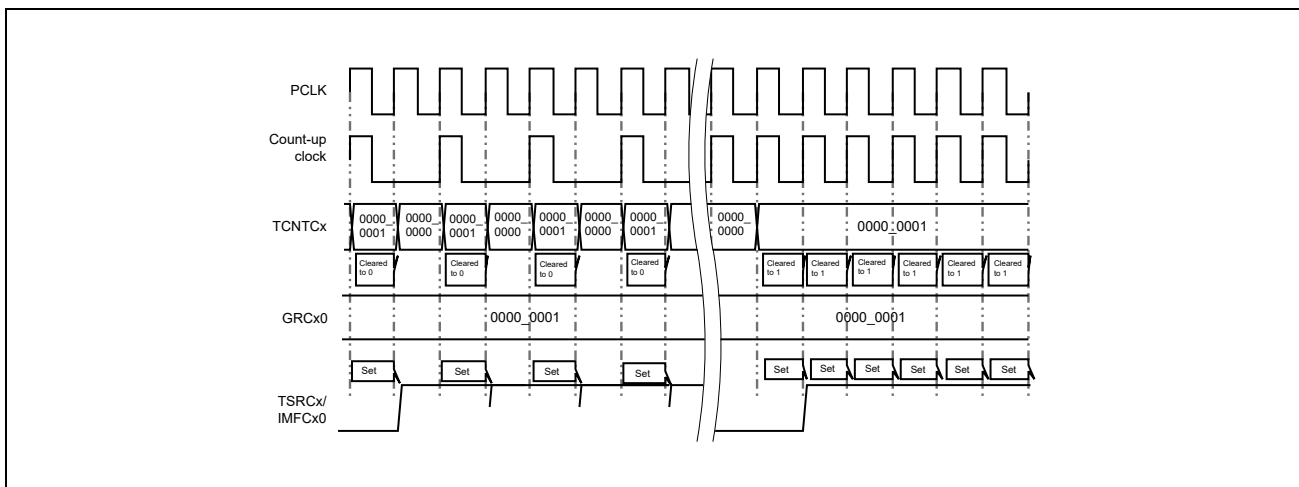


Figure 30.124 Simultaneous Occurrence of Count-Up and Counter Clearing

Timer	Counter	Compare Match Register	Remarks
Timer C	TCNTCx	GRCx0	Only when PWMx0 = “1”
	TCNTCx	CUCRCx	Only when PWMx0 = “0” and CLRCx = “1”
Timer D	TCNT1Dx	CUCR1Dx	Only when CLR1Dx = “1”
	TCNT2Dx	CUCR2Dx	Only when CLR2Dx = “1”
Timer F	ECNTAFx	GRAFx	Only when MDFx = “000”, “110”, or “111”
	ECNTBFx	GRBFx	Only when MDFx = “001”, “010”, or “100”
Timer G	TCNTGx	OCRGx	

30.14.4.3 Conflict between Writing to Counter and Overflow

When counter overflow occurs simultaneously with a write to TCNTCx, writing to TCNTCx takes priority. However, an overflow interrupt request is issued, and the overflow status is set (see the left portion of **Figure 30.125**). If the timing for writing to the counter is earlier than incrementation of the counter (waveforms in the right portion of **Figure 30.125**), the overflow status flag is not set.

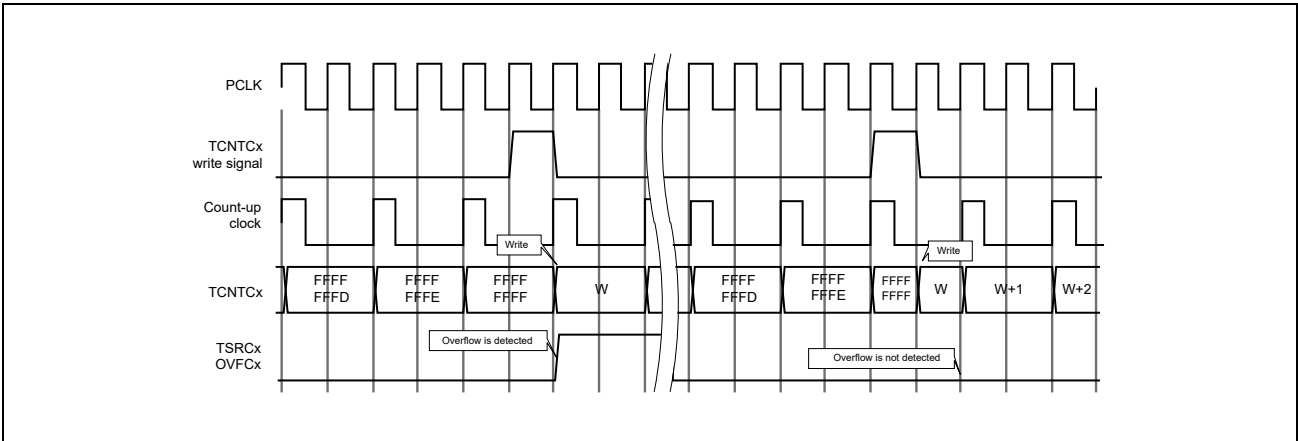


Figure 30.125 Conflict between Writing to TCNTCx and Counter Clearing on Overflow

Timer	Counter	Status
Timer A	TCNTA	OVFA
Timer C	TCNTCx	OVFCx
Timer D	TCNT1Dx	OVF1Dx
	TCNT2Dx	OVF2Dx
Timer E	TCNTExy	OVFExy
Timer F	ECNTAFx	OVFAFx
	ECNTBFx	OVFBFx
	ECNTCFx	OVFCFx

Timer	Counter	Status
Timer G	TCNTGx	OVFGx

30.14.4.4 Conflict between Setting and Clearing of Overflow Status Flag

When clearing and setting of the overflow status flag occur simultaneously, clearing takes priority. The left portion of **Figure 30.126** below shows an example where status flag setting by an overflow of the counter value (FFFF FFFF_H to 0000 0000_H) and flag clearing by the timer status clear register occur simultaneously. The waveforms in the right portion of **Figure 30.126** show the way the overflow status flag is set again immediately after the status flag has been cleared.

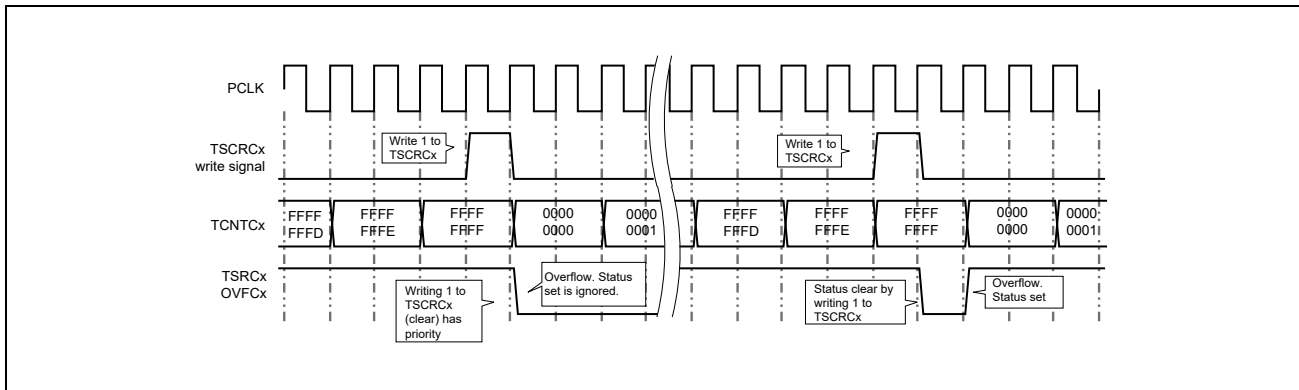


Figure 30.126 Conflict between Setting and Clearing of Overflow Status Flag

Timer	Counter	Status
Timer A	TCNTA	OVFA
Timer C	TCNTCx	OVFCx
Timer D	TCNT1Dx	OVF1Dx
	TCNT2Dx	OVF2Dx
Timer E	TCNTExy	OVFExy
Timer F	ECNTAFx	OVFAFx
	ECNTBFx	OVFBFx
	ECNTCFx	OVFCFx

Timer	Counter	Status
Timer G	TCNTGx	OVFGx

30.14.4.5 Conflict between Overflow and Counter Clearing by Compare Match

If the maximum value is set in a compare match register when the function to clear a counter by compare match is enabled, the counter is cleared when the counter reaches its maximum value. At this time, an overflow interrupt request is not issued even if the counter clock is 1/1PCLK, so an overflow is not detected.

Timer	Counter	Status	Remarks
Timer C	TCNTCx	OVFCx	Only when PWMx0 = "1"
Timer F	ECNTAFx	OVFAFx	Only when MDFx = "000", "110", or "111"
	ECNTBFx	OVFBFx	Only when MDFx = "001"
Timer G	TCNTGx	OVFGx	

If the compare match counter clear function is not provided or is disabled, an overflow interrupt request is issued and the overflow status is set. **Figure 30.127** shows operation when the PWMx bit of timer C is "1" (counter clearing is enabled) and when the PWMx0 bit is "0" (counter clearing is disabled).

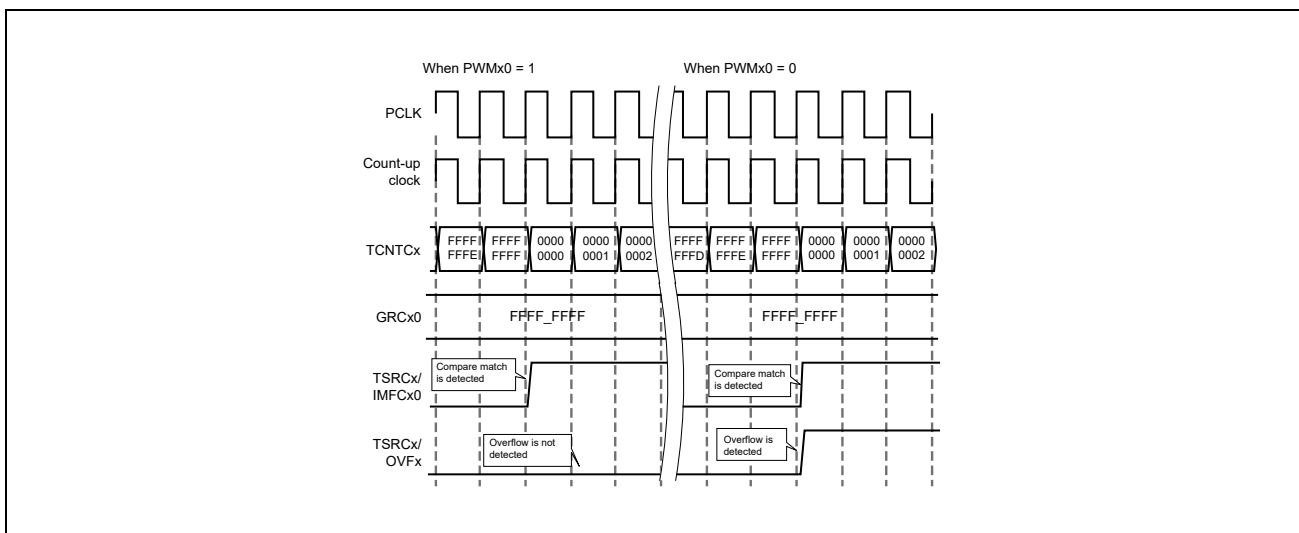


Figure 30.127 Conflict between Clearing of Compare Match Counter of Timer C and Overflow (PWMx0 = 1/0)

30.14.5 Noise Canceler Conflict Operation

This section describes conflicts in the noise cancelers.

Timer	Counter	Compare Match Register
Timer A	NCNTAx	NCRAx
Timer C	NCNTCxy	NCRCxy
Timer F	NCNTAFx	NCRAFx
	NCNTBFx	NCRBFx

30.14.5.1 Conflict between Writing to Noise Canceler Counter and Compare Match with Noise Canceler Register

When a write to NCNT occurs simultaneously with a compare match with NCR, writing takes priority. **Figure 30.128** shows an example in minimum time-at-level cancellation mode. In the left portion of the figure, since writing prevents compare match from occurring, input capture is also not performed. The right portion of the figure shows a case in which writing is performed one PCLK cycle later. In this case, compare match occurs and input capture processing is performed.

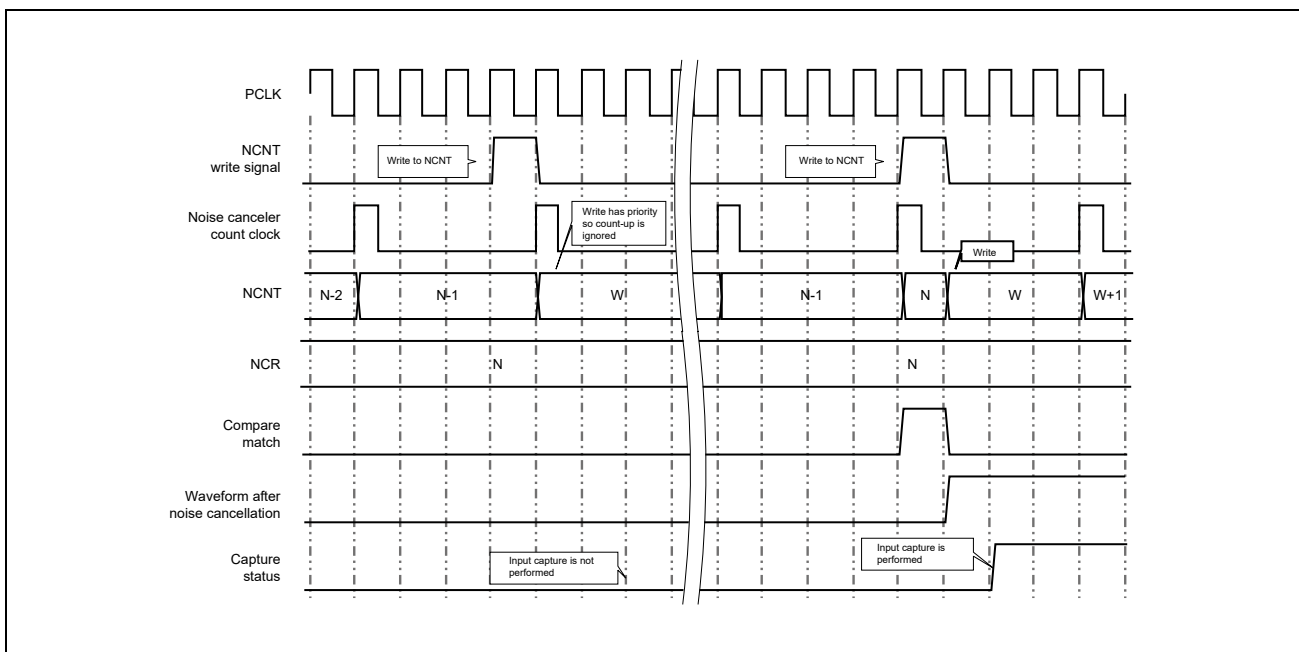


Figure 30.128 Conflict between Writing to NCNT and Compare Match of NCNT-NCR
(Example in Minimum Time-at-level Cancellation Mode)

30.14.5.2 Conflict between Writing to Noise Canceler Register and Compare Match with Noise Canceler Counter

When a write to NCR occurs simultaneously with a compare match with NCNT, writing takes priority. **Figure 30.129** shows an example in minimum time-at-level cancellation mode. In the left portion of the figure, since writing prevents compare match from occurring, input capture is also not performed. The right portion of the figure shows a case in which writing is performed one PCLK cycle later. In this case, compare match occurs and input capture processing is performed.

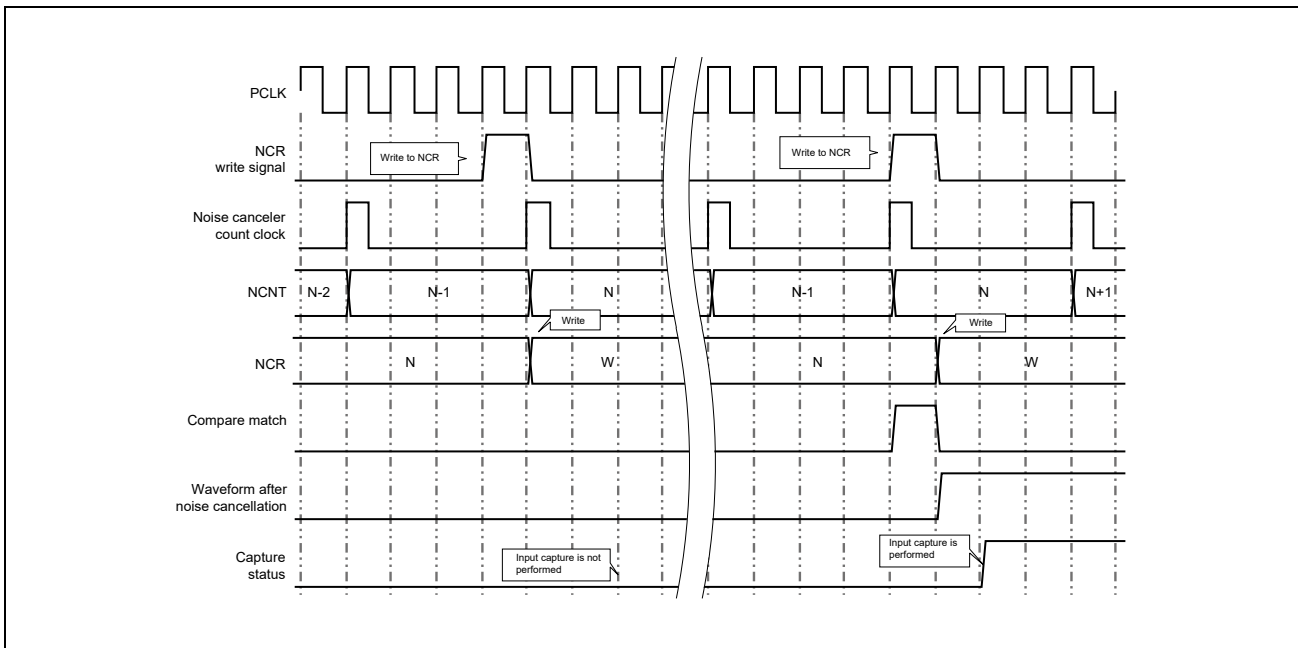


Figure 30.129 Conflict between Writing to NCR and Compare Match of NCR-NCNT (Example in Minimum Time-at-level Cancellation Mode)

30.14.6 Timer Down Counter Dxy Conflict Operation

This section describes conflicts in DCNTDxy.

30.14.6.1 Conflict between Writing to DCNTDxy Counter and Count-Down

When a write occurs simultaneously with decrementation of the down counter, writing to DCNTDxy is performed. The attempt to decrement the value is ignored and decrementation restart at the next count-down clock.

30.14.6.2 Conflict between Writing to DCNTDxy Counter and Underflow

When a write to DCNTDxy occurs simultaneously with underflow, writing to DCNTDxy is performed. The left portion of **Figure 30.130** shows operation when a count-down clock is input simultaneously with a write to DCNTDxy when the DCNTDxy value is 0000 0000_H. Though a new value is written to DCNTDxy, count-down operation is halted because underflow is detected. The underflow status flag is set. The right portion of **Figure 30.130** shows a case where data is written to DCNTDxy one cycle before, in which case an underflow interrupt request is not issued and underflow is not detected.

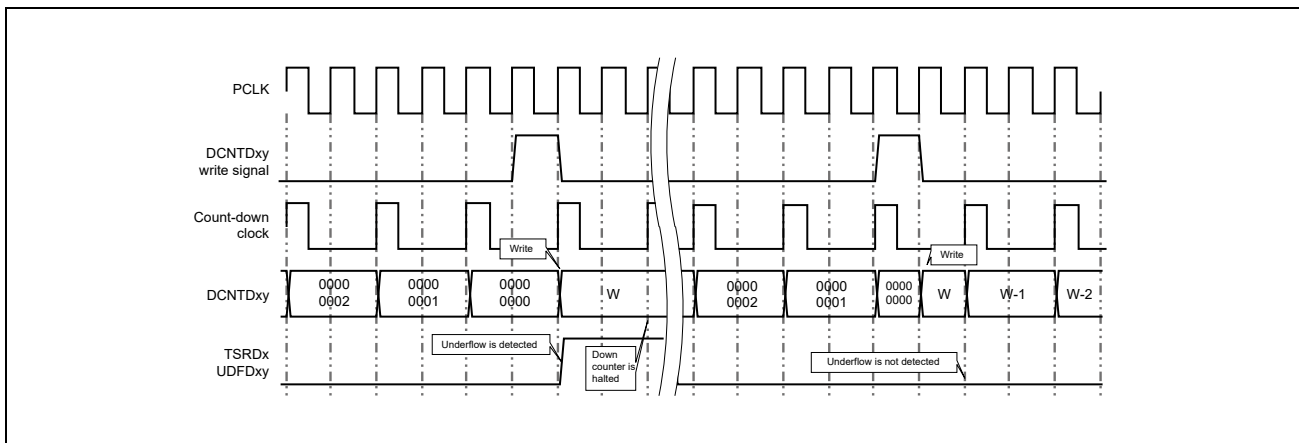


Figure 30.130 Conflict between Writing to DCNTDxy and Underflow

30.14.6.3 Conflict between Writing to DCNTDxy Counter and Compare Match B (Counter Stop Trigger)

When a write to DCNTDxy occurs simultaneously with compare match B, writing to DCNTDxy is performed (if compare match B is selected as a condition to stop the down counter).

The middle portion of **Figure 30.129** below shows an example in which writing to DCNTDxy occurs simultaneously with clearing of the counter by detection of compare match B. Counter clearing of DCNTDxy by compare match B is ignored, and writing takes priority. However, the output on TODxyB is turned off due to compare match B, and DCNTDxy halts with the written value retained. The waveforms in the right portion of **Figure 30.131** show a case in which the write cycle occurs one PCLK cycle earlier. During the PCLK cycle next to writing to DCNTDxy, the counter is cleared by compare match B. The waveforms in the left portion of **Figure 30.131** show an example in which writing is performed immediately after the counter has been cleared by compare match B.

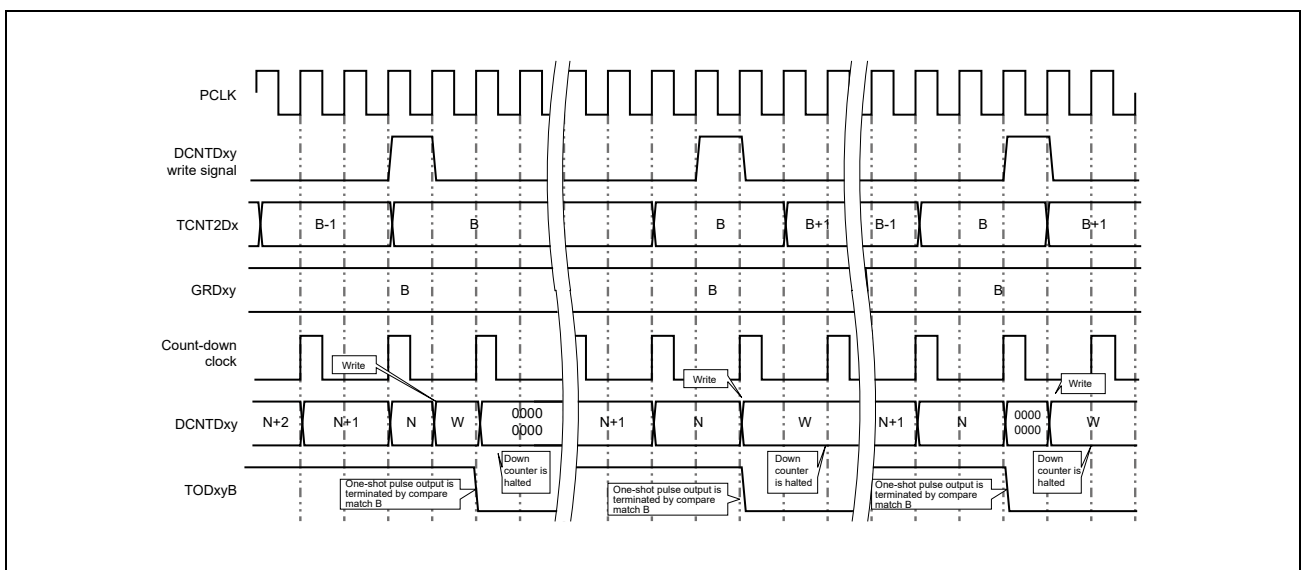


Figure 30.131 Conflict between Writing to DCNTDxy and Counter Clearing by Compare Match B

30.14.6.4 Conflict between Setting of Underflow Status Flag and Clearing by Writing 1 to Status Clear Register

When flag clearing by the timer status clear register and an underflow occurs simultaneously, flag clearing is performed. The left portion of **Figure 30.132** below shows an example where status flag setting by an underflow of DCNTDxy and flag clearing by the timer status clear register occur simultaneously. The waveforms in the right portion of **Figure 30.132** show how the status flag is set again by underflow occurrence immediately after the status flag has been cleared.

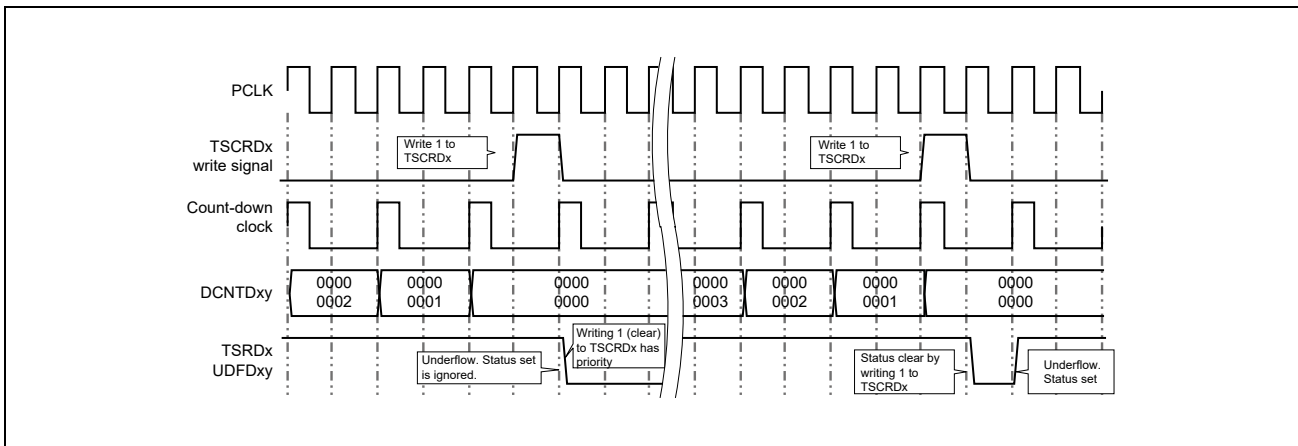


Figure 30.132 Conflict between Setting and Clearing of Underflow Status Flag

30.14.6.5 TODxyB Output upon Occurrence of Down Counter Start Trigger When Down Counter Value is 0000 0000_H

The TODxyB output is not started by down counter underflow and is terminated.

30.14.6.6 TODxyB Output at Simultaneous Occurrence of Start Trigger and Stop Trigger for Down Counter

When the down counter start trigger occurs simultaneously with the down counter stop trigger, the down counter is cleared to “0” by the stop trigger. In this case, the TODxyB output is terminated.

Similarly, if the down counter start trigger occurs and then the down counter stop trigger occurs before the first count-down clock is input, the down counter is cleared to “0” without being decremented even once and the TODxyB output is terminated.

30.14.6.7 Conflict between Down Counter Start Trigger and Underflow

When the down counter start trigger occurs simultaneously with underflow, DCNTDxy remains halted at the value of 0000 0000_H (see the waveforms in the middle of **Figure 30.133**). If the down counter had been in process of decrementing, the TODxyB output is turned off by underflow (see the waveforms in the left portion of **Figure 30.133**). If compare match A occurs while the down counter is halted (DCNTDxy = 0000 0000_H), the TODxyB output is kept negated (see the waveforms in the right portion of **Figure 30.133**). In both cases, the underflow interrupt request and setting of the underflow flag to 1 are performed upon detection of a down count start trigger or at the first cycle of the down count clock after detection.

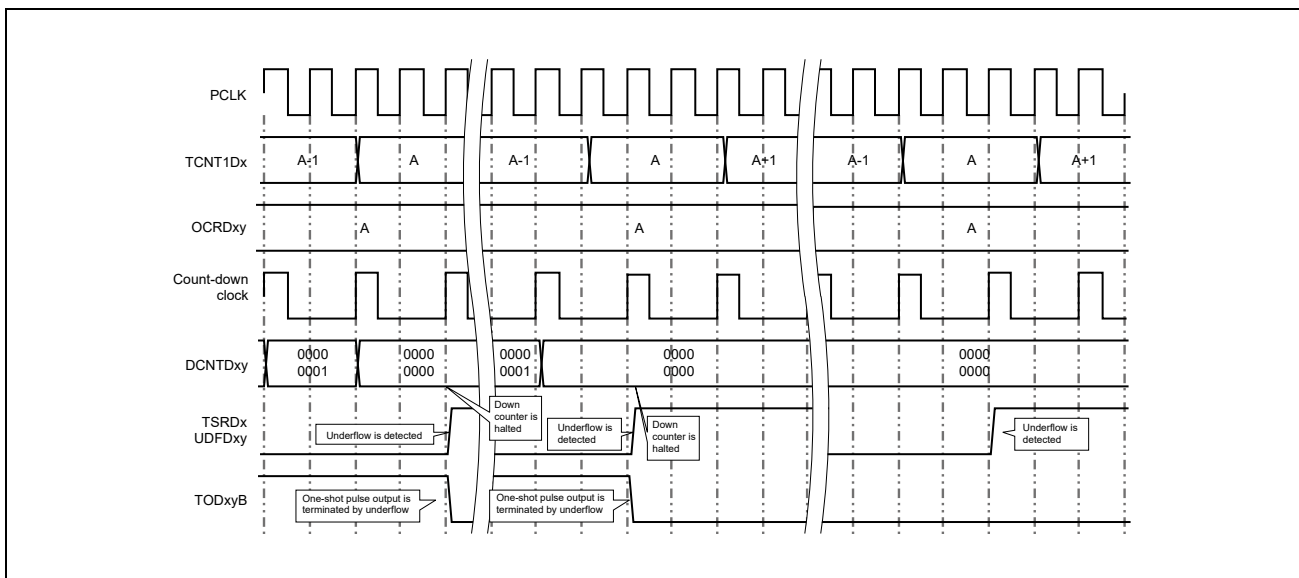


Figure 30.133 Conflict between Compare Match A and Underflow

30.14.7 Coordinated Operation of Timers A, B, and D

This section describes the conflict between counter clear requests from timer B and TCNT1Dx and TCNT2Dx.

30.14.7.1 Conflict Between Counter Clearing of TCNT1Dx and TCNT2Dx and Compare Match

Figure 30.134 shows the operation when clearing of the TCNT1Dx/TCNT2Dx counter by a counter clearing signal from timer B occurs simultaneously with compare match. The waveforms in the left portion of the figure show a case in which the counter is cleared prior to compare match. On the other hand, the waveforms in the right portion of the figure show a case in which the counter is cleared simultaneously with compare match.

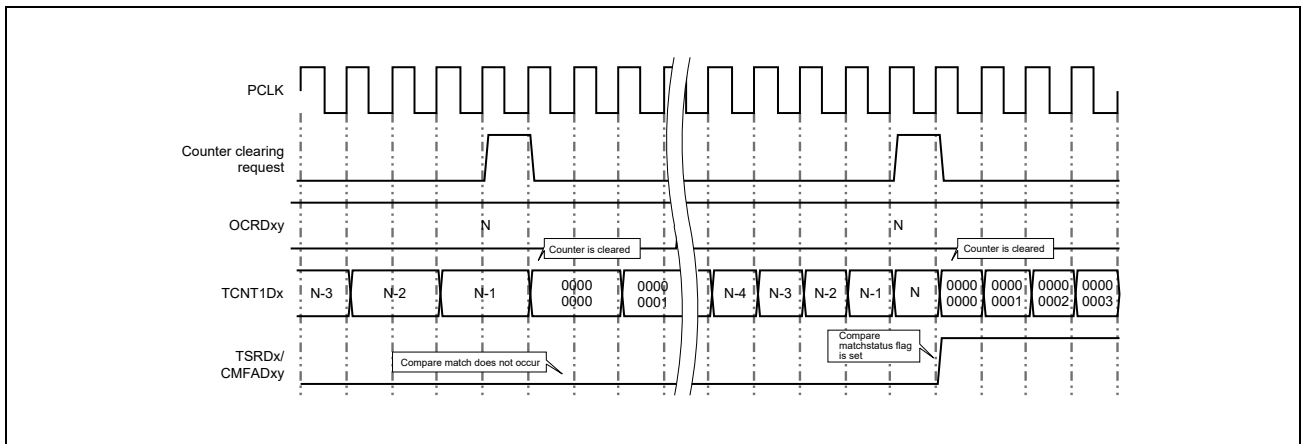


Figure 30.134 Conflict between Counter Clearing by Timer B and Compare Match

Timer	Counter	Counter Clearing Source	Compare Match Register	Status
Timer D	TCNT1Dx	TCNT1Dx/TCNT2Dx clearing request from timer B	OCR1Dxy	CMFADxy
	TCNT2Dx		OCR2Dxy	CMFBDxy

30.14.7.2 Conflict between Writing to TCNT1Dx/TCNT2Dx Counter and Counter Clearing by Timer B

When a write to TCNT1Dx/TCNT2Dx occurs simultaneously with a counter clearing signal from timer B, the counter is not cleared but writing to the counter is performed (see the waveforms in the left portion of **Figure 30.135**). The waveforms in the right portion of **Figure 30.135** show a case in which writing to TCNT1Dx is one PCLK cycle later. This is the same for TCNT2Dx.

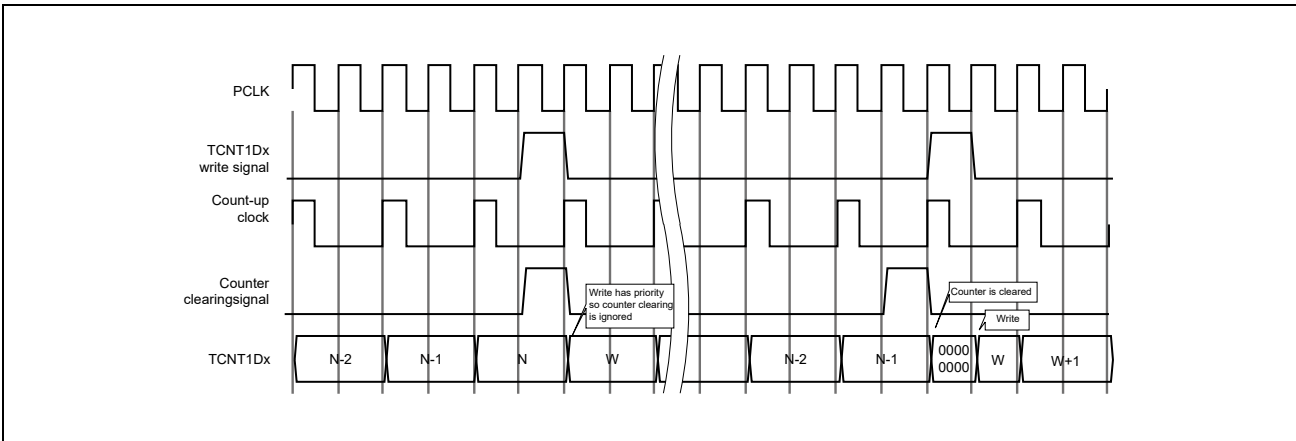


Figure 30.135 Conflict between Writing to TCNT1Dx and Counter Clearing

30.14.7.3 Conflict between TCNT1Dx/TCNT2Dx Counter Overflow and Counter Clearing by Timer B

When TCNT1Dx overflow by the counting up occurs simultaneously with clearing of TCNT1Dx from timer B, the counter value is cleared to 0000 0000_H by the counter clearing signal. In this case, an overflow interrupt request is not issued and the overflow status flag is not set (only for when C1CEDx = “1”). The same applies to TCNT2Dx overflow.

Counter clearing upon match with the value of the upper limit setting register and counter clearing by the signal from timer B must be used exclusively of each other. But if the two events erroneously occur at the same time, the counter gives priority to counter clearing and the signal from timer B, and its value is cleared to 0000 0000_H. In the case of overflow, match with the value of the upper limit setting register takes priority, and interrupt generation and flag setting are performed. The same applies to TCNT2Dx.

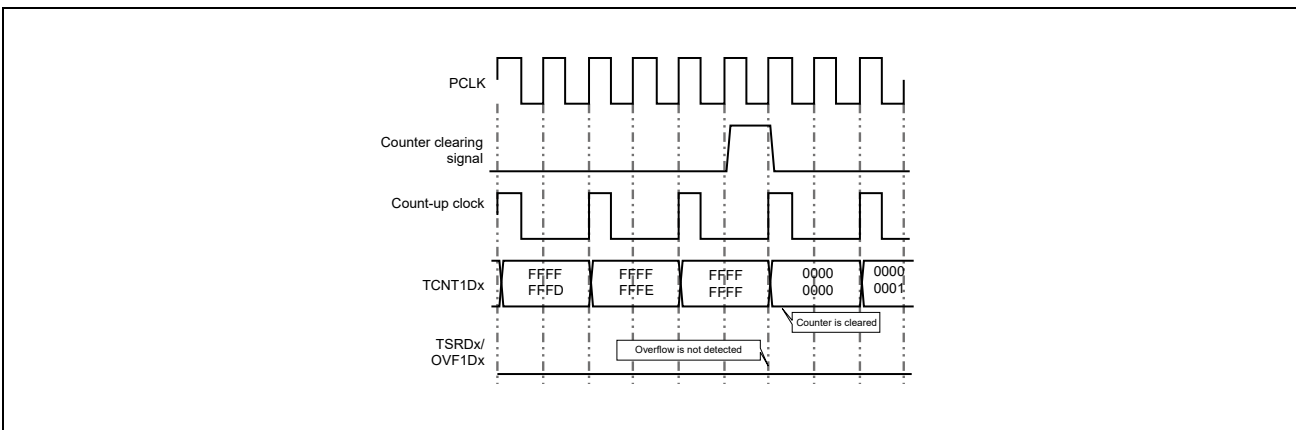


Figure 30.136 Conflict between Counter Clearing and Overflow

30.14.7.4 Conflict between TCNT1Dx Clearing by Clear Signal from Timer B and Input Capture to OSBRDx

Figure 30.137 shows the operation when clearing of the TCNT1Dx counter by timer B occurs simultaneously with input capture to an offset base register. If input capture and counter clearing occur simultaneously, the counter value before clearing is captured in OSBRDx and the TCNT1Dx counter is cleared at the same time (see the waveforms in the left portion of the figure). The waveforms in the right portion of the figure show a case in which input capture is performed one PCLK cycle after the counter has been cleared, and the counter value after clearing is captured in OSBRDx.

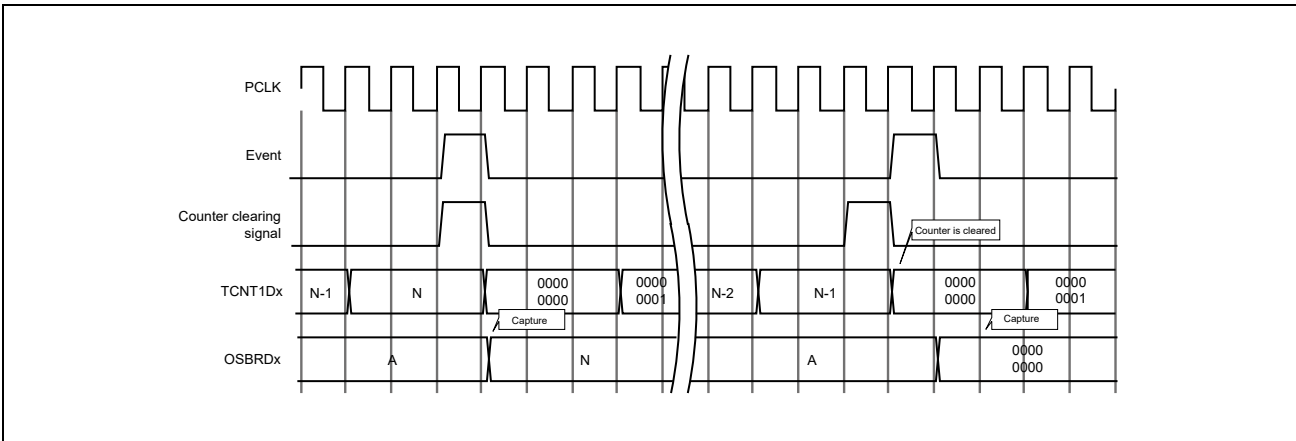


Figure 30.137 Conflict between TCNT1Dx Counter Clearing and Input Capture to OSBRDx

Section 31 Generic Timer Module (GTM)

Generic timer module (GTM) is a modular timer unit implementing common timer functionality for capture/compare of external signals and generation of complex output waveforms. It also implements complex angle clock to support powertrain control applications.

31.1 GTM Features

31.1.1 IP Version

The table below shows GTM-IP version and GTM-IP configuration included in this device.

Table 31.1 IP Version

Product	IP	Device	Revision
RH850/E2x-FCC1	GTM-IP	GTM-IP_355	v3.1.5.1
RH850/E2M	GTM-IP	GTM-IP_353	v3.1.5.1

31.1.2 Number of Sub-Units and Channels

The table below shows configuration.

Table 31.2 Sub-Units and Channels (1/2)

RH850/E2x-FCC1			
Submodule	Number of Instances	Instance Name	Channels
ARU	1	—	—
BRC	1	—	—
PSM (FIFO, AFD, F2A)	2	—	—
CMU	1	—	—
CCM	9	—	—
TBU	1	—	4 (0 to 3)
TIM	8	TIM0 to TIM7	8 (0 to 7)
TOM	5	TOM0 to TOM4	16 (0 to 15)
ATOM	9	ATOM0 to ATOM8	8 (0 to 7)
DTM	22	CDTM0_DTM[i] (i = 0, 1, 4, 5) CDTM1_DTM[i] (i = 0, 1, 4, 5) CDTM2_DTM[i] (i = 0, 1, 4, 5) CDTM3_DTM[i] (i = 0, 1, 4, 5) CDTM4_DTM[i] (i = 0, 1, 4, 5) CDTM5_DTM[i] (i = 4) CDTM6_DTM[i] (i = 4)	4 (0 to 3)
MCS	7	MCS0 to MCS6	8 (0 to 7)
MCFG	1	—	—
MAP	1	—	—
DPLL	1	—	—
SPE	4	—	—
ICM	1	—	—
CMP	1	—	—
MON	1	—	—

Table 31.2 Sub-Units and Channels (2/2)

RH850/E2M			
Submodule	Number of Instances	Instance Name	Channels
ARU	1	—	—
BRC	1	—	—
PSM (FIFO, AFD, F2A)	2	—	—
CMU	1	—	—
CCM	8	—	—
TBU	1	—	4 (0 to 3)
TIM	8	TIM0 to TIM7	8 (0 to 7)
TOM	4	TOM0 to TOM3	16 (0 to 15)
ATOM	8	ATOM0 to ATOM7	8 (0 to 7)
DTM	16	CDTM0_DTM[i] (i = 0, 1, 4, 5) CDTM1_DTM[i] (i = 0, 1, 4, 5) CDTM2_DTM[i] (i = 0, 1, 4, 5) CDTM3_DTM[i] (i = 4, 5) CDTM4_DTM[i] (i = 4, 5)	4 (0 to 3)
MCS	7	MCS0 to MCS6	8 (0 to 7)
MCFG	0	—	—
MAP	1	—	—
DPLL	1	—	—
SPE	2	—	—
ICM	1	—	—
CMP	1	—	—
MON	1	—	—

Table 31.3 Memory Size (1/2)

RH850/E2x-FCC1	
Submodule	Size
MCS RAM0	8 KB / instance
MCS RAM1	4 KB / instance
DPLL RAM1A	0.375 KB
DPLL RAM1B/C	1.125 KB
DPLL RAM2	12 KB
FIFO RAM	3.625 KB / instance

Table 31.3 Memory Size (2/2)

RH850/E2M	
Submodule	Size
MCS RAM0 (Cluster 0 to 1)	8 KB / instance
MCS RAM1 (Cluster 0 to 1)	4 KB / instance
MCS RAM0 (Cluster 2 to 6)	4 KB / instance
MCS RAM1 (Cluster 2 to 6)	2 KB / instance
DPLL RAM1A	0.375 KB
DPLL RAM1B/C	1.125 KB
DPLL RAM2	6 KB
FIFO RAM	3.625 KB / instance

31.1.3 Register/RAM Base Address

GTM base addresses are listed in the following table. GTM register and RAM addresses are given as offsets from the base addresses in general.

Table 31.4 Register and RAM Base Addresses

Base Address Name	Base Address	Bus Group
<GTM0_base>	FF80 0000 _H	Peripheral Group 2H
<GTM0_1_base>	FF90 0000 _H	Peripheral Group 2H

31.1.4 Clock Supply

GTM clock supplies are listed in the following table.

Table 31.5 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
GTM Cluster 0 to 1 (GTM cluster with DPLL core, MCS0-1 and associated RAM)	GTM Main Clock	CLK_UHSB Switchable between 1/1 (high) and 1/2 (low) frequency.
	Register access clock	CLK_HSB
GTM Cluster 2 to 8*1 (Other GTM cores and RAM)	GTM Main Clock	CLK_UHSB 1/2 (low) frequency only.
	Register access clock	CLK_HSB

Note 1. E2M: GTM Cluster 2 to 7

31.1.5 Interrupt Requests

Mapping of ICM outputs to interrupts and DMAC triggers is shown in the following table.

Table 31.6 Interrupt Requests (1/3)

GTM Interrupt Factors	Description	Interrupt Group	DMAC Trigger Group
GTM_AEI_IRQ	AEI Shared interrupt	GTM_INT_10	—
GTM_ARU_IRQ0	ARU_NEW_DATA0 Interrupt	GTM_INT_11	GTM_DMA_9
GTM_ARU_IRQ1	ARU_NEW_DATA1 Interrupt	GTM_INT_12	GTM_DMA_9
GTM_ARU_IRQ2	ARU_ACC_ACK Interrupt	GTM_INT_13	GTM_DMA_9
GTM_BRC_IRQ	BRC Shared interrupt	GTM_INT_14	GTM_DMA_9
GTM_CMP_IRQ	CMP Shared interrupt	GTM_INT_15	GTM_DMA_9
GTM_SPE0_IRQ	SPE0 Shared interrupt	GTM_INT_0	GTM_DMA_0
GTM_SPE1_IRQ	SPE1 Shared interrupt	GTM_INT_1	GTM_DMA_1
GTM_SPE2_IRQ*1	SPE2 Shared interrupt	GTM_INT_2	GTM_DMA_2
GTM_SPE3_IRQ*1	SPE3 Shared interrupt	GTM_INT_3	GTM_DMA_3
GTM_PSM0_IRQ[x]	PSM0 Shared interrupts (x: 0...7)	GTM_INT_8	GTM_DMA_9
GTM_PSM1_IRQ[x]	PSM1 Shared interrupts (x: 0...7)	GTM_INT_8	GTM_DMA_9
GTM_DPLL_IRQ0	DPLL_DCGI: DPLL direction change interrupt	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ1	DPLL EDI: DPLL enable or disable interrupt	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ2	DPLL_TINI: DPLL TRIG. min. hold time (THMI) viol. detected	GTM_INT_9	—
GTM_DPLL_IRQ3	DPLL_TAXI: DPLL TRIG. max. hold time (THMA) viol. detected	GTM_INT_9	—
GTM_DPLL_IRQ4	DPLL_SISI: DPLL STATE inactive slope detected	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ5	DPLL_TISI: DPLL TRIGGER inactive slope detected	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ6	DPLL_MSI: DPLL Missing STATE interrupt	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ7	DPLL_MTI: DPLL Missing TRIGGER interrupt	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ8	DPLL_SASI: DPLL STATE active slope detected	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ9	DPLL_TASI: DPLL TRIG. active slope detected while NTI_CNT is 0	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ10	DPLL_PWI: DPLL Plausibility window (PVT) viol. int. of TRIG.	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ11	DPLL_W2I: DPLL Write access to RAM region 2 interrupt	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ12	DPLL_W1I: DPLL Write access to RAM region 1b or 1c int.	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ13	DPLL_GL1I: DPLL Get of lock interrupt for SUB_INC1	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ14	DPLL_LL1I: DPLL Lost of lock interrupt for SUB_INC1	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ15	DPLL_EI: DPLL Error interrupt	GTM_INT_9	—
GTM_DPLL_IRQ16	DPLL_GL2I: DPLL Get of lock interrupt for SUB_INC2	GTM_INT_9	GTM_DMA_9

Table 31.6 Interrupt Requests (2/3)

GTM Interrupt Factors	Description	Interrupt Group	DMAC Trigger Group
GTM_DPLL_IRQ17	DPLL_LL2I: DPLL Lost of lock interrupt for SUB_INC2	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ18	DPLL_TE0I: DPLL TRIGGER event interrupt 0	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ19	DPLL_TE1I: DPLL TRIGGER event interrupt 1	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ20	DPLL_TE2I: DPLL TRIGGER event interrupt 2	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ21	DPLL_TE3I: DPLL TRIGGER event interrupt 3	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ22	DPLL_TE4I: DPLL TRIGGER event interrupt 4	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ23	DPLL_CDTI: DPLL calculated duration interrupt for TRIGGER	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ24	DPLL_CDSI: DPLL calculated duration interrupt for STATE	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ25	DPLL_TORI; TRIGGER out of range interrupt	GTM_INT_9	GTM_DMA_9
GTM_DPLL_IRQ26	DPLL_SORI; STATE out of range interrupt	GTM_INT_9	GTM_DMA_9
GTM_TIM0_IRQ[x]	TIM0 Shared interrupts (x: 0..7)	GTM_INT_0	GTM_DMA_0
GTM_TIM1_IRQ[x]	TIM1 Shared interrupts (x: 0..7)	GTM_INT_1	GTM_DMA_1
GTM_TIM2_IRQ[x]	TIM2 Shared interrupts (x: 0..7)	GTM_INT_2	GTM_DMA_2
GTM_TIM3_IRQ[x]	TIM3 Shared interrupts (x: 0..7)	GTM_INT_3	GTM_DMA_3
GTM_TIM4_IRQ[x]	TIM4 Shared interrupts (x: 0..7)	GTM_INT_4	GTM_DMA_4
GTM_TIM5_IRQ[x]	TIM5 Shared interrupts (x: 0..7)	GTM_INT_5	GTM_DMA_5
GTM_TIM6_IRQ[x]	TIM6 Shared interrupts (x: 0..7)	GTM_INT_6	GTM_DMA_6
GTM_TIM7_IRQ[x]	TIM7 Shared interrupts (x: 0..7)	GTM_INT_7	GTM_DMA_7
GTM_MCS0_IRQ[x]	MCS0 Interrupt for channel x (x: 0..8)	GTM_INT_0	GTM_DMA_0
GTM_MCS1_IRQ[x]	MCS1 Interrupt for channel x (x: 0..8)	GTM_INT_1	GTM_DMA_1
GTM_MCS2_IRQ[x]	MCS2 Interrupt for channel x (x: 0..8)	GTM_INT_2	GTM_DMA_2
GTM_MCS3_IRQ[x]	MCS3 Interrupt for channel x (x: 0..8)	GTM_INT_3	GTM_DMA_3
GTM_MCS4_IRQ[x]	MCS4 Interrupt for channel x (x: 0..8)	GTM_INT_4	GTM_DMA_4
GTM_MCS5_IRQ[x]	MCS5 Interrupt for channel x (x: 0..8)	GTM_INT_5	GTM_DMA_5
GTM_MCS6_IRQ[x]	MCS6 Interrupt for channel x (x: 0..8)	GTM_INT_6	GTM_DMA_6
GTM_TOM0_IRQ[x]	TOM0 Shared interrupts for x: 0..7 = {ch0 ch1,...,ch14 ch15}	GTM_INT_0	GTM_DMA_0
GTM_TOM1_IRQ[x]	TOM1 Shared interrupts for x: 0..7 = {ch0 ch1,...,ch14 ch15}	GTM_INT_1	GTM_DMA_1
GTM_TOM2_IRQ[x]	TOM2 Shared interrupts for x: 0..7 = {ch0 ch1,...,ch14 ch15}	GTM_INT_2	GTM_DMA_2
GTM_TOM3_IRQ[x]	TOM3 Shared interrupts for x: 0..7 = {ch0 ch1,...,ch14 ch15}	GTM_INT_3	GTM_DMA_3
GTM_TOM4_IRQ[x]*1	TOM4 Shared interrupts for x: 0..7 = {ch0 ch1,...,ch14 ch15}	GTM_INT_4	GTM_DMA_4

Table 31.6 Interrupt Requests (3/3)

GTM Interrupt Factors	Description	Interrupt Group	DMAC Trigger Group
GTM_ATOM0_IRQ[x]	ATOM0 Shared interrupts for x: 0..3 = {ch0 ch1,...,ch6 ch7}	GTM_INT_0	GTM_DMA_0
GTM_ATOM1_IRQ[x]	ATOM1 Shared interrupts for x: 0..3 = {ch0 ch1,...,ch6 ch7}	GTM_INT_1	GTM_DMA_1
GTM_ATOM2_IRQ[x]	ATOM2 Shared interrupts for x: 0..3 = {ch0 ch1,...,ch6 ch7}	GTM_INT_2	GTM_DMA_2
GTM_ATOM3_IRQ[x]	ATOM3 Shared interrupts for x: 0..3 = {ch0 ch1,...,ch6 ch7}	GTM_INT_3	GTM_DMA_3
GTM_ATOM4_IRQ[x]	ATOM4 Shared interrupts for x: 0..3 = {ch0 ch1,...,ch6 ch7}	GTM_INT_4	GTM_DMA_4
GTM_ATOM5_IRQ[x]	ATOM5 Shared interrupts for x: 0..3 = {ch0 ch1,...,ch6 ch7}	GTM_INT_5	GTM_DMA_5
GTM_ATOM6_IRQ[x]	ATOM6 Shared interrupts for x: 0..3 = {ch0 ch1,...,ch6 ch7}	GTM_INT_6	GTM_DMA_6
GTM_ATOM7_IRQ[x]	ATOM7 Shared interrupts for x: 0..3 = {ch0 ch1,...,ch6 ch7}	GTM_INT_7	GTM_DMA_7
GTM_ATOM8_IRQ[x]*1	ATOM8 Shared interrupts for x: 0..3 = {ch0 ch1,...,ch6 ch7}	GTM_INT_8	GTM_DMA_8
GTM_ERR_IRQ	GTM Error Interrupt	GTM_INT_16	—

Note 1. E2x-FCC1 only

Table 31.7 Interrupt Factor Groups

INT Groups	Interrupt Symbol Name	Registers	Interrupt Number
GTM_INT_0	INTGTM0IOS0n	GTM_IRQ_SEL0j	78 to 93
GTM_INT_1	INTGTM0IOS1n	GTM_IRQ_SEL1j	94 to 109
GTM_INT_2	INTGTM0IOS2n	GTM_IRQ_SEL2j	110 to 125
GTM_INT_3	INTGTM0IOS3n	GTM_IRQ_SEL3j	126 to 141
GTM_INT_4	INTGTM0IOS4n	GTM_IRQ_SEL4j	142 to 157
GTM_INT_5	INTGTM0IOS5n	GTM_IRQ_SEL5j	158 to 173
GTM_INT_6	INTGTM0IOS6n	GTM_IRQ_SEL6j	174 to 189
GTM_INT_7	INTGTM0IOS7n	GTM_IRQ_SEL7j	190 to 205
GTM_INT_8	INTGTM0IOS8n	GTM_IRQ_SEL8j	206 to 221
GTM_INT_9	INTGTM0DPLLn	GTM_IRQ_DPLL_SELj	229 to 244
GTM_INT_10	INTGTM0AEI	—	222
GTM_INT_11	INTGTM0ARU0	—	223
GTM_INT_12	INTGTM0ARU1	—	224
GTM_INT_13	INTGTM0ARU2	—	225
GTM_INT_14	INTGTM0BRC	—	226
GTM_INT_15	INTGTM0CMP	—	227
GTM_INT_16	INTGTM0ERR	—	228

Table 31.8 DMA Factor Groups

DMA Groups	Interrupt Symbol Name	Registers	sDMAC Trigger Number	DTS Trigger Number	
			Group1	Group2	Group3
GTM_DMA_0	INTGTM0DOS0n	GTM_DMA_SEL0j	0 to 7	46 to 53	88 to 95
GTM_DMA_1	INTGTM0DOS1n	GTM_DMA_SEL1j	8 to 15	54 to 61	96 to 103
GTM_DMA_2	INTGTM0DOS2n	GTM_DMA_SEL2j	16 to 23	62 to 69	104 to 111
GTM_DMA_3	INTGTM0DOS3n	GTM_DMA_SEL3j	24 to 31	70 to 77	112 to 119
GTM_DMA_4	INTGTM0DOS4n	GTM_DMA_SEL4j	32 to 39	78 to 85	120 to 127
GTM_DMA_5	INTGTM0DOS5n	GTM_DMA_SEL5j	40 to 47	86 to 93	46 to 53
GTM_DMA_6	INTGTM0DOS6n	GTM_DMA_SEL6j	48 to 55	94 to 101	54 to 61
GTM_DMA_7	INTGTM0DOS7n	GTM_DMA_SEL7j	56 to 63	102 to 109	62 to 69
GTM_DMA_8	INTGTM0DOS8n	GTM_DMA_SEL8j	64 to 71	110 to 117	70 to 77
GTM_DMA_9	INTGTM0DOS0n	GTM_DMA_COM_SELj	72 to 81	118 to 127	78 to 87

31.1.6 Reset Sources

The following table shows GTM reset sources and GTM responses to each reset source.

Table 31.9 Reset Sources

Register Name	Reset Condition					
	Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
All registers	✓	✓	✓	✓	—	—

31.1.7 External Input/Output Signals

External input/output signals of GTM are listed in the following table.

Table 31.10 External Input/Output Signals (1/2)

RH850/E2x-FCC1			
Unit Signal Name	I/O	Outline	Alternative Port Pin Signal
GTM			
TIM[i]_IN0 to TIM[i]_IN7 (i: 0 to 7)	I	Timer input signals for TIM[i] (i: 0 to 7)	TIM[i]_0 to TIM[i]_7 (i: 0 to 7)
TOM[i]_OUT0 to TOM[i]_OUT15 (i: 0 to 4)	O	Timer output signals for TOM[i] (i: 0 to 4)	TOM[i]_0 to TOM[i]_15 (i: 0 to 4)
TOM[i]_OUT0_N to TOM[i]_OUT7_N (i: 0 to 3)	O	Inverted timer output signal for TOM[i] (i: 0 to 3)	TOM[i]_0N to TOM[i]_7N (i: 0 to 3)
ATOM[i]_OUT0 to ATOM[i]_OUT7 (i: 0 to 8)	O	Timer output signals for ATOM[i] (i: 0 to 8)	ATOM[i]_0 to ATOM[i]_7 (i: 0 to 8)
ATOM[i]_OUT0_N to ATOM[i]_OUT7_N (i: 0 to 4)	O	Inverted timer output signal for ATOM[i] (i: 0 to 4)	ATOM[i]_0N to ATOM[i]_7N (i: 0 to 4)
ATOM[i]_OUT0_N to ATOM[i]_OUT3_N (i: 5 to 6)	O	Inverted timer output signal for ATOM[i] (i: 5 to 6)	ATOM[i]_0N to ATOM[i]_3N (i: 5 to 6)
CMU_ECLK0 to CMU_ECLK2	O	External Clock	GTMECLK0 to GTMECLK2

Table 31.10 External Input/Output Signals (2/2)

RH850/E2M			
Unit Signal Name	I/O	Outline	Alternative Port Pin Signal
GTM			
TIM[i]_IN0 to TIM[i]_IN7 (i: 0 to 7)	I	Timer input signals for TIM[i] (i: 0 to 7)	TIM[i]_0 to TIM[i]_7 (i: 0 to 7)
TOM[i]_OUT0 to TOM[i]_OUT15 (i: 0 to 3)	O	Timer output signals for TOM[i] (i: 0 to 3)	TOM[i]_0 to TOM[i]_15 (i: 0 to 3)
TOM[i]_OUT0_N to TOM[i]_OUT7_N (i: 0 to 2)	O	Inverted timer output signal for TOM[i] (i: 0 to 2)	TOM[i]_0N to TOM[i]_7N (i: 0 to 2)
ATOM[i]_OUT0 to ATOM[i]_OUT7 (i: 0 to 7)	O	Timer output signals for ATOM[i] (i: 0 to 7)	ATOM[i]_0 to ATOM[i]_7 (i: 0 to 7)
ATOM[i]_OUT0_N to ATOM[i]_OUT7_N (i: 0 to 4)	O	Inverted timer output signal for ATOM[i] (i: 0 to 4)	ATOM[i]_0N to ATOM[i]_7N (i: 0 to 4)
CMU_ECLK0 to CMU_ECLK2	O	External Clock	GTMECLK0 to GTMECLK2

31.2 Overview

31.2.1 Functional Overview

GTM consists of the following sub-modules.

- Advanced Routing Unit (ARU)
 - Data routing between GTM sub-modules.
- Broadcast Module (BRC)
 - Data broadcasting from a source sub-module to multiple destination sub-modules via ARU.
- Parameter Storage Module (PSM)
 - First In First Out Module. (FIFO)
 - AEI-to-FIFO Data Interface. (AFD)
 - FIFO-to-ARU Unit. (F2A)
- Clock Management Unit (CMU)
 - Clock pre-scaler for GTM internal and external clocks.
- Cluster Configuration Module (CCM)
 - Controlling cluster configurations:
 - Cluster's clock frequency.
 - Module clock gating.
 - Status monitoring of the cluster's MCS bus master.
 - Address range protection.
- Time Base Unit (TBU)
 - Three independent 24-bit time bases.
 - One module counter for angular conversions.
 - Supporting the following modes:
 - Free Running Counter Mode.
 - Forward/Backward Counter Mode.
 - Connected to angle clock. (DPLL)
- Timer Input Module (TIM)
 - Filter, capture/compare with a time stamp and input signal timeout.
 - Input from other peripherals to GTM.
- Timer Output Module (TOM)
 - Simple PWM waveform generation with 16-bit time base counter.
 - Input signal to other peripherals.
- ARU-connected Timer Output Module (ATOM)
 - Complex output waveform generation without CPU interactions with 24-bit time base counter.
 - Input signal to other peripherals.

- Dead Time Module (DTM)
 - Hardware support for dead time generation.
- Multi Channel Sequencer (MCS)
 - Programmable sequencer with RISC-like instruction set and dedicated RAM.
- Memory Configuration (MCFG)
 - Mapping MCS physical memory to MCS instances.
- TIM0 Input Mapping Module (MAP)
 - Mapping TIM0 input signals to DPLL.
- Digital PLL (DPLL)
 - Angle calculation and micro tick generation from up to two input signals.
 - Prediction of future behavior of input signal.
- Sensor Pattern Evaluation (SPE)
 - Hardware support for BLDC motors.
- Interrupt Concentrator Module (ICM)
 - Bundling GTM internal interrupts before they are routed to INTC and DMAC.
- Output Compare Unit (CMP)
 - Support for safety relevant applications, compare of DTM outputs.
- Monitoring Unit (MON)
 - Monitoring GTM internal clocks and MCS functionality.

31.2.2 Block Diagram

The following figure shows interconnections between GTM-IP and other peripherals.

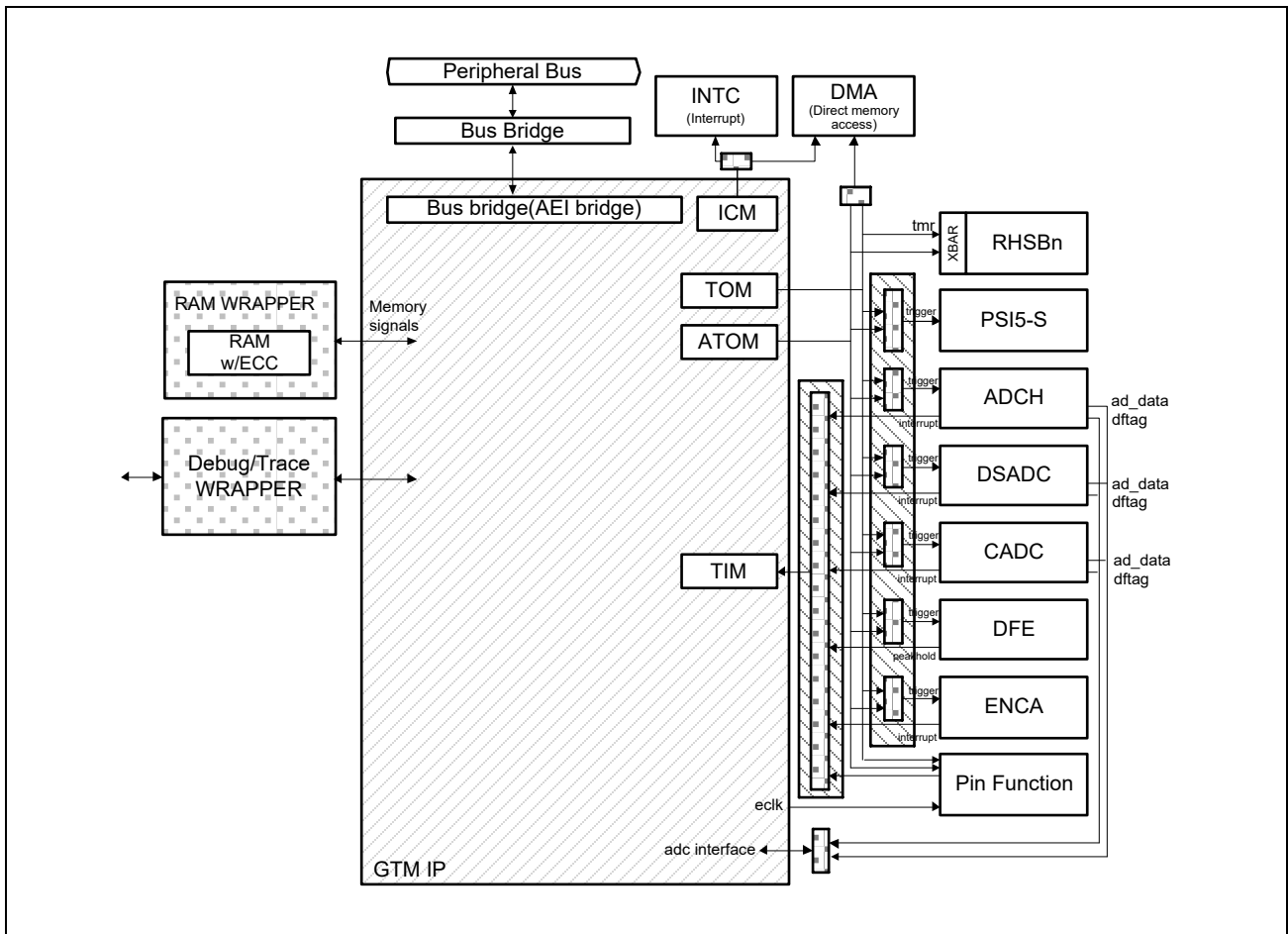


Figure 31.1 Block Diagram of GTM

31.3 Registers

31.3.1 List of Registers

Registers in the table below configure routing between ICM outputs and INTC and DMAC/DTS.

For details about <GTM0_base><GTM0_1_base>, see **Section 31.1.3, Register/RAM Base Address**.

Table 31.11 List of Registers (1/7)

Module Name	Register Name	Symbol	Address	Access	Access Protection
GTM0	All of GTM-IP registers	*1	<GTM0_base> + *1	32	—
GTM0_1	GTM Interrupt selection control register 000	GTM_IRQ_SEL000	<GTM0_1_base> + 000 _H	8, 16, 32	—
	GTM Interrupt selection control register 001	GTM_IRQ_SEL001	<GTM0_1_base> + 004 _H	8, 16, 32	—
	GTM Interrupt selection control register 002	GTM_IRQ_SEL002	<GTM0_1_base> + 008 _H	8, 16, 32	—
	GTM Interrupt selection control register 003	GTM_IRQ_SEL003	<GTM0_1_base> + 00C _H	8, 16, 32	—
	GTM Interrupt selection control register 004	GTM_IRQ_SEL004	<GTM0_1_base> + 010 _H	8, 16, 32	—
	GTM Interrupt selection control register 005	GTM_IRQ_SEL005	<GTM0_1_base> + 014 _H	8, 16, 32	—
	GTM Interrupt selection control register 006	GTM_IRQ_SEL006	<GTM0_1_base> + 018 _H	8, 16, 32	—
	GTM Interrupt selection control register 007	GTM_IRQ_SEL007	<GTM0_1_base> + 01C _H	8, 16, 32	—
	GTM Interrupt selection control register 008	GTM_IRQ_SEL008	<GTM0_1_base> + 020 _H	8, 16, 32	—
	GTM Interrupt selection control register 009	GTM_IRQ_SEL009	<GTM0_1_base> + 024 _H	8, 16, 32	—
	GTM Interrupt selection control register 010	GTM_IRQ_SEL010	<GTM0_1_base> + 028 _H	8, 16, 32	—
	GTM Interrupt selection control register 011	GTM_IRQ_SEL011	<GTM0_1_base> + 02C _H	8, 16, 32	—
	GTM Interrupt selection control register 012	GTM_IRQ_SEL012	<GTM0_1_base> + 030 _H	8, 16, 32	—
	GTM Interrupt selection control register 013	GTM_IRQ_SEL013	<GTM0_1_base> + 034 _H	8, 16, 32	—
	GTM Interrupt selection control register 014	GTM_IRQ_SEL014	<GTM0_1_base> + 038 _H	8, 16, 32	—
	GTM Interrupt selection control register 015	GTM_IRQ_SEL015	<GTM0_1_base> + 03C _H	8, 16, 32	—
	GTM Interrupt selection control register 100	GTM_IRQ_SEL100	<GTM0_1_base> + 040 _H	8, 16, 32	—
	GTM Interrupt selection control register 101	GTM_IRQ_SEL101	<GTM0_1_base> + 044 _H	8, 16, 32	—
	GTM Interrupt selection control register 102	GTM_IRQ_SEL102	<GTM0_1_base> + 048 _H	8, 16, 32	—
	GTM Interrupt selection control register 103	GTM_IRQ_SEL103	<GTM0_1_base> + 04C _H	8, 16, 32	—
	GTM Interrupt selection control register 104	GTM_IRQ_SEL104	<GTM0_1_base> + 050 _H	8, 16, 32	—
	GTM Interrupt selection control register 105	GTM_IRQ_SEL105	<GTM0_1_base> + 054 _H	8, 16, 32	—
	GTM Interrupt selection control register 106	GTM_IRQ_SEL106	<GTM0_1_base> + 058 _H	8, 16, 32	—
	GTM Interrupt selection control register 107	GTM_IRQ_SEL107	<GTM0_1_base> + 05C _H	8, 16, 32	—
	GTM Interrupt selection control register 108	GTM_IRQ_SEL108	<GTM0_1_base> + 060 _H	8, 16, 32	—
	GTM Interrupt selection control register 109	GTM_IRQ_SEL109	<GTM0_1_base> + 064 _H	8, 16, 32	—
	GTM Interrupt selection control register 110	GTM_IRQ_SEL110	<GTM0_1_base> + 068 _H	8, 16, 32	—
	GTM Interrupt selection control register 111	GTM_IRQ_SEL111	<GTM0_1_base> + 06C _H	8, 16, 32	—
	GTM Interrupt selection control register 112	GTM_IRQ_SEL112	<GTM0_1_base> + 070 _H	8, 16, 32	—
	GTM Interrupt selection control register 113	GTM_IRQ_SEL113	<GTM0_1_base> + 074 _H	8, 16, 32	—
	GTM Interrupt selection control register 114	GTM_IRQ_SEL114	<GTM0_1_base> + 078 _H	8, 16, 32	—
	GTM Interrupt selection control register 115	GTM_IRQ_SEL115	<GTM0_1_base> + 07C _H	8, 16, 32	—
	GTM Interrupt selection control register 200	GTM_IRQ_SEL200	<GTM0_1_base> + 080 _H	8, 16, 32	—
	GTM Interrupt selection control register 201	GTM_IRQ_SEL201	<GTM0_1_base> + 084 _H	8, 16, 32	—
	GTM Interrupt selection control register 202	GTM_IRQ_SEL202	<GTM0_1_base> + 088 _H	8, 16, 32	—
	GTM Interrupt selection control register 203	GTM_IRQ_SEL203	<GTM0_1_base> + 08C _H	8, 16, 32	—
	GTM Interrupt selection control register 204	GTM_IRQ_SEL204	<GTM0_1_base> + 090 _H	8, 16, 32	—

Table 31.11 List of Registers (2/7)

Module Name	Register Name	Symbol	Address	Access	Access Protection
GTM0_1	GTM Interrupt selection control register 205	GTM_IRQ_SEL205	<GTM0_1_base> + 094H	8, 16, 32	—
	GTM Interrupt selection control register 206	GTM_IRQ_SEL206	<GTM0_1_base> + 098H	8, 16, 32	—
	GTM Interrupt selection control register 207	GTM_IRQ_SEL207	<GTM0_1_base> + 09CH	8, 16, 32	—
	GTM Interrupt selection control register 208	GTM_IRQ_SEL208	<GTM0_1_base> + 0A0H	8, 16, 32	—
	GTM Interrupt selection control register 209	GTM_IRQ_SEL209	<GTM0_1_base> + 0A4H	8, 16, 32	—
	GTM Interrupt selection control register 210	GTM_IRQ_SEL210	<GTM0_1_base> + 0A8H	8, 16, 32	—
	GTM Interrupt selection control register 211	GTM_IRQ_SEL211	<GTM0_1_base> + 0ACH	8, 16, 32	—
	GTM Interrupt selection control register 212	GTM_IRQ_SEL212	<GTM0_1_base> + 0B0H	8, 16, 32	—
	GTM Interrupt selection control register 213	GTM_IRQ_SEL213	<GTM0_1_base> + 0B4H	8, 16, 32	—
	GTM Interrupt selection control register 214	GTM_IRQ_SEL214	<GTM0_1_base> + 0B8H	8, 16, 32	—
	GTM Interrupt selection control register 215	GTM_IRQ_SEL215	<GTM0_1_base> + 0BCH	8, 16, 32	—
	GTM Interrupt selection control register 300	GTM_IRQ_SEL300	<GTM0_1_base> + 0C0H	8, 16, 32	—
	GTM Interrupt selection control register 301	GTM_IRQ_SEL301	<GTM0_1_base> + 0C4H	8, 16, 32	—
	GTM Interrupt selection control register 302	GTM_IRQ_SEL302	<GTM0_1_base> + 0C8H	8, 16, 32	—
	GTM Interrupt selection control register 303	GTM_IRQ_SEL303	<GTM0_1_base> + 0CCH	8, 16, 32	—
	GTM Interrupt selection control register 304	GTM_IRQ_SEL304	<GTM0_1_base> + 0D0H	8, 16, 32	—
	GTM Interrupt selection control register 305	GTM_IRQ_SEL305	<GTM0_1_base> + 0D4H	8, 16, 32	—
	GTM Interrupt selection control register 306	GTM_IRQ_SEL306	<GTM0_1_base> + 0D8H	8, 16, 32	—
	GTM Interrupt selection control register 307	GTM_IRQ_SEL307	<GTM0_1_base> + 0DCH	8, 16, 32	—
	GTM Interrupt selection control register 308	GTM_IRQ_SEL308	<GTM0_1_base> + 0E0H	8, 16, 32	—
	GTM Interrupt selection control register 309	GTM_IRQ_SEL309	<GTM0_1_base> + 0E4H	8, 16, 32	—
	GTM Interrupt selection control register 310	GTM_IRQ_SEL310	<GTM0_1_base> + 0E8H	8, 16, 32	—
	GTM Interrupt selection control register 311	GTM_IRQ_SEL311	<GTM0_1_base> + 0ECH	8, 16, 32	—
	GTM Interrupt selection control register 312	GTM_IRQ_SEL312	<GTM0_1_base> + 0F0H	8, 16, 32	—
	GTM Interrupt selection control register 313	GTM_IRQ_SEL313	<GTM0_1_base> + 0F4H	8, 16, 32	—
	GTM Interrupt selection control register 314	GTM_IRQ_SEL314	<GTM0_1_base> + 0F8H	8, 16, 32	—
	GTM Interrupt selection control register 315	GTM_IRQ_SEL315	<GTM0_1_base> + 0FCH	8, 16, 32	—
	GTM Interrupt selection control register 400	GTM_IRQ_SEL400	<GTM0_1_base> + 100H	8, 16, 32	—
	GTM Interrupt selection control register 401	GTM_IRQ_SEL401	<GTM0_1_base> + 104H	8, 16, 32	—
	GTM Interrupt selection control register 402	GTM_IRQ_SEL402	<GTM0_1_base> + 108H	8, 16, 32	—
	GTM Interrupt selection control register 403	GTM_IRQ_SEL403	<GTM0_1_base> + 10CH	8, 16, 32	—
	GTM Interrupt selection control register 404	GTM_IRQ_SEL404	<GTM0_1_base> + 110H	8, 16, 32	—
	GTM Interrupt selection control register 405	GTM_IRQ_SEL405	<GTM0_1_base> + 114H	8, 16, 32	—
	GTM Interrupt selection control register 406	GTM_IRQ_SEL406	<GTM0_1_base> + 118H	8, 16, 32	—
	GTM Interrupt selection control register 407	GTM_IRQ_SEL407	<GTM0_1_base> + 11CH	8, 16, 32	—
	GTM Interrupt selection control register 408	GTM_IRQ_SEL408	<GTM0_1_base> + 120H	8, 16, 32	—
	GTM Interrupt selection control register 409	GTM_IRQ_SEL409	<GTM0_1_base> + 124H	8, 16, 32	—
	GTM Interrupt selection control register 410	GTM_IRQ_SEL410	<GTM0_1_base> + 128H	8, 16, 32	—
	GTM Interrupt selection control register 411	GTM_IRQ_SEL411	<GTM0_1_base> + 12CH	8, 16, 32	—
	GTM Interrupt selection control register 412	GTM_IRQ_SEL412	<GTM0_1_base> + 130H	8, 16, 32	—
	GTM Interrupt selection control register 413	GTM_IRQ_SEL413	<GTM0_1_base> + 134H	8, 16, 32	—
	GTM Interrupt selection control register 414	GTM_IRQ_SEL414	<GTM0_1_base> + 138H	8, 16, 32	—
	GTM Interrupt selection control register 415	GTM_IRQ_SEL415	<GTM0_1_base> + 13CH	8, 16, 32	—
	GTM Interrupt selection control register 500	GTM_IRQ_SEL500	<GTM0_1_base> + 140H	8, 16, 32	—

Table 31.11 List of Registers (3/7)

Module Name	Register Name	Symbol	Address	Access	Access Protection
GTM0_1	GTM Interrupt selection control register 501	GTM_IRQ_SEL501	<GTM0_1_base> + 144 _H	8, 16, 32	—
	GTM Interrupt selection control register 502	GTM_IRQ_SEL502	<GTM0_1_base> + 148 _H	8, 16, 32	—
	GTM Interrupt selection control register 503	GTM_IRQ_SEL503	<GTM0_1_base> + 14C _H	8, 16, 32	—
	GTM Interrupt selection control register 504	GTM_IRQ_SEL504	<GTM0_1_base> + 150 _H	8, 16, 32	—
	GTM Interrupt selection control register 505	GTM_IRQ_SEL505	<GTM0_1_base> + 154 _H	8, 16, 32	—
	GTM Interrupt selection control register 506	GTM_IRQ_SEL506	<GTM0_1_base> + 158 _H	8, 16, 32	—
	GTM Interrupt selection control register 507	GTM_IRQ_SEL507	<GTM0_1_base> + 15C _H	8, 16, 32	—
	GTM Interrupt selection control register 508	GTM_IRQ_SEL508	<GTM0_1_base> + 160 _H	8, 16, 32	—
	GTM Interrupt selection control register 509	GTM_IRQ_SEL509	<GTM0_1_base> + 164 _H	8, 16, 32	—
	GTM Interrupt selection control register 510	GTM_IRQ_SEL510	<GTM0_1_base> + 168 _H	8, 16, 32	—
	GTM Interrupt selection control register 511	GTM_IRQ_SEL511	<GTM0_1_base> + 16C _H	8, 16, 32	—
	GTM Interrupt selection control register 512	GTM_IRQ_SEL512	<GTM0_1_base> + 170 _H	8, 16, 32	—
	GTM Interrupt selection control register 513	GTM_IRQ_SEL513	<GTM0_1_base> + 174 _H	8, 16, 32	—
	GTM Interrupt selection control register 514	GTM_IRQ_SEL514	<GTM0_1_base> + 178 _H	8, 16, 32	—
	GTM Interrupt selection control register 515	GTM_IRQ_SEL515	<GTM0_1_base> + 17C _H	8, 16, 32	—
	GTM Interrupt selection control register 600	GTM_IRQ_SEL600	<GTM0_1_base> + 180 _H	8, 16, 32	—
	GTM Interrupt selection control register 601	GTM_IRQ_SEL601	<GTM0_1_base> + 184 _H	8, 16, 32	—
	GTM Interrupt selection control register 602	GTM_IRQ_SEL602	<GTM0_1_base> + 188 _H	8, 16, 32	—
	GTM Interrupt selection control register 603	GTM_IRQ_SEL603	<GTM0_1_base> + 18C _H	8, 16, 32	—
	GTM Interrupt selection control register 604	GTM_IRQ_SEL604	<GTM0_1_base> + 190 _H	8, 16, 32	—
	GTM Interrupt selection control register 605	GTM_IRQ_SEL605	<GTM0_1_base> + 194 _H	8, 16, 32	—
	GTM Interrupt selection control register 606	GTM_IRQ_SEL606	<GTM0_1_base> + 198 _H	8, 16, 32	—
	GTM Interrupt selection control register 607	GTM_IRQ_SEL607	<GTM0_1_base> + 19C _H	8, 16, 32	—
	GTM Interrupt selection control register 608	GTM_IRQ_SEL608	<GTM0_1_base> + 1A0 _H	8, 16, 32	—
	GTM Interrupt selection control register 609	GTM_IRQ_SEL609	<GTM0_1_base> + 1A4 _H	8, 16, 32	—
	GTM Interrupt selection control register 610	GTM_IRQ_SEL610	<GTM0_1_base> + 1A8 _H	8, 16, 32	—
	GTM Interrupt selection control register 611	GTM_IRQ_SEL611	<GTM0_1_base> + 1AC _H	8, 16, 32	—
	GTM Interrupt selection control register 612	GTM_IRQ_SEL612	<GTM0_1_base> + 1B0 _H	8, 16, 32	—
	GTM Interrupt selection control register 613	GTM_IRQ_SEL613	<GTM0_1_base> + 1B4 _H	8, 16, 32	—
	GTM Interrupt selection control register 614	GTM_IRQ_SEL614	<GTM0_1_base> + 1B8 _H	8, 16, 32	—
	GTM Interrupt selection control register 615	GTM_IRQ_SEL615	<GTM0_1_base> + 1BC _H	8, 16, 32	—
	GTM Interrupt selection control register 700	GTM_IRQ_SEL700	<GTM0_1_base> + 1C0 _H	8, 16, 32	—
	GTM Interrupt selection control register 701	GTM_IRQ_SEL701	<GTM0_1_base> + 1C4 _H	8, 16, 32	—
	GTM Interrupt selection control register 702	GTM_IRQ_SEL702	<GTM0_1_base> + 1C8 _H	8, 16, 32	—
	GTM Interrupt selection control register 703	GTM_IRQ_SEL703	<GTM0_1_base> + 1CC _H	8, 16, 32	—
	GTM Interrupt selection control register 704	GTM_IRQ_SEL704	<GTM0_1_base> + 1D0 _H	8, 16, 32	—
	GTM Interrupt selection control register 705	GTM_IRQ_SEL705	<GTM0_1_base> + 1D4 _H	8, 16, 32	—
	GTM Interrupt selection control register 706	GTM_IRQ_SEL706	<GTM0_1_base> + 1D8 _H	8, 16, 32	—
	GTM Interrupt selection control register 707	GTM_IRQ_SEL707	<GTM0_1_base> + 1DC _H	8, 16, 32	—
	GTM Interrupt selection control register 708	GTM_IRQ_SEL708	<GTM0_1_base> + 1E0 _H	8, 16, 32	—
	GTM Interrupt selection control register 709	GTM_IRQ_SEL709	<GTM0_1_base> + 1E4 _H	8, 16, 32	—
	GTM Interrupt selection control register 710	GTM_IRQ_SEL710	<GTM0_1_base> + 1E8 _H	8, 16, 32	—
	GTM Interrupt selection control register 711	GTM_IRQ_SEL711	<GTM0_1_base> + 1EC _H	8, 16, 32	—
	GTM Interrupt selection control register 712	GTM_IRQ_SEL712	<GTM0_1_base> + 1F0 _H	8, 16, 32	—
	GTM Interrupt selection control register 713	GTM_IRQ_SEL713	<GTM0_1_base> + 1F4 _H	8, 16, 32	—
	GTM Interrupt selection control register 714	GTM_IRQ_SEL714	<GTM0_1_base> + 1F8 _H	8, 16, 32	—
	GTM Interrupt selection control register 715	GTM_IRQ_SEL715	<GTM0_1_base> + 1FC _H	8, 16, 32	—

Table 31.11 List of Registers (4/7)

Module Name	Register Name	Symbol	Address	Access	Access Protection
GTM0_1	GTM Interrupt selection control register 800	GTM_IRQ_SEL800	<GTM0_1_base> + 200 _H	8, 16, 32	—
	GTM Interrupt selection control register 801	GTM_IRQ_SEL801	<GTM0_1_base> + 204 _H	8, 16, 32	—
	GTM Interrupt selection control register 802	GTM_IRQ_SEL802	<GTM0_1_base> + 208 _H	8, 16, 32	—
	GTM Interrupt selection control register 803	GTM_IRQ_SEL803	<GTM0_1_base> + 20C _H	8, 16, 32	—
	GTM Interrupt selection control register 804	GTM_IRQ_SEL804	<GTM0_1_base> + 210 _H	8, 16, 32	—
	GTM Interrupt selection control register 805	GTM_IRQ_SEL805	<GTM0_1_base> + 214 _H	8, 16, 32	—
	GTM Interrupt selection control register 806	GTM_IRQ_SEL806	<GTM0_1_base> + 218 _H	8, 16, 32	—
	GTM Interrupt selection control register 807	GTM_IRQ_SEL807	<GTM0_1_base> + 21C _H	8, 16, 32	—
	GTM Interrupt selection control register 808	GTM_IRQ_SEL808	<GTM0_1_base> + 220 _H	8, 16, 32	—
	GTM Interrupt selection control register 809	GTM_IRQ_SEL809	<GTM0_1_base> + 224 _H	8, 16, 32	—
	GTM Interrupt selection control register 810	GTM_IRQ_SEL810	<GTM0_1_base> + 228 _H	8, 16, 32	—
	GTM Interrupt selection control register 811	GTM_IRQ_SEL811	<GTM0_1_base> + 22C _H	8, 16, 32	—
	GTM Interrupt selection control register 812	GTM_IRQ_SEL812	<GTM0_1_base> + 230 _H	8, 16, 32	—
	GTM Interrupt selection control register 813	GTM_IRQ_SEL813	<GTM0_1_base> + 234 _H	8, 16, 32	—
	GTM Interrupt selection control register 814	GTM_IRQ_SEL814	<GTM0_1_base> + 238 _H	8, 16, 32	—
	GTM Interrupt selection control register 815	GTM_IRQ_SEL815	<GTM0_1_base> + 23C _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 0	GTM_IRQ_DPLL_SEL0	<GTM0_1_base> + 240 _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 1	GTM_IRQ_DPLL_SEL1	<GTM0_1_base> + 244 _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 2	GTM_IRQ_DPLL_SEL2	<GTM0_1_base> + 248 _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 3	GTM_IRQ_DPLL_SEL3	<GTM0_1_base> + 24C _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 4	GTM_IRQ_DPLL_SEL4	<GTM0_1_base> + 250 _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 5	GTM_IRQ_DPLL_SEL5	<GTM0_1_base> + 254 _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 6	GTM_IRQ_DPLL_SEL6	<GTM0_1_base> + 258 _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 7	GTM_IRQ_DPLL_SEL7	<GTM0_1_base> + 25C _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 8	GTM_IRQ_DPLL_SEL8	<GTM0_1_base> + 260 _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 9	GTM_IRQ_DPLL_SEL9	<GTM0_1_base> + 264 _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 10	GTM_IRQ_DPLL_SEL10	<GTM0_1_base> + 268 _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 11	GTM_IRQ_DPLL_SEL11	<GTM0_1_base> + 26C _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 12	GTM_IRQ_DPLL_SEL12	<GTM0_1_base> + 270 _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 13	GTM_IRQ_DPLL_SEL13	<GTM0_1_base> + 274 _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 14	GTM_IRQ_DPLL_SEL14	<GTM0_1_base> + 278 _H	8, 16, 32	—
	GTM Interrupt DPLL selection register 15	GTM_IRQ_DPLL_SEL15	<GTM0_1_base> + 27C _H	8, 16, 32	—

Table 31.11 List of Registers (5/7)

Module Name	Register Name	Symbol	Address	Access	Access Protection
GTM0_1	GTM DMA selection control register 00	GTM_DMA_SEL00	<GTM0_1_base> + 300 _H	8, 16, 32	—
	GTM DMA selection control register 01	GTM_DMA_SEL01	<GTM0_1_base> + 304 _H	8, 16, 32	—
	GTM DMA selection control register 02	GTM_DMA_SEL02	<GTM0_1_base> + 308 _H	8, 16, 32	—
	GTM DMA selection control register 03	GTM_DMA_SEL03	<GTM0_1_base> + 30C _H	8, 16, 32	—
	GTM DMA selection control register 04	GTM_DMA_SEL04	<GTM0_1_base> + 310 _H	8, 16, 32	—
	GTM DMA selection control register 05	GTM_DMA_SEL05	<GTM0_1_base> + 314 _H	8, 16, 32	—
	GTM DMA selection control register 06	GTM_DMA_SEL06	<GTM0_1_base> + 318 _H	8, 16, 32	—
	GTM DMA selection control register 07	GTM_DMA_SEL07	<GTM0_1_base> + 31C _H	8, 16, 32	—
	GTM DMA selection control register 10	GTM_DMA_SEL10	<GTM0_1_base> + 320 _H	8, 16, 32	—
	GTM DMA selection control register 11	GTM_DMA_SEL11	<GTM0_1_base> + 324 _H	8, 16, 32	—
	GTM DMA selection control register 12	GTM_DMA_SEL12	<GTM0_1_base> + 328 _H	8, 16, 32	—
	GTM DMA selection control register 13	GTM_DMA_SEL13	<GTM0_1_base> + 32C _H	8, 16, 32	—
	GTM DMA selection control register 14	GTM_DMA_SEL14	<GTM0_1_base> + 330 _H	8, 16, 32	—
	GTM DMA selection control register 15	GTM_DMA_SEL15	<GTM0_1_base> + 334 _H	8, 16, 32	—
	GTM DMA selection control register 16	GTM_DMA_SEL16	<GTM0_1_base> + 338 _H	8, 16, 32	—
	GTM DMA selection control register 17	GTM_DMA_SEL17	<GTM0_1_base> + 33C _H	8, 16, 32	—
	GTM DMA selection control register 20	GTM_DMA_SEL20	<GTM0_1_base> + 340 _H	8, 16, 32	—
	GTM DMA selection control register 21	GTM_DMA_SEL21	<GTM0_1_base> + 344 _H	8, 16, 32	—
	GTM DMA selection control register 22	GTM_DMA_SEL22	<GTM0_1_base> + 348 _H	8, 16, 32	—
	GTM DMA selection control register 23	GTM_DMA_SEL23	<GTM0_1_base> + 34C _H	8, 16, 32	—
	GTM DMA selection control register 24	GTM_DMA_SEL24	<GTM0_1_base> + 350 _H	8, 16, 32	—
	GTM DMA selection control register 25	GTM_DMA_SEL25	<GTM0_1_base> + 354 _H	8, 16, 32	—
	GTM DMA selection control register 26	GTM_DMA_SEL26	<GTM0_1_base> + 358 _H	8, 16, 32	—
	GTM DMA selection control register 27	GTM_DMA_SEL27	<GTM0_1_base> + 35C _H	8, 16, 32	—
	GTM DMA selection control register 30	GTM_DMA_SEL30	<GTM0_1_base> + 360 _H	8, 16, 32	—
	GTM DMA selection control register 31	GTM_DMA_SEL31	<GTM0_1_base> + 364 _H	8, 16, 32	—
	GTM DMA selection control register 32	GTM_DMA_SEL32	<GTM0_1_base> + 368 _H	8, 16, 32	—
	GTM DMA selection control register 33	GTM_DMA_SEL33	<GTM0_1_base> + 36C _H	8, 16, 32	—
	GTM DMA selection control register 34	GTM_DMA_SEL34	<GTM0_1_base> + 370 _H	8, 16, 32	—
	GTM DMA selection control register 35	GTM_DMA_SEL35	<GTM0_1_base> + 374 _H	8, 16, 32	—
	GTM DMA selection control register 36	GTM_DMA_SEL36	<GTM0_1_base> + 378 _H	8, 16, 32	—
	GTM DMA selection control register 37	GTM_DMA_SEL37	<GTM0_1_base> + 37C _H	8, 16, 32	—
	GTM DMA selection control register 40	GTM_DMA_SEL40	<GTM0_1_base> + 380 _H	8, 16, 32	—
	GTM DMA selection control register 41	GTM_DMA_SEL41	<GTM0_1_base> + 384 _H	8, 16, 32	—
	GTM DMA selection control register 42	GTM_DMA_SEL42	<GTM0_1_base> + 388 _H	8, 16, 32	—
	GTM DMA selection control register 43	GTM_DMA_SEL43	<GTM0_1_base> + 38C _H	8, 16, 32	—
	GTM DMA selection control register 44	GTM_DMA_SEL44	<GTM0_1_base> + 390 _H	8, 16, 32	—
	GTM DMA selection control register 45	GTM_DMA_SEL45	<GTM0_1_base> + 394 _H	8, 16, 32	—
	GTM DMA selection control register 46	GTM_DMA_SEL46	<GTM0_1_base> + 398 _H	8, 16, 32	—
	GTM DMA selection control register 47	GTM_DMA_SEL47	<GTM0_1_base> + 39C _H	8, 16, 32	—
	GTM DMA selection control register 50	GTM_DMA_SEL50	<GTM0_1_base> + 3A0 _H	8, 16, 32	—
	GTM DMA selection control register 51	GTM_DMA_SEL51	<GTM0_1_base> + 3A4 _H	8, 16, 32	—
	GTM DMA selection control register 52	GTM_DMA_SEL52	<GTM0_1_base> + 3A8 _H	8, 16, 32	—
	GTM DMA selection control register 53	GTM_DMA_SEL53	<GTM0_1_base> + 3AC _H	8, 16, 32	—
	GTM DMA selection control register 54	GTM_DMA_SEL54	<GTM0_1_base> + 3B0 _H	8, 16, 32	—
	GTM DMA selection control register 55	GTM_DMA_SEL55	<GTM0_1_base> + 3B4 _H	8, 16, 32	—
	GTM DMA selection control register 56	GTM_DMA_SEL56	<GTM0_1_base> + 3B8 _H	8, 16, 32	—
	GTM DMA selection control register 57	GTM_DMA_SEL57	<GTM0_1_base> + 3BC _H	8, 16, 32	—

Table 31.11 List of Registers (6/7)

Module Name	Register Name	Symbol	Address	Access	Access Protection
GTM0_1	GTM DMA selection control register 60	GTM_DMA_SEL60	<GTM0_1_base> + 3C0 _H	8, 16, 32	—
	GTM DMA selection control register 61	GTM_DMA_SEL61	<GTM0_1_base> + 3C4 _H	8, 16, 32	—
	GTM DMA selection control register 62	GTM_DMA_SEL62	<GTM0_1_base> + 3C8 _H	8, 16, 32	—
	GTM DMA selection control register 63	GTM_DMA_SEL63	<GTM0_1_base> + 3CC _H	8, 16, 32	—
	GTM DMA selection control register 64	GTM_DMA_SEL64	<GTM0_1_base> + 3D0 _H	8, 16, 32	—
	GTM DMA selection control register 65	GTM_DMA_SEL65	<GTM0_1_base> + 3D4 _H	8, 16, 32	—
	GTM DMA selection control register 66	GTM_DMA_SEL66	<GTM0_1_base> + 3D8 _H	8, 16, 32	—
	GTM DMA selection control register 67	GTM_DMA_SEL67	<GTM0_1_base> + 3DC _H	8, 16, 32	—
	GTM DMA selection control register 70	GTM_DMA_SEL70	<GTM0_1_base> + 3E0 _H	8, 16, 32	—
	GTM DMA selection control register 71	GTM_DMA_SEL71	<GTM0_1_base> + 3E4 _H	8, 16, 32	—
	GTM DMA selection control register 72	GTM_DMA_SEL72	<GTM0_1_base> + 3E8 _H	8, 16, 32	—
	GTM DMA selection control register 73	GTM_DMA_SEL73	<GTM0_1_base> + 3EC _H	8, 16, 32	—
	GTM DMA selection control register 74	GTM_DMA_SEL74	<GTM0_1_base> + 3F0 _H	8, 16, 32	—
	GTM DMA selection control register 75	GTM_DMA_SEL75	<GTM0_1_base> + 3F4 _H	8, 16, 32	—
	GTM DMA selection control register 76	GTM_DMA_SEL76	<GTM0_1_base> + 3F8 _H	8, 16, 32	—
	GTM DMA selection control register 77	GTM_DMA_SEL77	<GTM0_1_base> + 3FC _H	8, 16, 32	—
	GTM DMA selection control register 80	GTM_DMA_SEL80	<GTM0_1_base> + 400 _H	8, 16, 32	—
	GTM DMA selection control register 81	GTM_DMA_SEL81	<GTM0_1_base> + 404 _H	8, 16, 32	—
	GTM DMA selection control register 82	GTM_DMA_SEL82	<GTM0_1_base> + 408 _H	8, 16, 32	—
	GTM DMA selection control register 83	GTM_DMA_SEL83	<GTM0_1_base> + 40C _H	8, 16, 32	—
	GTM DMA selection control register 84	GTM_DMA_SEL84	<GTM0_1_base> + 410 _H	8, 16, 32	—
	GTM DMA selection control register 85	GTM_DMA_SEL85	<GTM0_1_base> + 414 _H	8, 16, 32	—
	GTM DMA selection control register 86	GTM_DMA_SEL86	<GTM0_1_base> + 418 _H	8, 16, 32	—
	GTM DMA selection control register 87	GTM_DMA_SEL87	<GTM0_1_base> + 41C _H	8, 16, 32	—
	GTM DMA Common selection function Register 0	GTM_DMA_COM_SEL0	<GTM0_1_base> + 420 _H	8, 16, 32	—
	GTM DMA Common selection function Register 1	GTM_DMA_COM_SEL1	<GTM0_1_base> + 424 _H	8, 16, 32	—
	GTM DMA Common selection function Register 2	GTM_DMA_COM_SEL2	<GTM0_1_base> + 428 _H	8, 16, 32	—
	GTM DMA Common selection function Register 3	GTM_DMA_COM_SEL3	<GTM0_1_base> + 42C _H	8, 16, 32	—
	GTM DMA Common selection function Register 4	GTM_DMA_COM_SEL4	<GTM0_1_base> + 430 _H	8, 16, 32	—
	GTM DMA Common selection function Register 5	GTM_DMA_COM_SEL5	<GTM0_1_base> + 434 _H	8, 16, 32	—
	GTM DMA Common selection function Register 6	GTM_DMA_COM_SEL6	<GTM0_1_base> + 438 _H	8, 16, 32	—
	GTM DMA Common selection function Register 7	GTM_DMA_COM_SEL7	<GTM0_1_base> + 43C _H	8, 16, 32	—
GTM DMA Common selection function Register 8	GTM_DMA_COM_SEL8	<GTM0_1_base> + 440 _H	8, 16, 32	—	
GTM DMA Common selection function Register 9	GTM_DMA_COM_SEL9	<GTM0_1_base> + 444 _H	8, 16, 32	—	

Table 31.11 List of Registers (7/7)

Module Name	Register Name	Symbol	Address	Access	Access Protection
GTM0_1	GTM ADCI channel selection register 0	GTM_ADCI_CHSEL0	<GTM0_1_base> + 500 _H	8, 16, 32	—
	GTM ADCI channel selection register 1	GTM_ADCI_CHSEL1	<GTM0_1_base> + 504 _H	8, 16, 32	—
	GTM ADCI channel selection register 2	GTM_ADCI_CHSEL2	<GTM0_1_base> + 508 _H	8, 16, 32	—
	GTM ADCI channel selection register 3	GTM_ADCI_CHSEL3	<GTM0_1_base> + 50C _H	8, 16, 32	—
	GTM ADCI channel selection register 4	GTM_ADCI_CHSEL4	<GTM0_1_base> + 510 _H	8, 16, 32	—

Note 1. For details of GTM-IP registers, please refer to "*GTM-IP Specification*" documents. Also, the access size of RAM has to be 32-bit wide.

The table below shows registers that have a different reset value compared to Robert Bosch's GTM-IP specification.

Table 31.12 Initial Value Difference between "GTM-IP Specification Appendix B" and Actual Device

Register Name	Reset Value	
	GTM-IP Specification	GTM-IP_355, GTM-IP_353
GTM_IRQ_MODE	0000 0000 _H	0000 0002 _H
GTM_BRIDGE_PTR1	0040 0000 _H	004X XXXX _H
CCM[i]_HW_CONF (i = 0 to 9)	284F 1339 _H	284F 0339 _H
CMP_IRQ_MODE	0000 0000 _H	0000 0002 _H
ARU_IRQ_MODE	0000 0000 _H	0000 0002 _H
BRC_IRQ_MODE	0000 0000 _H	0000 0002 _H
SPE[i]_IRQ_MODE	0000 0000 _H	0000 0002 _H
TIM[i]_CH[x]_IRQ_MODE	0000 0000 _H	0000 0002 _H
TOM[i]_CH[x]_IRQ_MODE	0000 0000 _H	0000 0002 _H
ATOM[i]_CH[x]_IRQ_MODE	0000 0000 _H	0000 0002 _H
FIFO[i]_CH[x]_IRQ_MODE	0000 0000 _H	0000 0002 _H
DPLL_IRQ_MODE	0000 0000 _H	0000 0002 _H
MCS[i]_CH[x]_IRQ_MODE	0000 0000 _H	0000 0002 _H
MCS[i]_CTRL_STAT	0001 0000 _H	0000 0000 _H

31.3.2 GTM_IRQ_SEL_{ij} — GTM Interrupt Selection Control Register ij

GTM interrupt factors forwarded to INTC are selected from ICM outputs. Please refer to ICM Chapter in “*GTM-IP Specification*” for list of ICM outputs. (i = 0 to 7, j = 00 to 15)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	GTM_SPE[i]_IRQ	GTM_MCS[i]_IRQ[7:0]							GTM_ATOM[i]_IRQ[3:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GTM_TOM[i]_IRQ[7:0]							GTM_TIM[i]_IRQ[7:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.13 GTM_IRQ_SEL_{ij} Register Contents

Bit Position	Bit Name	Function
31 to 29	—	Reserved When writing to these bits, write 0.
28	GTM_SPE[i]_IRQ	SPE[i] Shared interrupt (GTM_SPE[i]_IRQ) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
27 to 20	GTM_MCS[i]_IRQ [7:0]	MCS[i] Interrupt for channel 7 to 0 (GTM_MCS[i]_IRQ[7:0]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
19 to 16	GTM_ATOM[i]_IRQ [3:0]	ATOM[i] Shared interrupts for 3 to 0 (GTM_ATOM[i]_IRQ[3:0]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
15 to 8	GTM_TOM[i]_IRQ [7:0]	TOM[i] Shared interrupts for 7 to 0 (GTM_TOM[i]_IRQ[7:0]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
7 to 0	GTM_TIM[i]_IRQ [7:0]	TIM[i] Shared interrupts for 7 to 0 (GTM_TIM[i]_IRQ[7:0]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.

Note: i: Number of clusters, j: Number of interrupts per cluster

The bit of cluster i beyond the equipped maximum number of TIM/TOM/ATOM/MCS/SPE will be Reserved.

31.3.3 GTM_IRQ_SEL8j — GTM Interrupt Selection Control Register 8j

GTM interrupt factors forwarded to INTC are selected from ICM outputs. Please refer to ICM Chapter in “*GTM-IP Specification*” for list of ICM outputs. (j = 00 to 15)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTENCA[n]IEC	INTENCA[n]1	INTENCA[n]0	INTENCA[n]UD	INTENCA[n]IOV	—	—	—	—	—	—	—	GTM_ATOM8_IRQ[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GTM_PSM1_IRQ[7:0]							GTM_PSM0_IRQ[7:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.14 GTM_IRQ_SEL8j Register Contents

Bit Position	Bit Name	Function
31	INTENCA[n]IEC	Encoder clear interrupt (n = 0: j = 00 to 07, n = 1: j = 08 to 15) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
30	INTENCA[n]1	Capture / compare match interrupt 1 (n = 0: j = 00 to 07, n = 1: j = 08 to 15) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
29	INTENCA[n]0	Capture / compare match interrupt 0 (n = 0: j = 00 to 07, n = 1: j = 08 to 15) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
28	INTENCA[n]UD	Underflow interrupt (n = 0: j = 00 to 07, n = 1: j = 08 to 15) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
27	INTENCA[n]IOV	Overflow interrupt (n = 0: j = 00 to 07, n = 1: j = 08 to 15) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
26 to 20	—	Reserved When writing to these bits, write 0.
19 to 16	GTM_ATOM8_IRQ [3:0]	ATOM8 Shared interrupts for 3 to 0 (GTM_ATOM8_IRQ[3:0]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
15 to 8	GTM_PSM1_IRQ[7:0]	PSM1 Shared interrupts (GTM_PSM1_IRQ[7:0]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
7 to 0	GTM_PSM0_IRQ[7:0]	PSM0 Shared interrupts (GTM_PSM0_IRQ[7:0]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.

Note: j: Number of interrupts per cluster, n: Number of Encode Timer (ENCA_n)

31.3.4 GTM_IRQ_DPLL_SELj — GTM Interrupt DPLL Selection Register j

DPLL interrupt factors forwarded to INTC are selected from ICM outputs. Please refer to ICM Chapter in “*GTM-IP Specification*” for list of ICM outputs. (j = 0 to 15)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GTM_DPLL_IRQ[26:16]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GTM_DPLL_IRQ[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.15 GTM_IRQ_DPLL_SELj Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 27	—	Reserved When writing to these bits, write 0.
26	GTM_DPLL_IRQ[26]	DPLL_SORI: STATE out of range interrupt (GTM_DPLL_IRQ[26]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
25	GTM_DPLL_IRQ[25]	DPLL_TORI: TRIGGER out of range interrupt (GTM_DPLL_IRQ[25]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
24	GTM_DPLL_IRQ[24]	DPLL_CDSI: DPLL calculated duration interrupt for STATE (GTM_DPLL_IRQ[24]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
23	GTM_DPLL_IRQ[23]	DPLL_CDTI: DPLL calculated duration interrupt for TRIGGER (GTM_DPLL_IRQ[23]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
22	GTM_DPLL_IRQ[22]	DPLL_TE4I: DPLL TRIGGER event interrupt 4 (GTM_DPLL_IRQ[22]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
21	GTM_DPLL_IRQ[21]	DPLL_TE3I: DPLL TRIGGER event interrupt 3 (GTM_DPLL_IRQ[21]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.

Table 31.15 GTM_IRQ_DPLL_SELj Register Contents (2/3)

Bit Position	Bit Name	Function
20	GTM_DPLL_IRQ[20]	DPLL_TE2: DPLL TRIGGER event interrupt 2 (GTM_DPLL_IRQ[20]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
19	GTM_DPLL_IRQ[19]	DPLL_TE1: DPLL TRIGGER event interrupt 1 (GTM_DPLL_IRQ[19]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
18	GTM_DPLL_IRQ[18]	DPLL_TE0: DPLL TRIGGER event interrupt 0 (GTM_DPLL_IRQ[18]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
17	GTM_DPLL_IRQ[17]	DPLL_LL2: DPLL Lost of lock interrupt for SUB_INC2 (GTM_DPLL_IRQ[17]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
16	GTM_DPLL_IRQ[16]	DPLL_GL2: DPLL Get of lock interrupt for SUB_INC2 (GTM_DPLL_IRQ[16]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
15	GTM_DPLL_IRQ[15]	DPLL_EI: DPLL Error interrupt (GTM_DPLL_IRQ[15]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
14	GTM_DPLL_IRQ[14]	DPLL_LL1: DPLL Lost of lock interrupt for SUB_INC1 (GTM_DPLL_IRQ[14]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
13	GTM_DPLL_IRQ[13]	DPLL_GL1: DPLL Get of lock interrupt for SUB_INC1 (GTM_DPLL_IRQ[13]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
12	GTM_DPLL_IRQ[12]	DPLL_W1: DPLL Write access to RAM region 1b or 1c int. (GTM_DPLL_IRQ[12]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
11	GTM_DPLL_IRQ[11]	DPLL_W2: DPLL Write access to RAM region 2 interrupt (GTM_DPLL_IRQ[11]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
10	GTM_DPLL_IRQ[10]	DPLL_PWI: DPLL Plausibility window (PVT) viol. int. of TRIG. (GTM_DPLL_IRQ[10]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
9	GTM_DPLL_IRQ[9]	DPLL_TASI: DPLL TRIG. active slope detected while NTI_CNT is 0 (GTM_DPLL_IRQ[9]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
8	GTM_DPLL_IRQ[8]	DPLL_SASI: DPLL STATE active slope detected (GTM_DPLL_IRQ[8]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
7	GTM_DPLL_IRQ[7]	DPLL_MTI: DPLL Missing TRIGGER interrupt (GTM_DPLL_IRQ[7]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
6	GTM_DPLL_IRQ[6]	DPLL_MSI: DPLL Missing STATE interrupt (GTM_DPLL_IRQ[6]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
5	GTM_DPLL_IRQ[5]	DPLL_TISI: DPLL TRIGGER inactive slope detected (GTM_DPLL_IRQ[5]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
4	GTM_DPLL_IRQ[4]	DPLL_SISI: DPLL STATE inactive slope detected (GTM_DPLL_IRQ[4]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.

Table 31.15 GTM_IRQ_DPLL_SELj Register Contents (3/3)

Bit Position	Bit Name	Function
3	GTM_DPLL_IRQ[3]	DPLL_TAXI: DPLL TRIG. max. hold time (THMA) viol. detected (GTM_DPLL_IRQ[3]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
2	GTM_DPLL_IRQ[2]	DPLL_TINI: DPLL TRIG. min. hold time (THMI) viol. detected (GTM_DPLL_IRQ[2]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
1	GTM_DPLL_IRQ[1]	DPLL_EDI: DPLL enable or disable interrupt (GTM_DPLL_IRQ[1]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.
0	GTM_DPLL_IRQ[0]	DPLL_DCGI: DPLL direction change interrupt (GTM_DPLL_IRQ[0]) 0: Interrupt will not be forwarded to INTC. 1: Interrupt will be forwarded to INTC.

Note: j: Number of DPLL interrupts

31.3.5 GTM_DMA_SEL_{ij} — GTM DMA Selection Control Register ij

GTM outputs and ICM outputs can be forwarded to DMAC/DTS. Please refer to ICM, TOM and ATOM Sections in “GTM-IP Specification” for list of ICM outputs. (i = 0 to 7, j = 0 to 7)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	EDGSEL[1:0]		GTM_DMA_SEL[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.16 GTM_DMA_SEL_{ij} Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When writing to these bits, write 0.
7 to 6	EDGSEL[1:0]	Select an effective edge of GTM Timer Output GTM_TOM[i]_OUT[x] (x = 0 to 7) and GTM_ATOM[i]_OUT[x] (x = 0 to 7) 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited)
5 to 0	GTM_DMA_SEL[5:0]	Selects a DMA trigger from ICM outputs or GTM outputs. 0 + x: GTM_TIM[i]_IRQ[x] (x = 0 to 7) 8 + x: GTM_TOM[i]_IRQ[x] (x = 0 to 7) 16 + x: GTM_TOM[i]_OUT[x] (x = 0 to 15) 32 + x: GTM_ATOM[i]_IRQ[x] (x = 0 to 3) 36 + x: GTM_ATOM[i]_OUT[x] (x = 0 to 7) 44 + x: GTM_MCS[i]_IRQ[x] (x = 0 to 7) 52: GTM_SPE[i]_IRQ Other: Reserved

Note: i: Number of clusters, j: Number of DMA triggers per cluster

The bit of cluster i beyond the equipped maximum number of TIM/TOM/ATOM/MCS/SPE will be Reserved

31.3.6 GTM_DMA_SEL8j — GTM DMA Selection Control Register 8j

GTM interrupt factors forwarded to DMAC/DTS are selected from ICM outputs. Please refer to ICM Chapter in “*GTM-IP Specification*” for list of ICM outputs. (j=0 to 7)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	GTM_DMA_SEL[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.17 GTM_DMA_SEL8j Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved When writing to these bits, write 0.
5 to 0	GTM_DMA_SEL[5:0]	Selects a DMA trigger from ICM. 16 + x: GTM_ATOM8_IRQ[x] (x = 0 to 3) 32: ENCA0 INTENCA0IOV Overflow interrupt 33: ENCA0 INTENCA0IUD Underflow interrupt 34: ENCA0 INTENCA0I0 Capture / compare match interrupt 0 35: ENCA0 INTENCA0I1 Capture / compare match interrupt 1 36: ENCA0 INTENCA0IEC Encoder clear interrupt 37: ENCA1 INTENCA1IOV Overflow interrupt 38: ENCA1 INTENCA1IUD Underflow interrupt 39: ENCA1 INTENCA1I0 Capture / compare match interrupt 0 40: ENCA1 INTENCA1I1 Capture / compare match interrupt 1 41: ENCA1 INTENCA1IEC Encoder clear interrupt Other: Reserved

Note: j: Number of DMA triggers per cluster

31.3.7 GTM_DMA_COM_SELj — GTM DMA Common Selection Function Register j

GTM-IP interrupt factors forwarded to DMAC/DTS are selected from ICM outputs. Please refer to ICM Chapter in “GTM-IP Specification” for list of ICM outputs. (j = 0 to 9)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	GTM_DMA_COM_SEL[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.18 GTM_DMA_COM_SELj Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved When writing to these bits, write 0.
5 to 0	GTM_DMA_COM_SEL[5:0]	Selects a DMA trigger from ICM. 0: GTM_AEI_IRQ 1 + x: GTM_ARU_IRQ[x] (x = 0 to 2) 4: GTM_BRC_IRQ 5: GTM_CMP_IRQ 6 + x: GTM_PSM0_IRQ[x] (x = 0 to 7) 14 + x: GTM_PSM1_IRQ[x] (x = 0 to 7) 22 + x: GTM_DPLL_IRQ[x] (x = 0 to 1) 20 + x: GTM_DPLL_IRQ[x] (x = 4 to 14) 19 + x: GTM_DPLL_IRQ[x] (x = 16 to 26) Other: Reserved

Note: j: Number of DMA triggers per cluster

31.3.8 GTM_ADCI_CHSELn — GTM ADCI channel selection register n

Results of A/D conversion are stored in ADC_CH[y]_DATA registers ($y = 0..31$). Either registers 0 to 15 or registers 16 to 31 can be used based on GTM_ADCI_CHSELn.CHS. In addition, a register designation is also based on DFTAG (ADCH and CADC) or DFES (DSADC). For more information about DFTAG and DFES configuration please refer to **Section 34, Analog to Digital Converter (ADCH)**, **Section 35, Delta-Sigma Analog to Digital Converter (DSADC)** and **Section 36, Cyclic Analog to Digital Converter (CADC)**. ($n = 0$ to 3: ADCHn, $n = 4$: CADC)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CHS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 31.19 GTM_ADCI_CHSELn Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When writing to these bits, write 0.
0	CHS	The analog to digital conversion data storage territory related in DFTAG of ADCH is designated. 0: AD data is stored in ADC_CH[y]_DATA ($y = 0$ to 15) 1: AD data is stored in ADC_CH[y]_DATA ($y = 16$ to 31)

31.4 Operation

31.4.1 Peripheral Bus Access

GTM is connected to a system bus as a slave. Access by CPU or DMA to registers and RAM (MCS and DPLL) is provided through AEI interface. Only 32 bit access to GTM registers and RAM is supported.

GTM MCS and DPLL RAM are not shared with other functionality. They can be accessed via system bus in a certain address range. For details of the memory map, please refer to **Figure 31.2**, **Figure 31.3**.

GTM is accessible by all cores with the same access speed except the ICU-M core. Please refer to **Section 3, CPU System** for more details on CPU to peripheral access speed.

	RH850/E2x-FCC1	RH850/E2M
FF80_0000 _h	GTM Global	GTM Global
FF80_002C _h		
FF80_0030 _h	BRIDGE	BRIDGE
FF80_00FC _h		
FF80_0100 _h	TBU	TBU
FF80_017C _h		
FF80_0180 _h	MON	MON
FF80_019C _h		
FF80_0200 _h	CMP	CMP
FF80_027C _h		
FF80_0280 _h	ARU	ARU
FF80_02FC _h		
FF80_0300 _h	CMU	CMU
FF80_03FC _h		
FF80_0400 _h	BRC	BRC
FF80_05FC _h		
FF80_0600 _h	ICM	ICM
FF80_07FC _h		
FF80_0800 _h	SPE0 to 1	SPE0 to 1
FF80_08FC _h		
FF80_0900 _h	SPE2 to 3	Access prohibited
FF80_09FC _h		
FF80_0A00 _h	Access prohibited	
FF80_0EFC _h		
FF80_0F00 _h	MAP	MAP
FF80_0F3C _h		
FF80_0F40 _h	MCFG	Access prohibited
FF80_0FFC _h		
FF80_1000 _h	TIM0 to 2	TIM0 to 2
FF80_27FC _h		
FF80_2800 _h	TIM3 to 7	TIM3 to 7
FF80_4FFC _h		
FF80_5000 _h	Access prohibited	Access prohibited
FF80_7FFC _h		
FF80_8000 _h	TOM0 to 1	TOM0 to 1
FF80_8FFC _h		
FF80_9000 _h	TOM2 to 3	TOM2 to 3
FF80_9FFC _h		
FF80_A000 _h	TOM4	Access prohibited
FF80_A7FC _h		
FF80_A800 _h	Access prohibited	
FF81_7FFC _h		

Figure 31.2 Memory Map - GTM (1/2)

	RH850/E2x-FCC1	RH850/E2M
FF81_8000 _h	F2A0	F2A0
FF81_807C _h		
FF81_8080 _h	AFD0	AFD0
FF81_83FC _h		
FF81_8400 _h	FIFO0	FIFO0
FF81_8FFC _h		
FF81_9000 _h	FIFO0 MEMORY	FIFO0 MEMORY
FF81_9FFC _h		
FF81_A000 _h	Access prohibited	Access prohibited
FF81_BFFC _h		
FF81_C000 _h	F2A1	F2A1
FF81_C07C _h		
FF81_C080 _h	AFD1	AFD1
FF81_C3FC _h		
FF81_C400 _h	FIFO1	FIFO1
FF81_CFFC _h		
FF81_D000 _h	FIFO1 MEMORY	FIFO1 MEMORY
FF81_DFFC _h		
FF81_E000 _h	Access prohibited	Access prohibited
FF82_7FFC _h		
FF82_8000 _h	DPLL	DPLL
FF82_81FC _h		
FF82_8200 _h	DPLL RAM1A	DPLL RAM1A
FF82_83FC _h		
FF82_8400 _h	DPLL RAM1B	DPLL RAM1B
FF82_85FC _h		
FF82_8600 _h	DPLL RAM1C	DPLL RAM1C
FF82_89FC _h		
FF82_8A00 _h	Access prohibited	Access prohibited
FF82_8DFC _h		
FF82_8E00 _h	DPLL	DPLL
FF82_8F7C _h		
FF82_8F80 _h	Access prohibited	Access prohibited
FF82_BFFC _h		
FF82_C000 _h	DPLL RAM2	DPLL RAM2
FF82_DFFC _h		
FF82_E000 _h		Access prohibited
FF82_FFFC _h		
FF83_0000 _h	Access prohibited	
FF83_7FFC _h		
FF83_8000 _h	MCS RAM	MCS RAM
FF86_AFFC _h	Refer to "Memory Map - GTM RAM"	Refer to "Memory Map - GTM RAM"
FF86_B000 _h	Access prohibited	Access prohibited
FF86_FFFC _h		
FF8E_2000 _h	CCM0 to 3	CCM0 to 3
FF8E_27FC _h		
FF8E_2800 _h	CCM4 to 7	CCM4 to 7
FF8E_2FFC _h		
FF8E_3000 _h	CCM8	Access prohibited
FF8E_31FC _h		
FF8E_3200 _h	Access prohibited	
FF8E_3FFC _h		
FF8E_4000 _h	CDTM[i]_DTM[j]	CDTM[i]_DTM[j]
FF8E_593C _h		
FF8E_5940 _h	Access prohibited	Access prohibited
FF8E_7FFC _h		
FF8E_8000 _h	ATOM0 to 3	ATOM0 to 3
FF8E_9FFC _h		
FF8E_A000 _h	ATOM4 to 7	ATOM4 to 7
FF8E_BFFC _h		
FF8E_C000 _h	ATOM8	Access prohibited
FF8E_C7FC _h		
FF8E_C800 _h	Access prohibited	
FF8E_FFFC _h		
FF8F_0000 _h	MCS0 to 2	MCS0 to 2
FF8F_2FFC _h		
FF8F_3000 _h	MCS3 to 6	MCS3 to 6
FF8F_6FFC _h		

Figure 31.2 Memory Map - GTM (2/2)

RH850/E2x-FCC1		RH850/E2M	
	GTM_RAM_HALF_MODE ¹ = 0	GTM_RAM_HALF_MODE ¹ = 1	
FF83_8000 _H	MCS0_RAM0(8KB)	MCS0_RAM0(8KB)	MCS0_RAM0(8KB)
FF83_9FFC _H			
FF83_A000 _H	MCS0_RAM1(4KB)	MCS0_RAM1(4KB)	MCS0_RAM1(4KB)
FF83_AFFC _H			
FF83_B000 _H	Access prohibited	Access prohibited	Access prohibited
FF83_FFFC _H			
FF84_0000 _H	MCS1_RAM0(8KB)	MCS1_RAM0(8KB)	MCS1_RAM0(8KB)
FF84_1FFC _H			
FF84_2000 _H	MCS1_RAM1(4KB)	MCS1_RAM1(4KB)	MCS1_RAM1(4KB)
FF84_2FFC _H			
FF84_3000 _H	Access prohibited	Access prohibited	Access prohibited
FF84_7FFC _H			
FF84_8000 _H	MCS2_RAM0(8KB)		
FF84_8FFC _H			
FF84_9000 _H		MCS2_RAM0(4KB)	MCS2_RAM0(4KB)
FF84_9FFC _H			
FF84_A000 _H	MCS2_RAM1(4KB)	MCS2_RAM1(2KB)	MCS2_RAM1(2KB)
FF84_A7FC _H			
FF84_A800 _H		Access prohibited	Access prohibited
FF84_AFFC _H			
FF84_B000 _H	Access prohibited		
FF84_FFFC _H			

Note 1. Refer to **Section 41, Flash Memory Configuration Setting Area (Option Bytes, Reset Vector)** about GTM_HALF_MODE change method of E2x-FCC1.

Figure 31.3 Memory Map - GTM RAM (1/3)

RH850/E2x-FCC1		RH850/E2M	
GTM_RAM_HALF_MODE ¹ = 0		GTM_RAM_HALF_MODE ¹ = 1	
FF85_0000 _n	MCS3_RAM0(8KB)	Access prohibited	Access prohibited
FF85_0FFC _n FF85_1000 _n	MCS3_RAM0(8KB)	MCS3_RAM0(4KB)	MCS3_RAM0(4KB)
FF85_1FFC _n FF85_2000 _n		MCS3_RAM1(2KB)	MCS3_RAM1(2KB)
FF85_27FC _n FF85_2800 _n	MCS3_RAM1(4KB)	MCS3_RAM1(2KB)	MCS3_RAM1(2KB)
FF85_2FFC _n FF85_3000 _n	Access prohibited	Access prohibited	Access prohibited
FF85_7FFC _n FF85_8000 _n		MCS4_RAM0(8KB)	
FF85_8FFC _n FF85_9000 _n	MCS4_RAM0(8KB)	MCS4_RAM0(4KB)	MCS4_RAM0(4KB)
FF85_9FFC _n FF85_A000 _n		MCS4_RAM1(2KB)	MCS4_RAM1(2KB)
FF85_A7FC _n FF85_A800 _n	MCS4_RAM1(4KB)	MCS4_RAM1(2KB)	MCS4_RAM1(2KB)
FF85_AFFC _n FF85_B000 _n	Access prohibited	Access prohibited	Access prohibited
FF85_BFFC _n FF86_0000 _n		MCS5_RAM0(8KB)	
FF86_0FFC _n FF86_1000 _n	MCS5_RAM0(8KB)	MCS5_RAM0(4KB)	MCS5_RAM0(4KB)
FF86_1FFC _n FF86_2000 _n		MCS5_RAM1(2KB)	MCS5_RAM1(2KB)
FF86_27FC _n FF86_2800 _n	MCS5_RAM1(4KB)	MCS5_RAM1(2KB)	MCS5_RAM1(2KB)
FF86_2FFC _n FF86_3000 _n	Access prohibited	Access prohibited	Access prohibited
FF86_7FFC _n FF86_8000 _n		MCS6_RAM0(8KB)	
FF86_8FFC _n FF86_9000 _n	MCS6_RAM0(8KB)	MCS6_RAM0(4KB)	MCS6_RAM0(4KB)
FF86_9FFC _n FF86_A000 _n		MCS6_RAM1(2KB)	MCS6_RAM1(2KB)
FF86_A7FC _n FF86_A800 _n	MCS6_RAM1(4KB)	MCS6_RAM1(2KB)	MCS6_RAM1(2KB)
FF86_AFFC _n FF86_B000 _n	Access prohibited	Access prohibited	Access prohibited
FF86_BFFC _n			

Note 1. Refer to **Section 41, Flash Memory Configuration Setting Area (Option Bytes, Reset Vector)** about GTM_HALF_MODE change method of E2x-FCC1.

Figure 31.3 Memory Map - GTM RAM (2/3)

	RH850/E2x-FCC1	RH850/E2M
FF81_9000 _h FF81_9FFC _h	FIFO0_MEMORY(3.625KB)	FIFO0_MEMORY(3.625KB)
FF81_D000 _h FF81_DFFC _h	FIFO1_MEMORY(3.625KB)	FIFO1_MEMORY(3.625KB)
FF82_8200 _h FF82_83FC _h	DPLL_RAM1A(0.375KB)	DPLL_RAM1A(0.375KB)
FF82_8400 _h FF82_89FC _h	DPLL_RAM1B(1.125KB) (include DPLL_RAM1C)	DPLL_RAM1B(1.125KB) (include DPLL_RAM1C)
FF82_C000 _h FF82_DFFC _h FF82_E000 _h FF82_FFFC _h	DPLL_RAM2(12KB)	DPLL_RAM2(6KB) Access prohibited

Figure 31.3 Memory Map - GTM RAM (3/3)

31.4.2 Reset

GTM is resettable by a dedicated bit, which allows the software to set GTM in reset state without using external reset.

Please refer to “GTM-IP Specification” document.

31.4.3 Interconnections of Interrupt Factors

GTM internal interrupt factors are consolidated by GTM Interrupt Concentrator Module (ICM) into fewer interrupts visible outside GTM. The functionality of ICM is described in GTM-IP Specification. For list of interrupt factors visible outside GTM-IP chapter see **Section 31.1.5, Interrupt Requests**.

Outputs of ICM are further multiplexed outside GTM and then routed to interrupt controller (INTC) and DMA controller (DMAC/DTS). This multiplexing is described here. For register definitions please see **Section 31.3.1, List of Registers**.

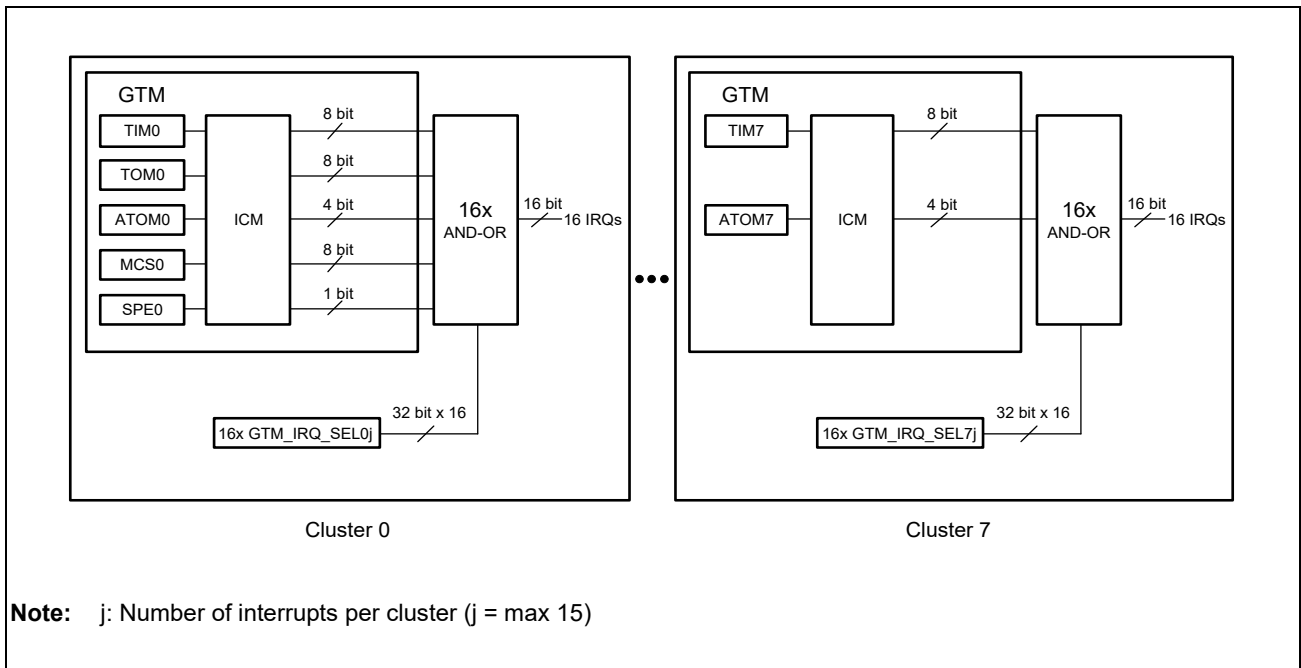


Figure 31.4 Interrupt Signal Bundling (1/6)

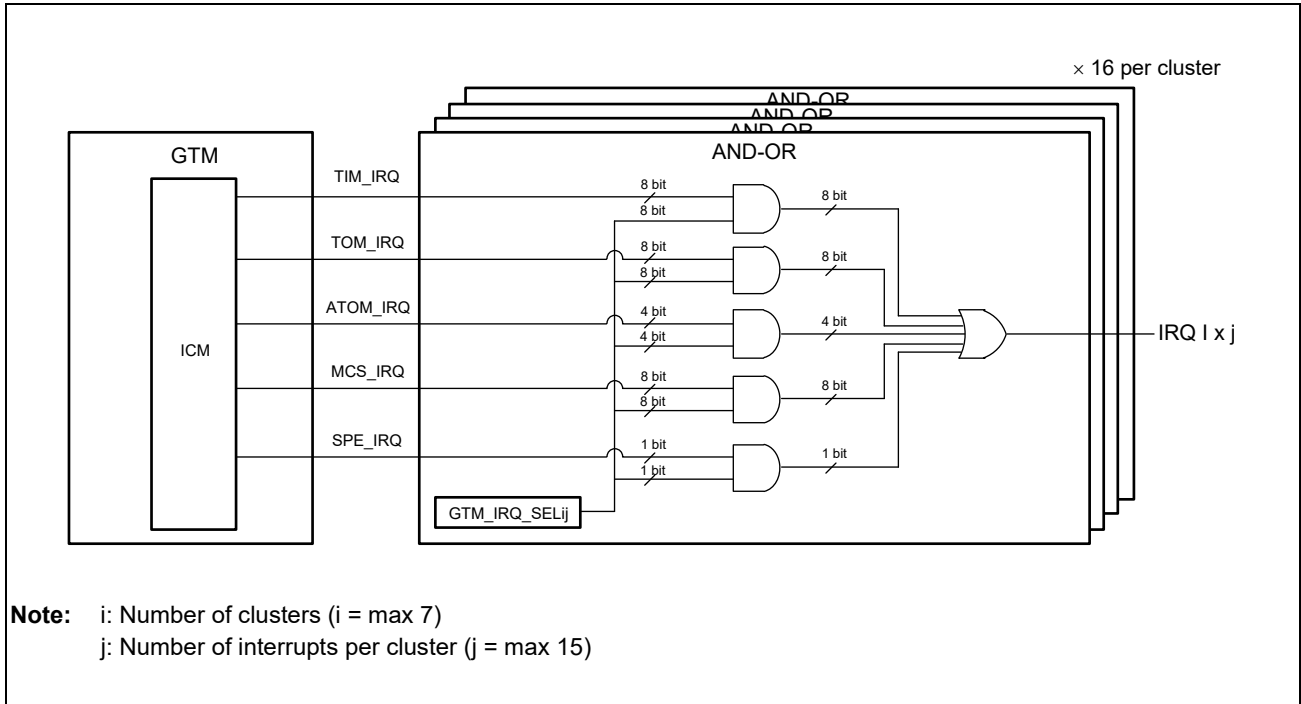


Figure 31.4 Interrupt Signal Bundling (2/6)

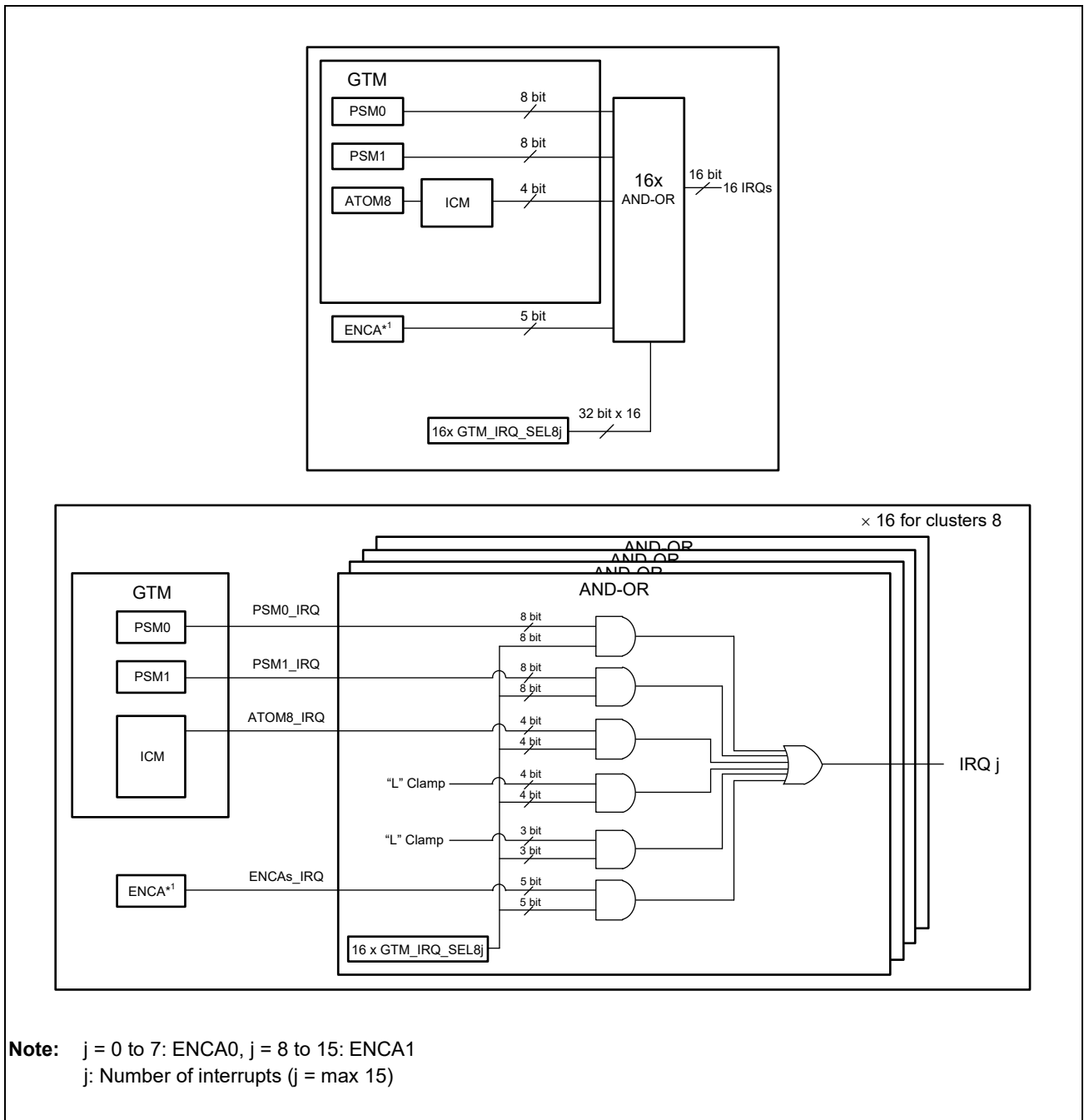


Figure 31.4 Interrupt Signal Bundling (3/6)

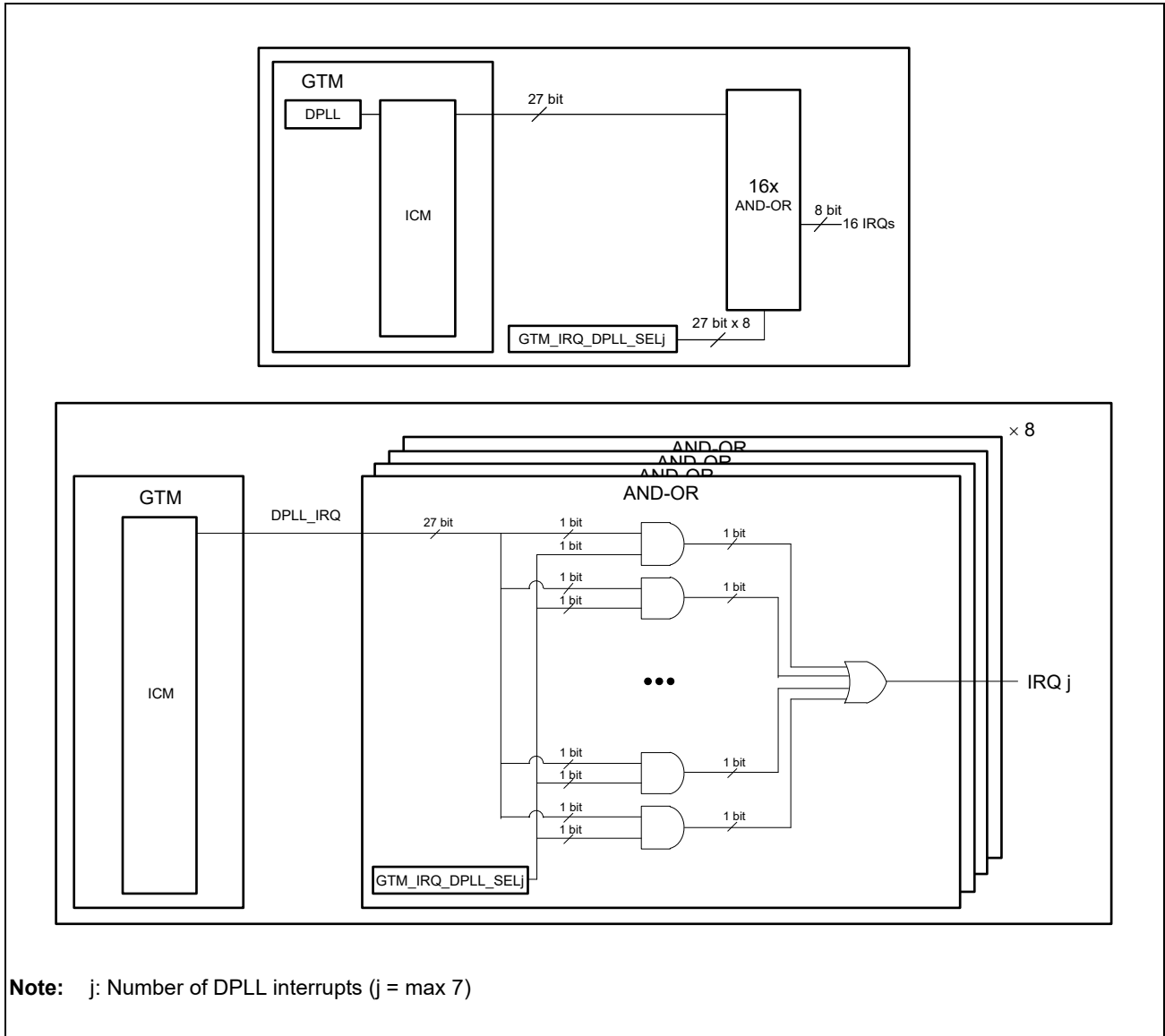


Figure 31.4 Interrupt Signal Bundling (4/6)

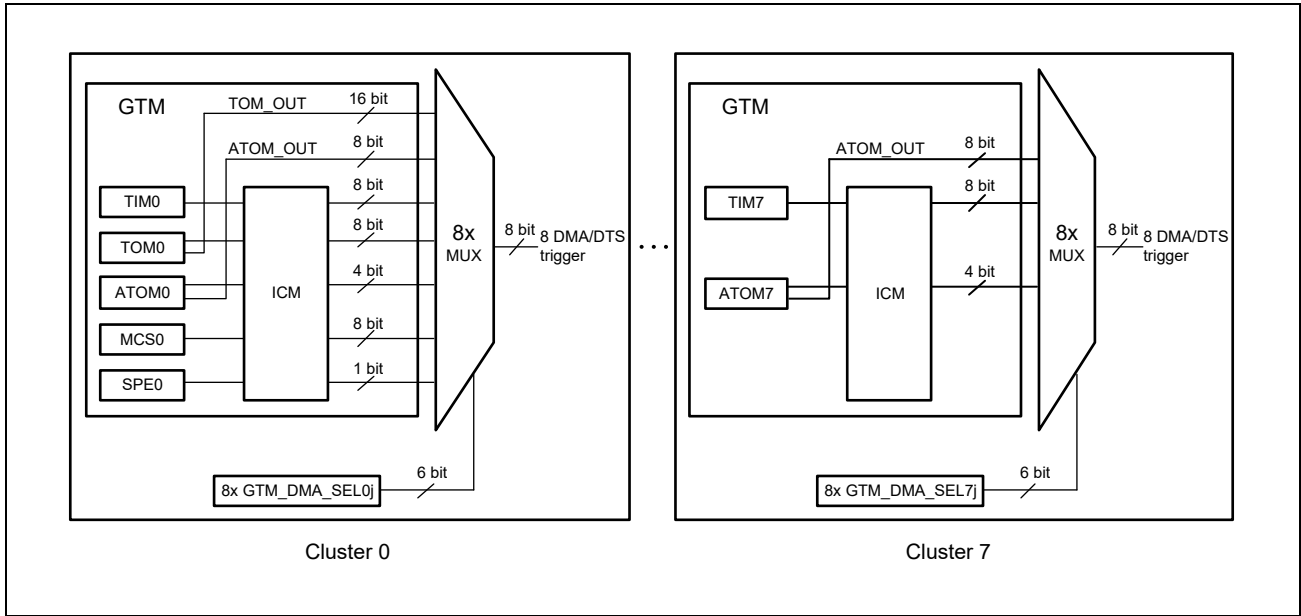


Figure 31.4 Interrupt Signal Bundling (5/6)

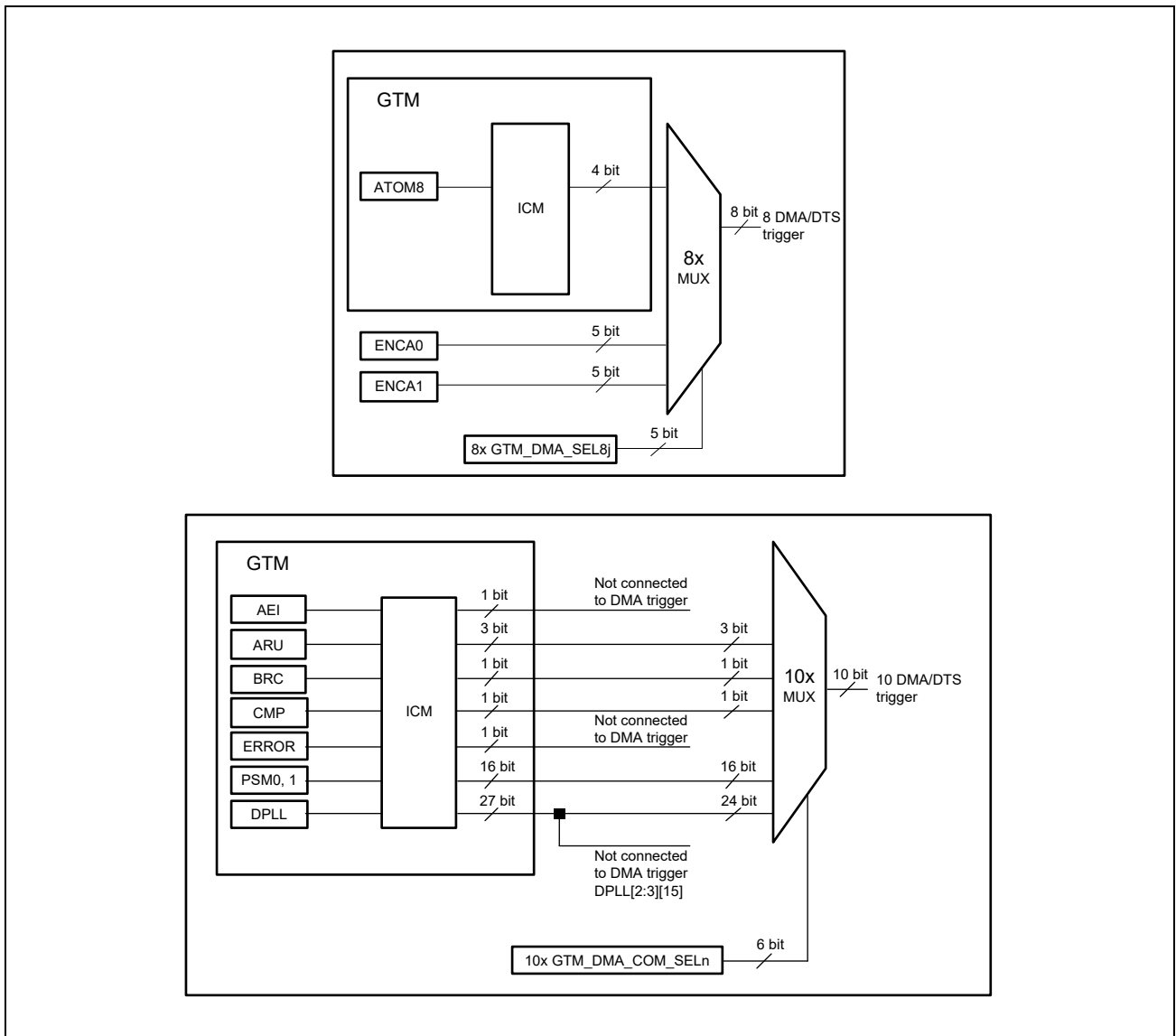


Figure 31.4 Interrupt Signal Bundling (6/6)

31.4.4 Outputs to AD

31.4.4.1 Outputs to ADCH

GTM outputs can trigger start of ADCH conversion. The following GTM outputs can be used as ADCH trigger sources:

- TOM_OUT
- ATOM_OUT

Signals specified above are multiplexed together and routed to the ADCH. Multiplexing scheme is described in **Section 33, Peripheral Interconnection (PIC)**. The GTM can trigger AD groups 0 to 4. TOM_OUT and ATOM_OUT are described in “*GTM-IP Specification*”. For more information about ADCH functionality please refer to **Section 34, Analog to Digital Converter (ADCH)**.

31.4.4.2 Outputs to DSADC

GTM outputs can trigger

- Starting and stopping of DSADC conversion
- Incrementing of DSADC input channel
- Gate reading of DSADC samples and time stamps

The following GTM outputs can be used as DSADC trigger sources

- TOM_OUT
- ATOM_OUT

Signals specified above are multiplexed together and routed to the DSADC. Multiplexing scheme is described in **Section 33, Peripheral Interconnection (PIC)**. All DSADC triggers can be mapped to GTM outputs. TOM_OUT and ATOM_OUT are described in “*GTM-IP Specification*”. For more information about DSADC functions please refer to **Section 35, Delta-Sigma Analog to Digital Converter (DSADC)**.

31.4.4.3 Outputs to CADC

GTM outputs can trigger

- Starting and stopping of CADC conversion
- Incrementing of CADC input channel
- Gate reading of CADC samples

The following GTM outputs can be used as CADC trigger sources

- TOM_OUT
- ATOM_OUT

Signals specified above are multiplexed together and routed to the CADC. Each CADC trigger can be mapped to GTM outputs. Multiplexing scheme is described in **Section 33, Peripheral Interconnection (PIC)**. TOM_OUT and ATOM_OUT signals are described in “*GTM-IP Specification*”. For more information about CADC functions please refer to **Section 36, Cyclic Analog to Digital Converter (CADC)**.

31.4.5 Outputs to ENCA

GTM outputs can trigger the ENCA timer counter capture. The following GTM outputs can be used as ENCS trigger sources.

- TOM_OUT
- ATOM_OUT

Signals specified above are multiplexed together and routed to the ENCA. The GTM outputs can trigger capture trigger inputs 0 and 1. Multiplexing scheme is described in **Section 33, Peripheral Interconnection (PIC)**. TOM_OUT and ATOM_OUT signals are described in “*GTM-IP Specification*”. For more information about ENCA functionality please refer to **Section 32, Encoder Timer (ENCA)**.

31.4.6 Outputs to DFE

GTM outputs can trigger DFE

- Timer trigger compare
- Timer DFE Capture trigger
- Timer FIFO Capture trigger

The following GTM outputs can be used as DFE trigger sources.

- TOM_OUT
- ATOM_OUT

Signals specified above are multiplexed together and routed to the DFE. Multiplexing scheme is described in **Section 33, Peripheral Interconnection (PIC)**. TOM_OUT and ATOM_OUT signals are described in “*GTM-IP Specification*”. For more information about DFE functionality please refer to **Section 37, Digital Filter Engine (DFE)**.

31.4.7 Outputs to PSI5-S

GTM outputs can be routed to PSI5-S signals enable, clear, clock and sync pulse signal. The following GTM outputs can be routed to PSI5-S

- TOM_OUT
- ATOM_OUT

Signals specified above are multiplexed together and routed to the PSI5-S. Multiplexing scheme is described in **Section 33, Peripheral Interconnection (PIC)**. TOM_OUT and ATOM_OUT signals are described in “*GTM-IP Specification*”. For more information about PSI5-S functionality please refer to **Section 25, Peripheral Sensor Interface 5 S (PSI5-S)**.

31.4.8 Peripheral Inputs

Outputs of other peripherals can be connected to GTM inputs. Those outputs are routed to TIM input channels. See **Section 33, Peripheral Interconnection (PIC)** for routing schemes.

31.4.8.1 Inputs from PORT

If port pin alternative GTM function is selected, a signal from a micro pin is routed to a corresponding TIM channel. Please see **Section 2, Pin Function** for possible routing options to TIM channels. Whether a signal from a micro pin is used by TIM channel as an input also depends on TIM sub-module configuration inside GTM. For more details about TIM please refer to “*GTM-IP Specification*”.

31.4.8.2 Inputs from ADCH

The following interrupt factors of ADCH can be routed to TIM input channels

- Conversion ending of the ADCH scan group
- Analog Boundary Flag (via Virtual PORT)

31.4.8.3 Inputs from DSADC

The following interrupt factors of DSADC can be routed to TIM input channels

- completion of DSADC conversion
- Analog Boundary Flag (via Virtual PORT)

31.4.8.4 Inputs from CADC

The following interrupt factors of CADC can be routed to TIM input channels

- completion of CADC conversion
- Analog Boundary Flag (via Virtual PORT)

31.4.8.5 Input from ENCA

All ENCA interrupt factors can be routed to GTM TIM input channels:

- Encoder input (phase A, B, Z)
- Compare match or capture
- clear interrupt by ENCA input
- clock enable output
- encoder output count status

31.4.8.6 Inputs from DFE

The following DFE interrupt factors can be routed to TIM input channels:

- peak and hold update
- output data available

31.4.9 ADC Interface

Results of A/D conversion are stored in ADC_CH[y]_DATA registers (y = 0..31). Either registers 0 to 15 or registers 16 to 31 can be used based on GTM_ADCl_CHSELj.CHS. In addition, a register designation is also based on DFTAG (ADCH and CADC) or DFES (DSADC). For more information about DFTAG and DFES configuration please refer to **Section 34, Analog to Digital Converter (ADCH)**, **Section 35, Delta-Sigma Analog to Digital Converter (DSADC)** and **Section 36, Cyclic Analog to Digital Converter (CADC)**.

It is not possible to send one piece of A/D data to multiple ADC_CH[y]_DATA registers based on DFTAG. Sending multiple A/D results to a same ADC_CH[y]_DATA register based on DFTAG is possible but has to be carefully designed to avoid data overwrites.*¹

The data format in ADC_CH[y]_DATA register matches configuration of corresponding A/D channel.

Note 1. For example, choosing DFTAG=3h for conversion result of ADCH and DFES=0h and DFTAG=3h for conversion result of Delta/Sigma ADC is not allowed.

Table 31.20 Allocation of an ADC_CH[y]_DATA Register and DFTAG, DFES Setting (1/2)

GTM Registers	ADCH		DSADC		CADC	
	CHS	DFTAG[3:0]	DFES	DFTAG[3:0]	CHS	DFTAG[3:0]
ADC_CH0_DATA	0 _H	0 _H	0 _H	0 _H	0 _H	0 _H
ADC_CH1_DATA	0 _H	1 _H	0 _H	1 _H	0 _H	1 _H
ADC_CH2_DATA	0 _H	2 _H	0 _H	2 _H	0 _H	2 _H
ADC_CH3_DATA	0 _H	3 _H	0 _H	3 _H	0 _H	3 _H
ADC_CH4_DATA	0 _H	4 _H	0 _H	4 _H	0 _H	4 _H
ADC_CH5_DATA	0 _H	5 _H	0 _H	5 _H	0 _H	5 _H
ADC_CH6_DATA	0 _H	6 _H	0 _H	6 _H	0 _H	6 _H
ADC_CH7_DATA	0 _H	7 _H	0 _H	7 _H	0 _H	7 _H
ADC_CH8_DATA	0 _H	8 _H	0 _H	8 _H	0 _H	8 _H
ADC_CH9_DATA	0 _H	9 _H	0 _H	9 _H	0 _H	9 _H
ADC_CH10_DATA	0 _H	A _H	0 _H	A _H	0 _H	A _H
ADC_CH11_DATA	0 _H	B _H	0 _H	B _H	0 _H	B _H
ADC_CH12_DATA	0 _H	C _H	0 _H	C _H	0 _H	C _H
ADC_CH13_DATA	0 _H	D _H	0 _H	D _H	0 _H	D _H
ADC_CH14_DATA	0 _H	E _H	0 _H	E _H	0 _H	E _H
ADC_CH15_DATA	0 _H	F _H	0 _H	F _H	0 _H	F _H
ADC_CH16_DATA	1 _H	0 _H	1 _H	0 _H	1 _H	0 _H
ADC_CH17_DATA	1 _H	1 _H	1 _H	1 _H	1 _H	1 _H
ADC_CH18_DATA	1 _H	2 _H	1 _H	2 _H	1 _H	2 _H
ADC_CH19_DATA	1 _H	3 _H	1 _H	3 _H	1 _H	3 _H
ADC_CH20_DATA	1 _H	4 _H	1 _H	4 _H	1 _H	4 _H

Table 31.20 Allocation of an ADC_CH[y]_DATA Register and DFTAG, DFES Setting (2/2)

GTM Registers	ADCH		DSADC		CADC	
	CHS	DFTAG[3:0]	DFES	DFTAG[3:0]	CHS	DFTAG[3:0]
ADC_CH21_DATA	1 _H	5 _H	1 _H	5 _H	1 _H	5 _H
ADC_CH22_DATA	1 _H	6 _H	1 _H	6 _H	1 _H	6 _H
ADC_CH23_DATA	1 _H	7 _H	1 _H	7 _H	1 _H	7 _H
ADC_CH24_DATA	1 _H	8 _H	1 _H	8 _H	1 _H	8 _H
ADC_CH25_DATA	1 _H	9 _H	1 _H	9 _H	1 _H	9 _H
ADC_CH26_DATA	1 _H	A _H	1 _H	A _H	1 _H	A _H
ADC_CH27_DATA	1 _H	B _H	1 _H	B _H	1 _H	B _H
ADC_CH28_DATA	1 _H	C _H	1 _H	C _H	1 _H	C _H
ADC_CH29_DATA	1 _H	D _H	1 _H	D _H	1 _H	D _H
ADC_CH30_DATA	1 _H	E _H	1 _H	E _H	1 _H	E _H
ADC_CH31_DATA	1 _H	F _H	1 _H	F _H	1 _H	F _H

The figure below shows the bit correspondence between the AD conversion result and the ADC_CH[y]_DATA register of GTM.

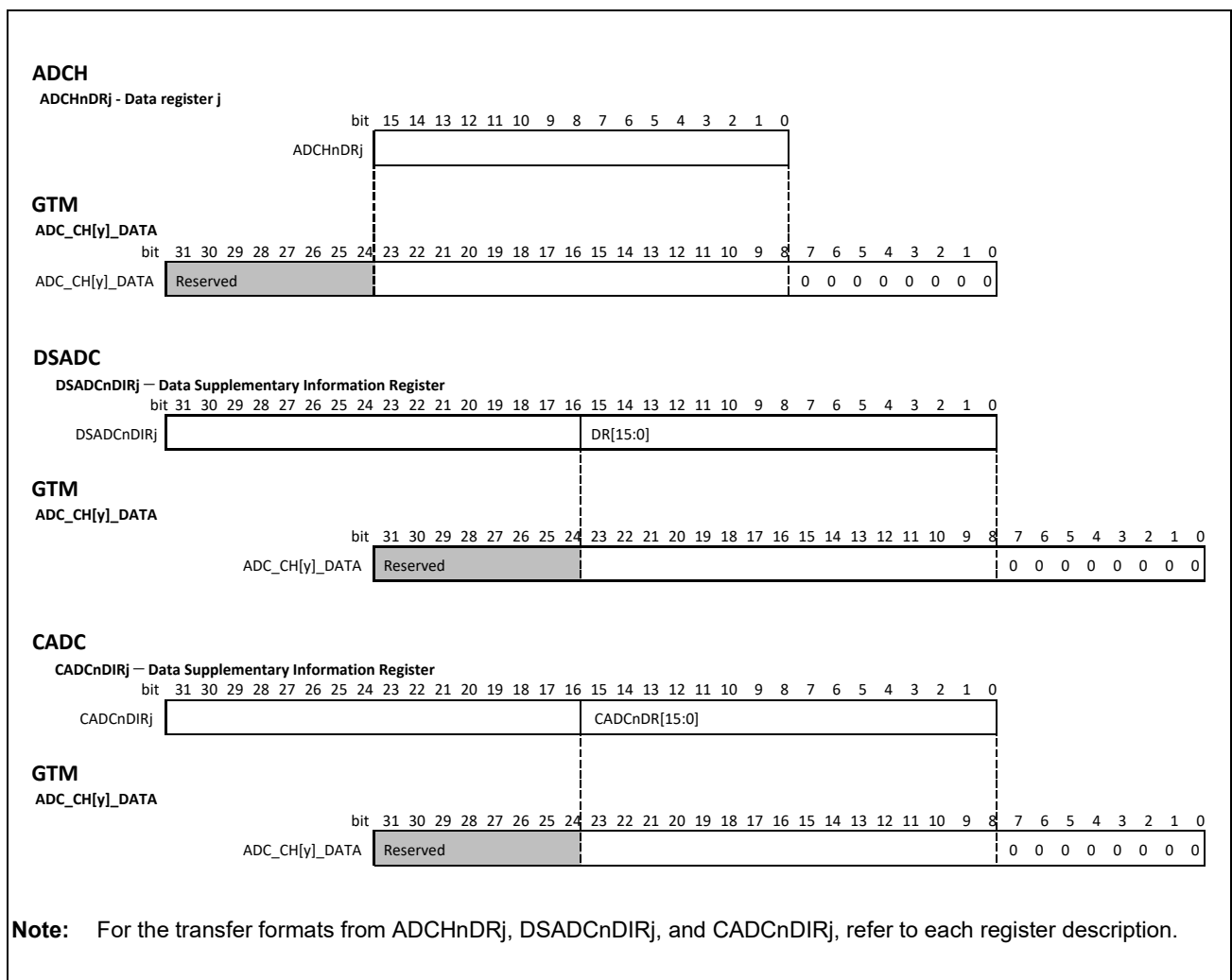


Figure 31.5 Comparison of AD conversion result with ADC_CH[y]_DATA register of GTM

CAUTION

ADC_CH[y]_STA has no information and returns the same value as ADC_CH[y]_DATA when read.

Bits 31 to 24 of ADC_CH[y]_DATA are always reads as 0xC0. ADC_NEW_DATA is not supported. For details about ADC_NEW_DATA, please refer to “GTM-IP Specification” documents.

31.4.10 Safety Mechanism

MCS RAM is protected by ECC function. ECC error of MCS RAM is also reported by MCS channel. For more details about the ECC function, please refer to **Section 38, Functional Safety**.

31.4.11 GTM Debug Function

GTM debug feature is as follows.

- Run, break
- Single step execution of MCS
- On-chip breakpoint by:
 - MCS instruction or data access
 - DPLL RAM data access
 - ARU data access
 - TBU counter
- Trace the followings with timestamp: (E2xFCC1 only)
 - MCS instruction or data access
 - DPLL RAM data access
 - ARU data access
 - TBU counter
 - TIM/TOM/ATOM state change

CAUTION

The register behaviors of AFD[i]_CH[x]_BUFFACC, TIM[i]_CH[x]_GPR0/1 and ATOM[i]_CH[x]_SR0/1 in case of debugger read access described in “GTM-IP Specification” are not supported.

Section 32 Encoder Timer (ENCA)

This section contains general descriptions of the Encoder Timer (ENCA).

The first part in this section describes all specific properties of this product, such as the number of units, and register base addresses.

The remainder of the section describes the functions and registers of the ENCA.

32.1 Features of ENCA

32.1.1 Number of Units

This microcontroller has the following number of ENCA units.

Each ENCA unit has one-channel encoder timer.

Table 32.1 Number of Units

Product Name	RH850/E2x-FCC1	
	373 pins	292 pins
Number of Units	2	2
Name	ENCAn (n = 0 to 1)	ENCAn (n = 0 to 1)

Table 32.2 Number of Units

Product Name	RH850/E2M	
	373 pins	292 pins
Number of Units	2	2
Name	ENCAn (n = 0 to 1)	ENCAn (n = 0 to 1)

Table 32.3 Index

Index	Description
n	Throughout this section, the individual ENCA units are identified by the index “n” (n = 0, 1); for example, ENCAnCTL is the ENCAn control register.

32.1.2 Register Base Addresses

The ENCA_n base addresses are listed in the following table.

The ENCA_n register addresses are given as offsets from the base addresses.

Table 32.4 Register Base Addresses

Base Address Name	Base Address	Bus Group
<ENCA0_base>	FFE8 0000 _H	Peripheral Group 2H
<ENCA1_base>	FFE8 1000 _H	Peripheral Group 2H

32.1.3 Clock Supply

The ENCA_n clock supply is shown in the following table.

Table 32.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
ENCA _n	PCLK	CLK_HSB

32.1.4 Interrupt Requests

The ENCA interrupt requests are listed in the following table.

Table 32.6 Interrupts and DMA/DTS Requests

Interrupt Symbol Name	Description	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
ENCA0				
INTENCA0IOV	ENCATIOV	206-213 (selectable)	group1 64-71 (selectable)	group2 110-117, group3 70-77 (selectable)
INTENCA0IUD	ENCATIUD	206-213 (selectable)	group1 64-71 (selectable)	group2 110-117, group3 70-77 (selectable)
INTENCA0I0	ENCATINT0	206-213 (selectable)	group1 64-71 (selectable)	group2 110-117, group3 70-77 (selectable)
INTENCA0I1	ENCATINT1	206-213 (selectable)	group1 64-71 (selectable)	group2 110-117, group3 70-77 (selectable)
INTENCA0IEC	ENCATIEC	206-213 (selectable)	group1 64-71 (selectable)	group2 110-117, group3 70-77 (selectable)
ENCA1				
INTENCA1IOV	ENCATIOV	214-221 (selectable)	group1 64-71 (selectable)	group2 110-117, group3 70-77 (selectable)
INTENCA1IUD	ENCATIUD	214-221 (selectable)	group1 64-71 (selectable)	group2 110-117, group3 70-77 (selectable)
INTENCA1I0	ENCATINT0	214-221 (selectable)	group1 64-71 (selectable)	group2 110-117, group3 70-77 (selectable)
INTENCA1I1	ENCATINT1	214-221 (selectable)	group1 64-71 (selectable)	group2 110-117, group3 70-77 (selectable)
INTENCA1IEC	ENCATIEC	214-221 (selectable)	group1 64-71 (selectable)	group2 110-117, group3 70-77 (selectable)

NOTES

1. Interrupt Number can be selected at the GTM_IRQ_SEL8j register. (j = 0 to 15)
2. DMAC Trigger Number can be selected at the GTM_DMA_SEL8j (j = 0 to 7)
3. GTM_IRQ_SEL8j (j = 0 to 15) and GTM_DMA_SEL8j (j = 0 to 7) can be set without depending on the setting of option byte (OPBT8.ATU_GTM_SEL) and module standby (MSR_GTM.MS_GTM).

32.1.5 Reset Sources

The ENCA_n reset sources are listed in the following table. The ENCA_n is initialized by these reset sources.

Table 32.7 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
ENCA _n	All registers	✓	✓	✓	✓	—	—

32.1.6 External Input/Output Signals

The external input/output signals of ENCA_n are listed below.

Table 32.8 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal Name
ENCA0		
ENCA0E0	Encoder input (phase A)*1	ENCA _n E0
ENCA0E1	Encoder input (phase B)*1	ENCA _n E1
ENCA0EC	Encoder input (phase Z)*1	ENCA _n EC
ENCA1		
ENCA1E0	Encoder input (phase A)*1	ENCA _n E0
ENCA1E1	Encoder input (phase B)*1	ENCA _n E1
ENCA1EC	Encoder input (phase Z)*1	ENCA _n EC

Note 1. This input passes PIC.

32.2 Overview

32.2.1 Functional Overview

- Counter control signal generation from the encoder input signal, and count operation in synchronization with PCLK
- Capture function for capturing the count value with an external trigger signal
- Compare function for compare match judgment with the count value
- Two capture/compare registers that can be set separately for capture operation and for compare operation
- Interrupt mask function for masking the interrupt request signal output as a result of compare match judgment during compare operation
- Function for loading the value of the capture/compare register to the counter upon underflow occurrence
- Encoder input signal can be used as the timer counter clear condition
- Edge or level can be selected for determining the presence of the encoder input signal that is used as the timer counter clear condition
- Detection of counter overflow and underflow, and output of error flags and error occurrence interrupts
- Five interrupts: two capture/compare interrupts, one counter clear interrupt, one overflow interrupt, and one underflow interrupt

32.2.2 Block Diagram

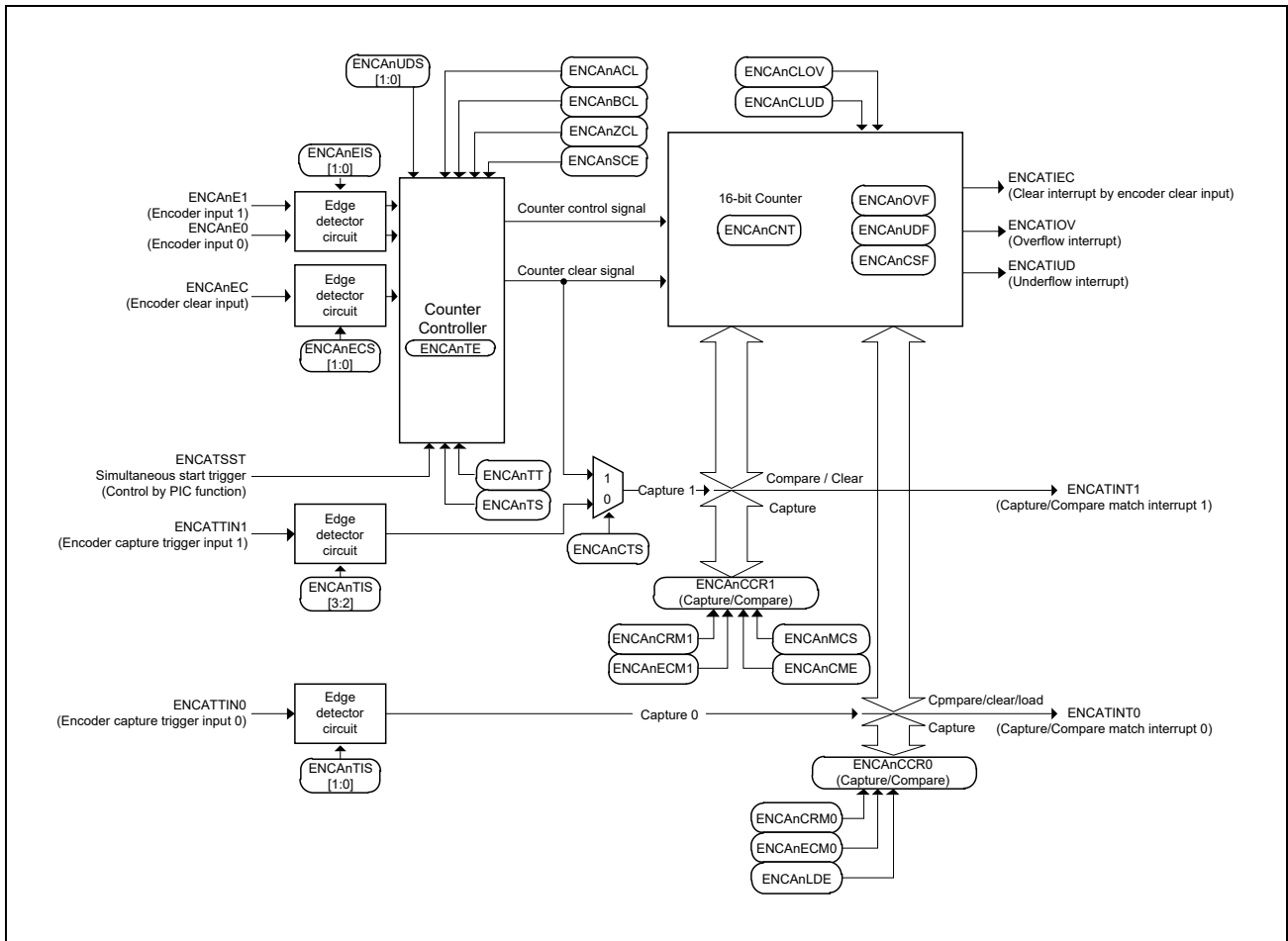


Figure 32.1 ENCA Block Diagram

32.3 Registers

32.3.1 List of Registers

The ENCA registers are listed in the following table.

<ENCA_n_base> is defined in Register Base Address.

Table 32.9 Registers

Module	Register	Symbol	Address	Access	Access Protection
ENCA _n	ENCA _n Capture/Compare Register 0	ENCA _n CCR0	<ENCA _n _base>	16	—
ENCA _n	ENCA _n Capture/Compare Register 1	ENCA _n CCR1	<ENCA _n _base> + 04 _H	16	—
ENCA _n	ENCA _n Counter Register	ENCA _n CNT	<ENCA _n _base> + 08 _H	16	—
ENCA _n	ENCA _n Status Flag Register	ENCA _n FLG	<ENCA _n _base> + 0C _H	8	—
ENCA _n	ENCA _n Status Flag Clear Register	ENCA _n FGC	<ENCA _n _base> + 10 _H	8	—
ENCA _n	ENCA _n Timer Enable Status Register	ENCA _n TE	<ENCA _n _base> + 14 _H	8	—
ENCA _n	ENCA _n Timer Start Trigger Register	ENCA _n TS	<ENCA _n _base> + 18 _H	8	—
ENCA _n	ENCA _n Timer Stop Trigger Register	ENCA _n TT	<ENCA _n _base> + 1C _H	8	—
ENCA _n	ENCA _n I/O Control Register 0	ENCA _n IOC0	<ENCA _n _base> + 20 _H	8	—
ENCA _n	ENCA _n Control Register	ENCA _n CTL	<ENCA _n _base> + 40 _H	16	—
ENCA _n	ENCA _n I/O Control Register 1	ENCA _n IOC1	<ENCA _n _base> + 44 _H	8	—
ENCA _n	ENCA _n Emulation Register	ENCA _n EMU	<ENCA _n _base> + 48 _H	8	—

32.3.2 Details of Registers

32.3.2.1 ENCA_nCTL — ENCA_n Control Register

This register is used to configure various operation settings of the Encoder Timer.

Writing to this register during operation is prohibited.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENCA _n CME	ENCA _n MCS	—	—	—	—	ENCA _n CRM1	ENCA _n CRM0	ENCA _n CTS	—	—	ENCA _n LDE	ENCA _n ECM1	ENCA _n ECM0	ENCA _n UDS [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Table 32.10 ENCA_nCTL Register Contents (1/2)

Bit Position	Bit Name	Function
15	ENCA _n CME	Encoder Clear Mask Enable This bit is used to enable/disable masking of compare-match interrupt detection when the compare function is used. 0: Disables the compare-match interrupt (ENCA _n TINT1) mask function for the ENCA _n CCR1 register 1: Enables the compare-match interrupt (ENCA _n TINT1) mask function for the ENCA _n CCR1 register. This bit is valid only when ENCA _n CRM1 = 0. When this bit is set to "1", setting ENCA _n ECM1 to "1" is prohibited.
14	ENCA _n MCS	Encoder Mask Clear Select This bit is used to select the trigger for cancelling masking of compare-match interrupt detection ENCA _n TINT1 when the compare function is used. This bit is valid only when ENCA _n CRM1 = 0. 0: Masking of compare-match interrupt detection is canceled when the ENCA _n CCR1 register is written. 1: Masking of compare match interrupt detection is canceled when one of the following three operations is performed. Timer counter clear operation accompanying encoder clear input. Timer counter clear operation upon compare-match between ENCA _n CNT and ENCA _n CCR0 when ENCA _n ECM0 = 1. Loading from ENCA _n CCR0 to the timer counter upon underflow detection when ENCA _n LDE = 1.
13 to 10	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
9	ENCA _n CRM1	ENCA _n CCR1 Register Mode 0: ENCA _n CCR1 used as compare register. 1: ENCA _n CCR1 used as capture register.
8	ENCA _n CRM0	ENCA _n CCR0 Register Mode 0: ENCA _n CCR0 used as compare register. 1: ENCA _n CCR0 used as capture register.

Table 32.10 ENCACTL Register Contents (2/2)

Bit Position	Bit Name	Function
7	ENCACTS	<p>ENCACCR1 Capture Trigger Select</p> <p>This is a trigger selection bit for the capture operation to the ENCACCR1 register.</p> <p>This bit is valid only when ENACRM1 = 1.</p> <p>0: Uses ENCATTIN1 of capture trigger 1 signal as the capture trigger for the ENCACCR1 register.</p> <p>1: Uses the counter clear signal selected with ENASCE as the capture trigger for the ENCACCR1 register.</p>
6, 5	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
4	ENCALDE	<p>ENCA Counter Load Enable</p> <p>This bit is used to enable/disable setting value loading to the counter upon underflow occurrence.</p> <p>This bit is valid only when ENACRM0 = 0.</p> <p>When ENACRM0 = 1, ENACCR0 register setting value is not loaded to the counter upon occurrence of an underflow, regardless of the value of this bit.</p> <p>0: Disables loading of ENACCR0 register setting value to counter upon occurrence of a counter underflow.</p> <p>1: Enables loading of ENACCR0 register setting value to counter upon occurrence of a counter underflow.</p>
3	ENACEM1	<p>Encoder Clear Mode 1</p> <p>This bit is used to set the counter clear operation upon match between the count value and ENACCR1 setting value.</p> <p>This bit is valid only when ENACRM1 = 0.</p> <p>0: Does not clear the counter to 0000_H upon match of timer count value and ENACCR1 setting value.</p> <p>1: Clears the counter to 0000_H upon match of timer count value and ENACCR1 setting value if the next count is a down-count.</p>
2	ENACEM0	<p>Encoder Clear Mode 0</p> <p>This bit is used to set the counter clear operation upon match between the count value and ENACCR0 setting value.</p> <p>This bit is valid only when ENACRM0 = 0.</p> <p>0: Does not clear the counter to 0000_H upon match of timer count value and ENACCR0 setting value.</p> <p>1: Clears the counter to 0000_H upon match of timer count value and ENACCR0 setting value if the next count is an up-count.</p>
1, 0	ENAU[1:0]	<p>Up/down Count Selection 1 and 0</p> <p>These are the counter up/down control bits using ENAE0 and ENAE1.</p> <p>00: Upon detection of valid edge of ENAE0, - down-count when ENAE1 = H - up-count when ENAE1 = L</p> <p>01: Upon detection of valid edge of ENAE0, up-count, Upon detection of valid edge of ENAE1, down-count</p> <p>10: At rising edge of ENAE0, down-count At falling edge of ENAE0, up-count However, count operation is performed only when ENAE1 = L.</p> <p>11: Detection of both edges of ENAE0, ENAE1. The count operation is determined based on the combination of the detected edge and level.</p>

32.3.2.2 ENCA_nIOC0 — ENCA_n I/O Control Register 0

This register is used to select the input edge of capture triggers 0 and 1 (ENCATTIN0, ENCATTIN1).

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	ENCA _n TIS[3:2]		ENCA _n TIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 32.11 ENCA_nIOC0 Register Contents

Bit Position	Bit Name	Function
7 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3, 2	ENCA _n TIS[3:2]	Input Edge Selection for Capture Trigger 1 These bits are valid only when ENCA _n CTL.ENCA _n CRM1 = 1 and ENCA _n CTL.ENCA _n CTS = 0. All other settings of ENCA _n CRM1 and ENCA _n CTS are invalid. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection
1, 0	ENCA _n TIS[1:0]	Input Edge Selection for Capture Trigger 0 These bits are valid only when ENCA _n CTL.ENCA _n CRM0 = 1. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection

32.3.2.3 ENCAIOC1 — ENCA I/O Control Register 1

This register is used to perform the clear condition setting and edge selection for input from the encoder.

Writing to this register during operation is prohibited.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	ENCA _n SCE	ENCA _n ZCL	ENCA _n BCL	ENCA _n ACL	ENCA _n ECS[1:0]		ENCA _n EIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.12 ENCAIOC1 Register Contents (1/2)

Bit Position	Bit Name	Function
7	ENCA _n SCE	Encoder Special-Clear Enable This is an encoder special clear enable bit. When setting this bit to 1, set ENCA _n UDS1 and ENCA _n UDS0 to 10 _B or 11 _B . The operation is not guaranteed if this bit is set to 1 with ENCA _n UDS1 and ENCA _n UDS0 set to 00 _B or 01 _B . 0: Clears the counter upon detection of ENCA _n EC valid edge (set with ENCA _n ECS1 and ENCA _n ECS0). 1: Clears the counter upon detection of input level condition of ENCA _n E0, ENCA _n E1 and ENCA _n EC (set with ENCA _n ZCL bit, ENCA _n BCL bit, and ENCA _n ACL bit).
6	ENCA _n ZCL	Input-Z Clear Condition Selection This bit is used to set the condition for clearing the encoder clear input (ENCA _n EC) when the encoder special clear function is used. This bit is valid only when ENCA _n SCE = 1; and is invalid when ENCA _n SCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
5	ENCA _n BCL	Input-B Clear Condition Selection This bit is used to set the condition for clearing the encoder input 1 (ENCA _n E1) when the encoder special clear function is used. This bit is valid only when ENCA _n SCE = 1; and is invalid when ENCA _n SCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
4	ENCA _n ACL	Input-A Clear Condition Selection This bit is used to set the condition for clearing the encoder input 0 (ENCA _n E0) when the encoder special clear function is used. This bit is valid only when ENCA _n SCE = 1; and is invalid when ENCA _n SCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
3, 2	ENCA _n ECS[1:0]	Encoder Clear Input Edge Selection 1 and 0 These are the encoder clear input edge selection bits. These bits are valid only when ENCA _n SCE = 0; and are invalid when ENCA _n SCE = 1. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection

Table 32.12 ENCA_nIOC1 Register Contents (2/2)

Bit Position	Bit Name	Function
1, 0	ENCA _n EIS[1:0]	Encoder Edge Input Selection 1 and 0 These are the encoder input edge selection bits. These bits are valid when ENCA _n UDS1 and ENCA _n UDS0 = 00 _B or 01 _B , and are invalid when ENCA _n UDS1 and ENCA _n UDS0 = 10 _B or 11 _B . 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection

32.3.2.4 ENCA_nFLG — ENCA_n Status Flag Register

This register holds the status flags of the timer counter of ENCA_n.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	ENCA _n CSF	ENCA _n UDF	ENCA _n OVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 32.13 ENCA_nFLG Register Contents

Bit Position	Bit Name	Function
7 to 3	—	Reserved When read, the value after reset is returned.
2	ENCA _n CSF	Counter Status Flag This bit reflects the current timer counter operation. 0: Timer counter in up-count status 1: Timer counter in down-count status
1	ENCA _n UDF	Underflow Flag This bit reflects the occurrence of an underflow during the timer counter operation. This bit is cleared at the start of count operation. 0: This flag is cleared to 0 upon any of the following events: - "1" is written to ENCA _n FGC.ENCA _n CLUD. - The flag is cleared to 0 by setting ENCA _n TS bit to "1" when ENCA _n TE = 0. 1: This flag is set to "1" upon occurrence of an underflow during the encoder timer count operation.
0	ENCA _n OVF	Overflow Flag This bit reflects the occurrence of an overflow during the timer counter operation. This bit is cleared at the start of count operation. 0: This flag is cleared to 0 upon any of the following events: - "1" is written to ENCA _n FGC.ENCA _n CLOV. - The flag is cleared to 0 by setting ENCA _n TS bit to "1" when ENCA _n TE = 0. 1: This flag is set to "1" upon occurrence of an overflow during the encoder timer count operation.

32.3.2.5 ENCA_nFGC — ENCA_n Status Flag Clear Register

This register is used to clear the timer counter status flags of ENCA_nFLG.

This register always returns 0 when read.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ENCA _n CLUD	ENCA _n CLOV
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W

Table 32.14 ENCA_nFGC Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When writing, write the value after reset.
1	ENCA _n CLUD	Underflow Flag Clear This bit clears the underflow flag. 0: Writing is ignored. 1: Clears ENCA _n UDF of the ENCA _n FLG register (clears underflow detection).
0	ENCA _n CLOV	Overflow Flag Clear This bit clears the overflow flag. 0: Writing is ignored. 1: Clears ENCA _n OVF of the ENCA _n FLG register (clears overflow detection).

32.3.2.6 ENCA_nCCR0 — ENCA_n Capture/Compare Register 0

This register is a 16-bit capture/compare register 0.

When this register functions as a capture register, only reading is possible, and write operation is ignored.

When this register functions as a compare register, reading and writing is possible.

Value after reset: 0000_H

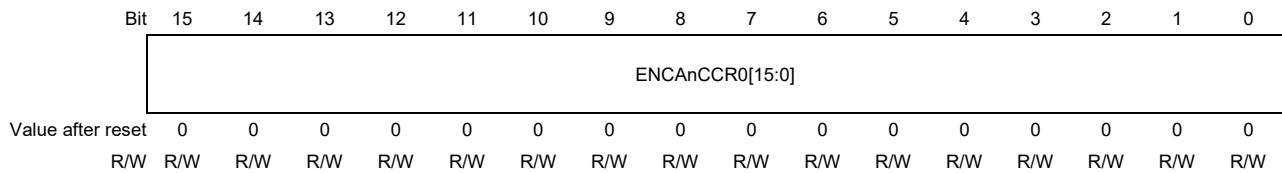


Table 32.15 ENCA_nCCR0 Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCA _n CCR0 [15:0]	Capture/Compare Register 0 Upon occurrence of an underflow, the setting value of this register may be loaded to the counter according to the ENCA _n CTL.ENCA _n LDE setting. See the description of the ENCA _n LDE bit in ENCA control register ENCA _n CTL for details. <ul style="list-style-type: none"> • If ENCA_nCTL.ENCA_nCRM0 = 0: ENCA_nCCR0 is a compare register. Set the value to be compared with the timer count value. • If ENCA_nCTL.ENCA_nCRM0 = 1: ENCA_nCCR0 is a capture register. The captured timer count value is stored.

32.3.2.7 ENCA_nCCR1 — ENCA_n Capture/Compare Register 1

This register is a 16-bit capture/compare register 1.

When this register functions as a capture register, only reading is possible, and write operation is ignored.

When this register functions as a compare register, reading and writing is possible.

Value after reset: 0000_H

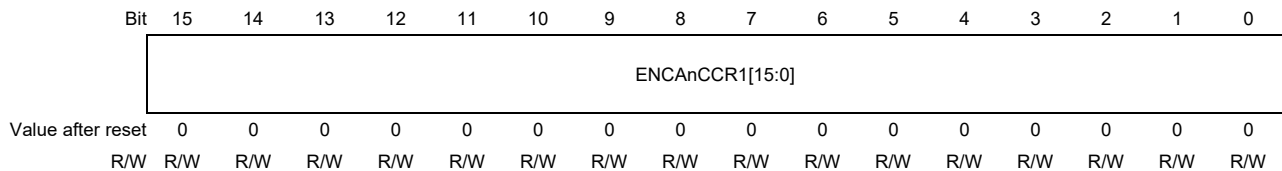


Table 32.16 ENCA_nCCR1 Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCA _n CCR1 [15:0]	Capture/Compare Register 1 During capture operation, the capture trigger to this register differs according to the ENCA _n CTL.ENCA _n CTS setting. See the description of the ENCA _n CTS bit in ENCA control register ENCA _n CTL for details if ENCA _n CTL.ENCA _n CRM1 = 0: ENCA _n CCR1 is a compare register. Set the value to be compared with the timer count value. If ENCA _n CTL.ENCA _n CRM1 = 1: ENCA _n CCR1 is a capture register. The captured timer count value is stored.

32.3.2.8 ENCA_nCNT — ENCA_n Counter Register

This register is the 16-bit timer counter register.

This register can be written only when the operation is stopped.

Value after reset: 0000_H

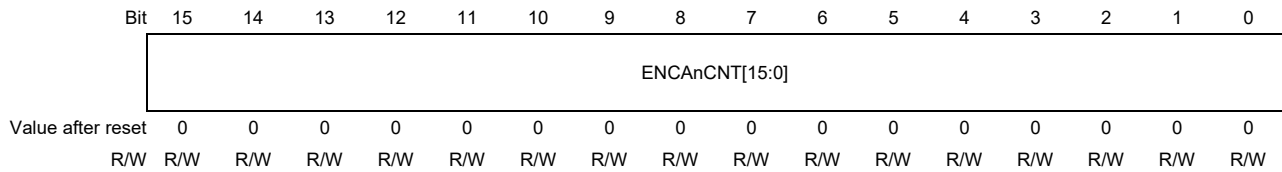


Table 32.17 ENCA_nCNT Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCA _n CNT [15:0]	Counter Register <ul style="list-style-type: none"> ENCA_nTE.ENCA_nTE status: 0 (initial setting): Count stop An arbitrary value can be set to timer counter. ENCA_nTE.ENCA_nTE status: 0 → 1 (operation start): Count operation start Up/down count operation is started with the set arbitrary value. ENCA_nTE.ENCA_nTE status: 1 (operating): Counting Up/down count operation is performed. ENCA_nTE.ENCA_nTE status: 1 → 0 (stopped): Count stop The count value immediately before the operation was stopped is held, and the count operation is stopped.

32.3.2.9 ENCA_nTE — ENCA_n Timer Enable Status Register

This register indicates the operating status of ENCA_n.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ENCA _n TE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 32.18 ENCA_nTE Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned.
0	ENCA _n TE	Timer Status Enable This is a status bit that indicates the operation enabled/stopped status of ENCA _n . This bit is cleared to 0 when “1” is written to ENCA _n TT.ENCA _n TT. This bit is set to “1” when “1” is written to ENCA _n TS.ENCA _n TS. 0: Operation stopped 1: Operation enabled

32.3.2.10 ENCA_nTS — ENCA_n Timer Start Trigger Register

This register provides the trigger bit for setting the ENCA_n to the operation enabled state.

This register is always read as 00_H. It can be written only when ENCA_nTE.ENCA_nTE is 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ENCA _n TS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 32.19 ENCA_nTS Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When writing, write the value after reset.
0	ENCA _n TS	Timer Start Trigger This is the trigger bit that sets the ENCA _n to the operation enabled state. 0: Writing is ignored. 1: The ENCA _n is set to the operation enabled state by setting ENCA _n TE.ENCA _n TE = 1.

32.3.2.11 ENCA_nTT — ENCA_n Timer Stop Trigger Register

This register provides the trigger bit for setting the ENCA_n to the operation stopped state.

This register is always read as 00_H.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ENCA _n TT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 32.20 ENCA_nTT Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When writing, write the value after reset.
0	ENCA _n TT	Timer Stop Trigger This is the trigger bit that sets the ENCA _n to the operation stopped state. 0: Writing is ignored. 1: Clears ENCA _n TE. ENCA _n TE to "0", to set the ENCA _n to the count operation stopped state.

32.3.2.12 ENCA_nEMU — ENCA_n Emulation Register

This register controls operations by SVSTOP.

Writing to this register should be performed in the counter operation stopped status (ENCA_nTE.ENCA_nTE = 0 and EPC.SVSTOP = 0).

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	ENCA _n SVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 32.21 ENCA_nEMU Register Contents

Bit Position	Bit Name	Function
7	ENCA _n SVSDIS	When EPC.SVSTOP bit = 0: The count clock continues when the debugger assumes control of the microcontroller (at a break point, etc.), regardless of the value of this bit (1 or 0). When EPC.SVSTOP bit = 1: 0: The count clock stops when the debugger assumes control of the microcontroller (at a break point, etc.). 1: The count clock continues when the debugger assumes control of the microcontroller (at a break point, etc.).
6 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

32.4 Operation

The ENCA_n operates the timer counter with counter up/down control and clear control by encoder inputs. The ENCA_nCCR0 and ENCA_nCCR1 registers can be used as dedicated compare registers or as dedicated capture registers.

32.4.1 Timer Counter Operation

The timer counter operations of the ENCA_n are described below.

The figure below shows the operation phases. See the corresponding section with the section number for detailed descriptions on each operation.

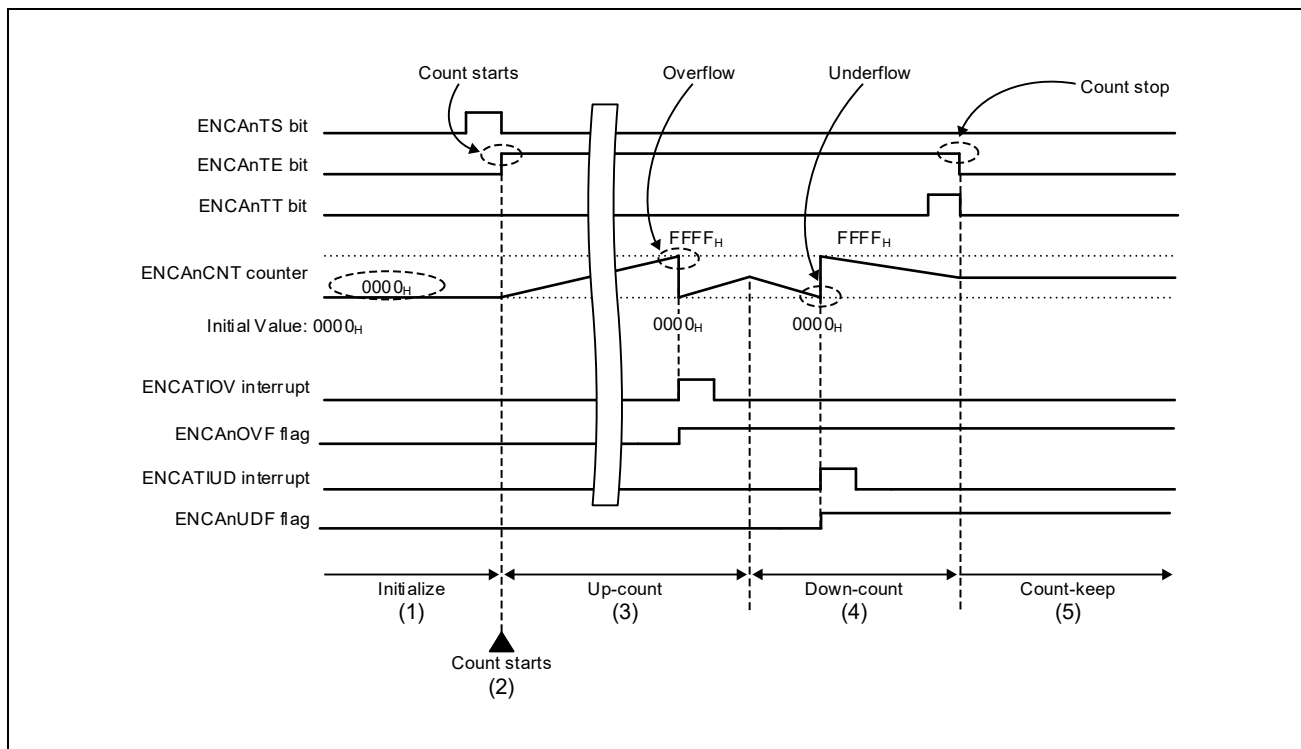


Figure 32.2 Timer Counter Initial Value Setting/Start/Stop

(1) Timer Counter Initial Value Setting

The initial value of the ENCA_n counter register (ENCA_nCNT) can be set to the counter operation stopped state (ENCA_nTE = 0).

(2) Timer Counter Startup

By writing “1” to the timer start trigger bit (ENCA_nTS), the timer status enable bit (ENCA_nTE) is set to “1”, the count operation is enabled, and counting is performed upon detection of the valid edge of the encoder input.

(3) Overflow Operation

An overflow occurs when up-counting is performed when the count value is FFFF_H. If the count value changes from FFFF_H to 0000_H, an overflow interrupt (ENCA_nTIOV) is generated, and the overflow flag (ENCA_nOVF) is set to “1”. The overflow flag (ENCA_nOVF) is cleared to “0” when “1” is set to the overflow flag clear bit (ENCA_nCLOV). For details about the operation, see “**Section 32.6.1, Overflow Occurrence and Overflow Flag Clear Operation**”

(4) Underflow Operation

An underflow occurs when down-counting is performed when the count value is 0000_H. If the count value changes

from 0000_H to FFFF_H, an underflow interrupt (ENCATIUD) is generated, and the underflow flag (ENCA_nUDF) is set to “1”. The underflow flag (ENCA_nUDF) is cleared to “0” when “1” is set to the underflow flag clear bit (ENCA_nCLUD). For details about the operation, see “**Section 32.6.2, Underflow Occurrence and Underflow Flag Clear Operation**”.

(5) Timer Counter Stop

By writing “1” to the timer stop trigger bit (ENCA_nTT), the timer status enable bit (ENCA_nTE) is cleared to “0”, and the count operation is stopped. At this time, the timer counter is not reset to 0000_H and holds the value before count operation stops.

32.4.2 Up/Down Control of Timer Counter

Up/down control is performed by judging the phase of the encoder inputs (ENCA_nE0, ENCA_nE1) according to the settings of ENCA_nUDS1 and ENCA_nUDS0.

32.4.2.1 When the ENCAAnUDS1/ENCAAnUDS0 Bits in the ENCAAnCTL Register = 00_B

Table 32.22 When ENCAAnUDS1/ENCAAnUDS0 Bits = 00_B

ENCAAnUDS1	ENCAAnUDS0	Operation Description		
		ENCAAnE0 Pin	ENCAAnE1 Pin	Count Operation
0	0	Rising edge	High level	Down
		Falling edge		
		Rising and falling edges		
		Rising edge	Low level	Up
		Falling edge		
		Rising and falling edges		

The valid edge for ENCAAnE0 is specified by setting ENCAAnEIS1 and ENCAAnEIS0.

Up/down count operation is performed when the valid edges and levels of ENCAAnE0 and ENCAAnE1 match.

The following timing chart shows the count operation when ENCAAnUDS1 and ENCAAnUDS0 bits = 00_B.

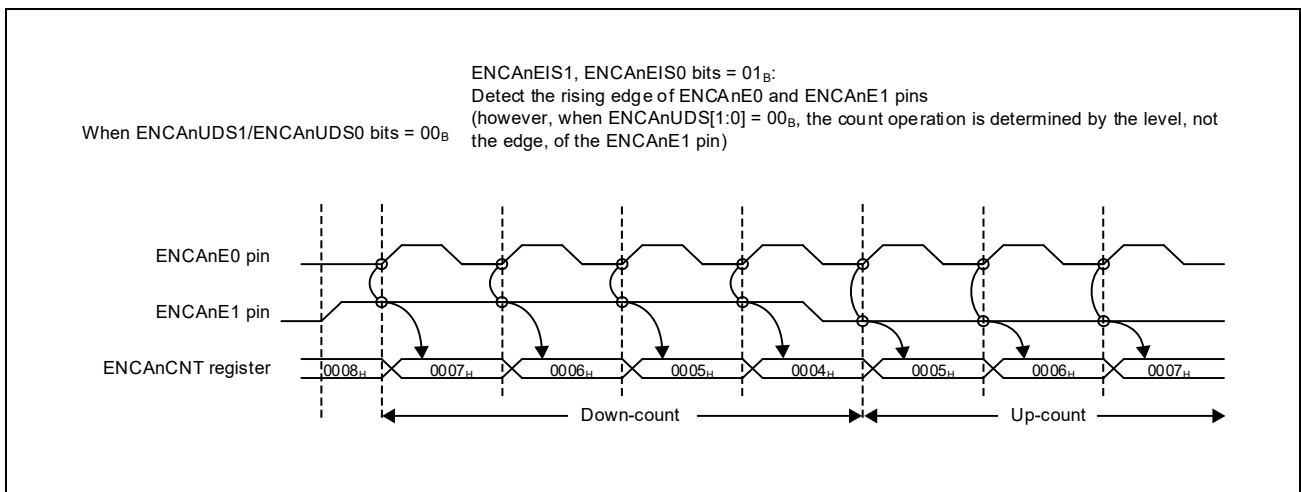


Figure 32.3 Count Operation when the ENCAAnUDS1/ENCAAnUDS0 Bits in the ENCAAnCTL Register = 00_B

32.4.2.2 When the ENCAAnUDS1/ENCAAnUDS0 Bits in the ENCAAnCTL Register = 01_B

Table 32.23 When the ENCAAnUDS1/ENCAAnUDS0 Bits = 01_B

ENCAAnUDS1	ENCAAnUDS0	Operation Description			
		ENCAAnE0 Pin	ENCAAnE1 Pin	Count Operation	
0	1	Low level	Rising edge	Down	
			Falling edge		
			Rising and falling edges		
		High level	Rising edge		
			Falling edge		
			Rising and falling edges		
		Rising edge	Low level	Up	
		Falling edge			
		Rising and falling edges			
		Rising edge	High level		Up
		Falling edge			
		Rising and falling edges			
Simultaneous input			Hold		

The valid edges for ENCAAnE0 and ENCAAnE1 are specified by setting ENCAAnEIS1 and ENCAAnEIS0.

Up/down count operation is performed when the valid edges and levels of the ENCAAnE0/ENCAAnE1 pins match, and the count is held when the valid edges overlap.

The following timing chart shows the count operation when ENCAAnUDS1 and ENCAAnUDS0 bits = 01_B.

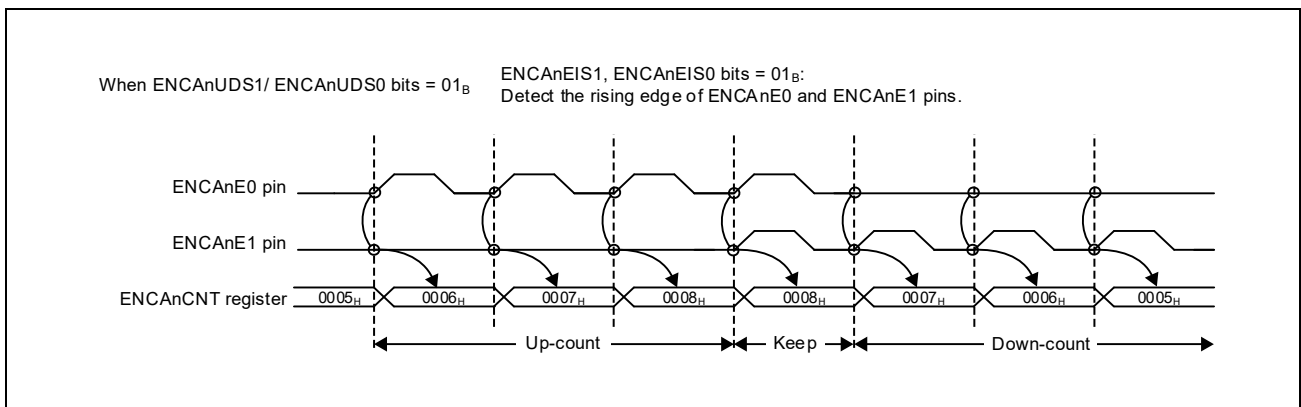


Figure 32.4 Count Operation when the ENCAAnUDS1/ENCAAnUDS0 Bits in the ENCAAnCTL Register = 01_B

32.4.2.3 When the ENCAAnUDS1 and ENCAAnUDS0 Bits in the ENCAAnCTL Register = 10_B

Table 32.24 When the ENCAAnUDS1, ENCAAnUDS0 Bits = 10_B

ENCAAnUDS1	ENCAAnUDS0	Operation Description		
		ENCAAnE0 Pin	ENCAAnE1 Pin	Count Operation
1	0	Rising edge	Low level	Down
		Rising edge	Falling edge	
		Falling edge	Low level	Up
		Falling edge	Falling edge	
		Low level	Rising edge	Hold
		Rising edge	Rising edge	
		High level	Rising edge	
		Falling edge	Rising edge	
		Low level	Falling edge	
		Rising edge	High level	
		High level	Falling edge	
		Falling edge	High level	

Specification of valid edges for ENCAAnE0 and ENCAAnE1 (settings of ENCAAnEIS1 and ENCAAnEIS0) is invalid.

The following timing chart shows the count operation when the ENCAAnUDS1/ENCAAnUDS0 bits = 10_B.

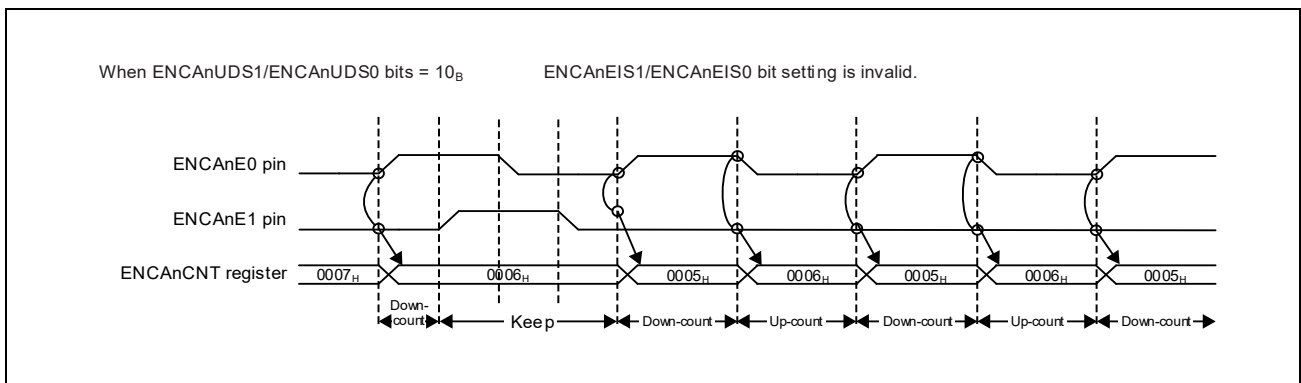


Figure 32.5 Count Operation when ENCAAnUDS1/ENCAAnUDS0 Bits in ENCAAnCTL Register = 10_B

32.4.2.4 When ENCA_nUDS1/ENCA_nUDS0 Bits in the ENCA_nCTL Register = 11_B

Table 32.25 When ENCA_nUDS1/ENCA_nUDS0 Bits = 11_B

ENCA _n UDS1	ENCA _n UDS0	Operation Description		
		ENCA _n E0 Pin	ENCA _n E1 Pin	Count Operation
1	1	Low level	Falling edge	Down
		Rising edge	Low level	
		High level	Rising edge	
		Falling edge	High level	
		Rising edge	High level	Up
		High level	Falling edge	
		Falling edge	Low level	
		Low level	Rising edge	
		Simultaneous input		

Specification of valid edges for ENCA_nE0 and ENCA_nE1 (settings of ENCA_nEIS1 and ENCA_nEIS0) is invalid.

The count value is held when the valid edges of ENCA_nE0 and ENCA_nE1 overlap.

The following timing chart shows the count operation when ENCA_nUDS1/ENCA_nUDS0 Bits = 11_B.

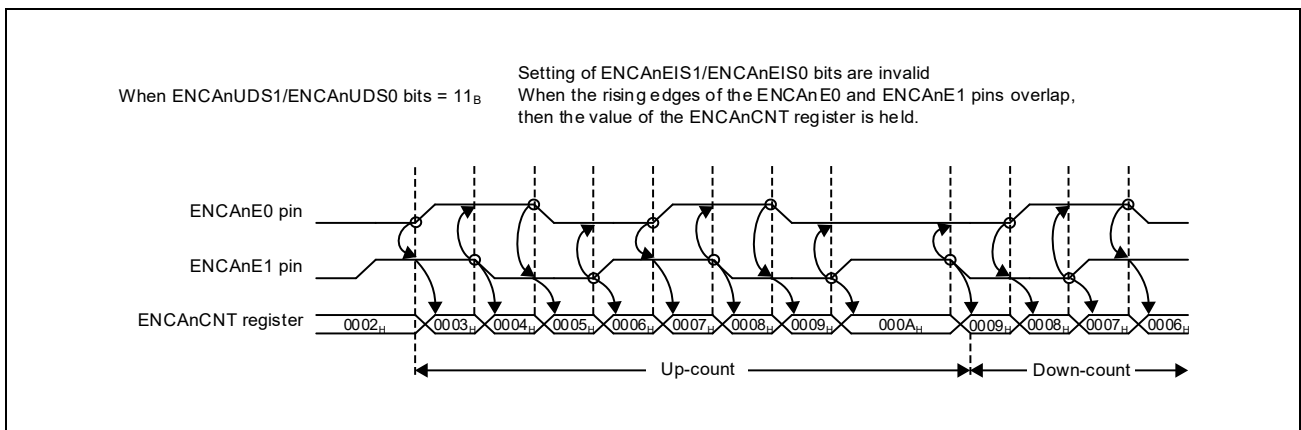


Figure 32.6 Count Operation when ENCA_nUDS1 and ENCA_nUDS0 Bits in the ENCA_nCTL Register = 11_B

32.4.3 Timer Counter Clear Control by Encoder Input

The timer counter is cleared to 0000_H by encoder clear input (ENCAnEC).

Two types of clearing methods can be selected by controlling the ENCA_nSCE, ENCA_nZCL, ENCA_nBCL, ENCA_nACL, ENCA_nECS1, and ENCA_nECS0 bits of the ENCA_nIOC1 register.

Table 32.26 Timer Counter Clear Control by Encoder Input

Clearing Method	ENCA _n SCE	ENCA _n ZCL	ENCA _n BCL	ENCA _n ACL	ENCA _n ECS1, ENCA _n ECS0
(1)	0	Invalid	Invalid	Invalid	Valid
(2)	1	Valid	Valid	Valid	Invalid

32.4.3.1 Clearing Method when ENCA_nSCE = 0

- Upon detection of the valid edge of ENCA_nEC, the timer counter is cleared to 0000_H in synchronization with the operation clock.
- The valid edge of ENCA_nEC is specified by the setting of the ENCA_nECS1 and ENCA_nECS0 bits.
- The settings of the ENCA_nZCL, ENCA_nBCL, and ENCA_nACL bits are invalid.
- An encoder clear interrupt request signal (ENCATIEC) is output simultaneously with timer counter clearing.

For details about clear operation when ENCA_nSCE = 0, see the timing chart in **32.6.19 Capture Operation Performed upon Clearing by ENCA_nEC when ENCA_nSCE = 0**.

32.4.3.2 Clearing Method when ENCA_nSCE = 1

- When the clear levels of the ENCA_nEC, ENCA_nE1, ENCA_nE0 inputs are detected, the timer counter is cleared to 0000_H in synchronization with the operating clock.
- Specify the clear levels of the ENCA_nEC, ENCA_nE1, ENCA_nE0 inputs by the settings of the ENCA_nZCL, ENCA_nBCL, and ENCA_nACL bits.
- The settings of the ENCA_nECS1 and ENCA_nECS0 bits are invalid.
- An encoder clear interrupt request signal (ENCATIEC) is output simultaneously with timer counter clearing.

The clearing conditions of the timer counter according to the ENCA_nZCL, ENCA_nBCL, and ENCA_nACL settings are listed in the table below.

Table 32.27 Clearing Conditions of the Timer Counter

Counter Clear Condition Setting			Encoder Pin Input Level		
ENCA _n ZCL	ENCA _n BCL	ENCA _n ACL	ENCAnEC	ENCAnE1	ENCAnE0
0	0	0	Low	Low	Low
0	0	1	Low	Low	High
0	1	0	Low	High	Low
0	1	1	Low	High	High
1	0	0	High	Low	Low
1	0	1	High	Low	High
1	1	0	High	High	Low
1	1	1	High	High	High

32.4.4 Functions of ENCA_nCCR0

32.4.4.1 Compare Function

- When ENCA_nCRM0 = 0, the ENCA_nCCR0 register functions as a dedicated compare register.
- Upon compare match between the values of the timer counter and the ENCA_nCCR0 setting value, a compare 0 match interrupt (ENCATINT0) is output.
- When ENCA_nECM0 = 1, the timer counter is cleared to 0000_H in synchronization with the operating clock upon compare match if the next count operation is up-count.

Table 32.28 Compare Function of ENCA_nCCR0

ENCA _n CCR0 Function	Compare Match Clear Control	Next Count Operation	Timer Counter Clearing Upon Compare Match with ENCA _n CCR0
ENCA _n CRM0	ENCA _n ECM0		
0 (Compare)	0	Up-count	Does not clear (continues count operation).
		Down-count	
	1	Up-count	Clears timer counter to 0000 _H .
		Down-count	Does not clear (continues count operation).

- When ENCA_nLDE = 1
 - Upon occurrence of an underflow, the setting value of the ENCA_nCCR0 register is loaded to the timer counter.
 - An underflow interrupt (ENCATIUD) is output.

NOTE

For the timing chart when ENCA_nLDE = 1, see “**Section 32.6.8, Using the ENCA_nLDE Function Immediately after Startup**” to “**Section 32.6.12, Up-count after Conflict between ENCA_nLDE Function (Loading Count Value) and Encoder Clearing**”.

32.4.4.2 Capture Function

- When ENCA_nCRM0 = 1, the ENCA_nCCR0 register functions as a dedicated capture register.
- Upon valid edge detection of the capture trigger input 0 (ENCATTIN0), the value of the timer counter is stored into ENCA_nCCR0.
- A capture 0 interrupt (ENCATINT0) is output during capture operation.

NOTE

For details about capture operation for ENCA_nCCR0, see the timing charts in “**Section 32.6.15, Encoder Operation when Compare Match Clear Control is Enabled and ENCA_nCTS = 0**” and “**Section 32.6.17, Encoder Operation when Compare Match Clear Control is Disabled**”.

32.4.5 Functions of ENCA_nCCR1

32.4.5.1 Compare Function

- When ENCA_nCRM1 = 0, the ENCA_nCCR1 register functions as a dedicated compare register.
- Upon compare match between the value of the timer counter and the ENCA_nCCR1 setting value, a compare 1 match interrupt (ENCA_nTINT1) is output.
- When ENCA_nECM1 = 1, the timer counter is cleared to 0000_H in synchronization with the operating clock upon compare match if the next count operation is down-count.

Table 32.29 Compare Function of ENCA_nCCR1

ENCA _n CCR1 Function	Compare Match Clear Control	Next Count Operation	Timer Counter Clearing Upon Compare Match with ENCA _n CCR1
ENCA _n CRM1	ENCA _n ECM1		
0 (Compare)	0	Up-count	Does not clear (continues count operation).
		Down-count	
	1	Up-count	Does not clear (continues count operation).
		Down-count	Clears timer counter to 0000 _H .

Compare match interrupt mask function

- When ENCA_nCME = 1, the compare 1 match interrupt mask function is enabled. In this state, the compare 1 match interrupt is output upon the first match of the value of the timer counter and the ENCA_nCCR1 setting value, and interrupts are then masked for the second and subsequent compare matches.
- When ENCA_nCME = 1 and ENCA_nMCS = 0, a compare 1 match interrupt is output once upon the first compare match by writing to the ENCA_nCCR1 register (interrupts are masked for the second and subsequent matches until the cancel trigger occurs again).
- When ENCA_nCME = 1 and ENCA_nMCS = 1, a compare 1 match interrupt is output once upon the first compare match by a timer counter clear operation accompanying encoder clear input or by a timer counter clear operation upon match between the ENCA_nCCR0 register value and the timer count value (interrupts are masked for the second and subsequent matches until the cancel trigger occurs again).
- When ENCA_nCME = 1, ENCA_nMCS = 1 and ENCA_nLDE = 1, a compare 1 match interrupt is output once upon the first compare match by a loading operation of the ENCA_nCCR0 register to the timer counter upon underflow detection (interrupts are masked for the second and subsequent matches until the cancel trigger occurs again).
- Setting ENCA_nECM1 to “1” is prohibited when enabling the compare 1 match interrupt mask function.

Table 32.30 Compare Match Interrupt Mask Function of ENCA_nCCR1

ENCA _n CCR1 Function	Compare 1 Match Interrupt Mask	Interrupt Mask Cancel Trigger	Compare 1 Match Interrupt Output upon Compare Match with ENCA _n CCR1
ENCA _n CRM1	ENCA _n CME	ENCA _n MCS	
0 (Compare)	0 (Mask function disabled)	— (Setting invalid)	Outputs compare 1 match interrupt upon each compare match.
		0 (Write operation to ENCA _n CCR1)	
	1 (Mask function enabled)	1 (Timer counter clear operation) (Loading from ENCA _n CCR0 to the timer counter upon underflow occurrence when ENCA _n LDE = 1)	Outputs compare 1 match interrupt once upon the first compare match. (Interrupts are masked for the second and subsequent matches until the cancel trigger occurs.)

32.4.5.2 Capture Function

When ENCA_nCRM1 = 1, the ENCA_nCCR1 register functions as a dedicated capture register.

NOTE

For details about capture operation of ENCA_nCCR1, see the timing chart in “**Section 32.6.13, Capture Operation between Count Clocks (ENCA_nCCR1)**”.

The operations for each of the ENCA_nCTS settings are shown in the table below.

Table 32.31 Operations for Each of the ENCA_nCTS Settings

ENCA _n CCR1 Function	Capture Trigger Selection			
ENCA _n CRM1	ENCA _n CTS	Capture Trigger Signal	Timer Counter Clearing	Interrupt Occurrence
1 (Capture)	0	Capture trigger 1 input (ENCATTIN1)	Does not clear timer counter.	(1) Capture 1 interrupt (ENCATINT1)
	1	Encoder clear input (set with ENCA _n SCE)	Clears timer counter.	(1) Capture 1 interrupt (ENCATINT1) (2) Encoder clear interrupt (ENCATIEC)

NOTE

For details about the timing chart when ENCA_nCTS = 0 or ENCA_nCTS = 1, see the following: “**Section 32.6.3, Count Clearing and Capture Operation by Encoder Clear Input (ENCA_nEC Pin)**”, “**Section 32.6.4, Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCA_nEC Pin)**”, “**Section 32.6.5, Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCA_nEC Pin)**”, “**Section 32.6.11, Conflict between ENCA_nLDE Function (Loading Count Value) and Clear Operation by Encoder Clear Input (ENCA_nEC Pin)**” and “**Section 32.6.12, Up-count after Conflict between ENCA_nLDE Function (Loading Count Value) and Encoder Clearing**”.

32.4.5.3 Timer Counter Clearing upon Compare Register Match

Timer counter clearing upon compare match between the values of the timer counter and the ENCA_nCCR0/1 setting value, according to the settings of the ENCA_nECM1 and ENCA_nECM0 bits in the ENCA_nCTL register, is detailed in the following table.

Table 32.32 Timer Counter Clearing Operation upon Compare Register Match

ENCA _n ECM1 and ENCA _n ECM0	Next Count Operation	Timer Counter Clearing upon Compare Match with ENCA _n CCR1	Timer Counter Clearing upon Compare Match with ENCA _n CCR0
00	Up-count	Does not clear (continues count operation).	Does not clear (continues count operation).
	Down-count	Does not clear (continues count operation).	Does not clear (continues count operation).
01	Up-count	Does not clear (continues count operation).	Clears timer counter to 0000 _H .
	Down-count	Does not clear (continues count operation).	Does not clear (continues count operation).
10	Up-count	Does not clear (continues count operation).	Does not clear (continues count operation).
	Down-count	Clears timer counter to 0000 _H .	Does not clear (continues count operation).
11	Up-count	Does not clear (continues count operation).	Clears timer counter to 0000 _H .
	Down-count	Clears timer counter to 0000 _H .	Does not clear (continues count operation).

32.5 ENCA_n Setting Sequences

32.5.1 ENCA_n Setting Procedure

The ENCA_n setting procedure is described below.

Table 32.33 ENCA_n Setting Procedure

Initial Setting	Action	Setting status
Initial setting	Reset deassertion	Power-on, operation stopped. (Writing to each register is enabled)
ENCA _n initial setting	Perform the following initial settings. <ul style="list-style-type: none"> • Setting for counter • Setting for counter clear • Setting for ENCA_nCCR0 register • Setting for ENCA_nCCR1 register 	Count operation stopped. The value of the ENCA _n TE bit indicating the operating status is 0.
	Perform the counter initial value settings. <ul style="list-style-type: none"> • Set any 16-bit value to ENCA_nCNT register. (When, after setting this register, the ENCA_nTS bit is set to "1", the counter operation starts from the set count value.) 	The value is set as the initial value of the counter register.
Operation start	Start the counter operation. <ul style="list-style-type: none"> • Set the ENCA_nTS bit to "1". 	Counter operation started. The value of the ENCA _n TE bit indicating the operating status is 1, and the count clock is supplied to the internal circuit.
Operating	Only those registers whose setting can be changed during operation can be rewritten. <ul style="list-style-type: none"> • ENCA_nCCR0 register setting. • ENCA_nCCR1 register setting. • ENCA_nIOC0 register setting. 	The count operation set with the initial setting is performed, and up/down counting is performed according to ENCA _n E0 and ENCA _n E1 pins.
Operation stop	Stop the counter operation during operation. <ul style="list-style-type: none"> • Set the ENCA_nTT bit to "1". 	Counter operation stopped. The value of the ENCA _n TE bit indicating the operating status is 0.
ENCA _n stop	Reset	The setting registers are initialized.

32.5.1.1 Initial Setting Procedure for the Counter

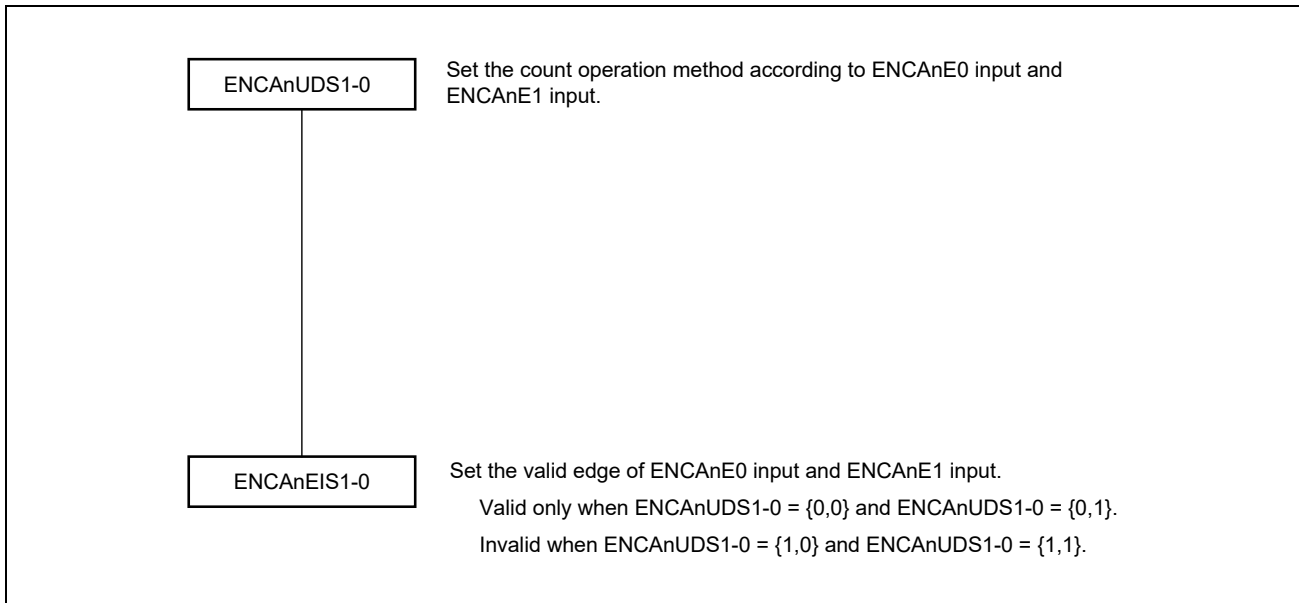


Figure 32.7 Initial Setting Procedure for the Counter

32.5.1.2 Initial Setting Procedure for Counter Clearing

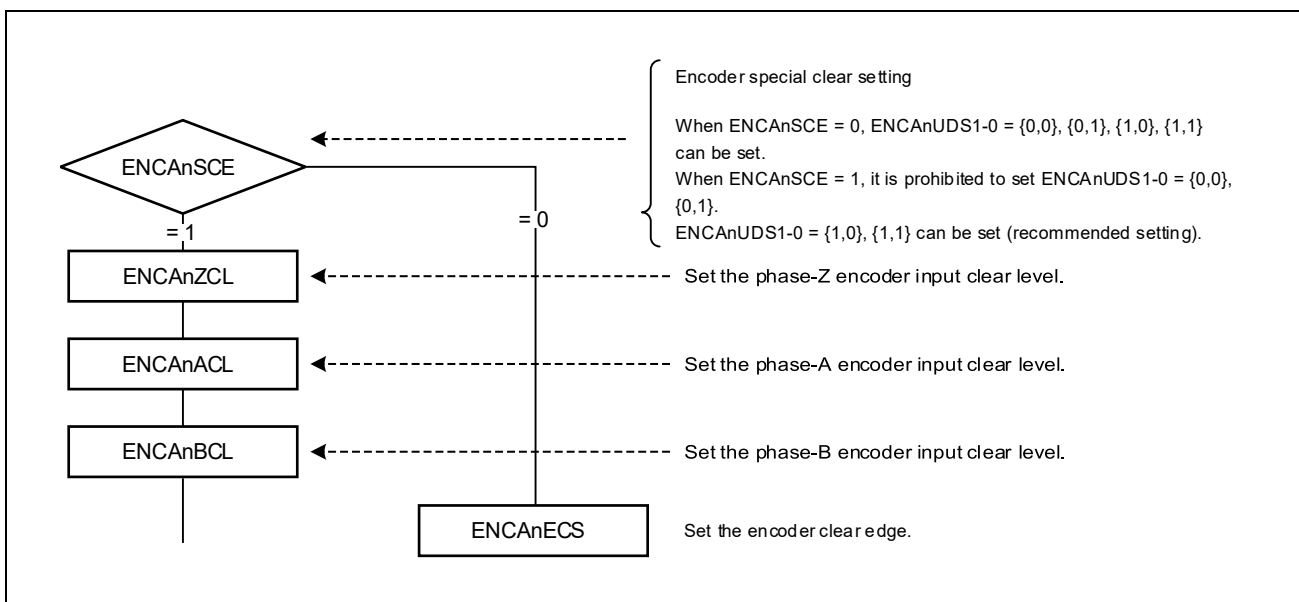


Figure 32.8 Initial Setting Procedure for Counter Clearing

32.5.1.3 Setting Procedure for ENCA_nCCR0 Register

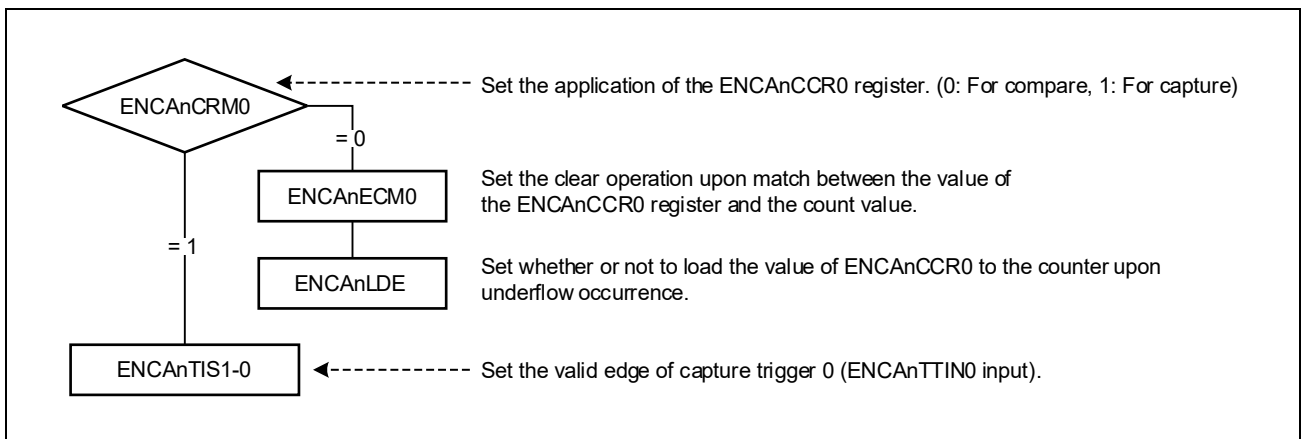


Figure 32.9 Setting Procedure for ENCA_nCCR0 Register

32.5.1.4 Setting Procedure for ENCA_nCCR1 Register

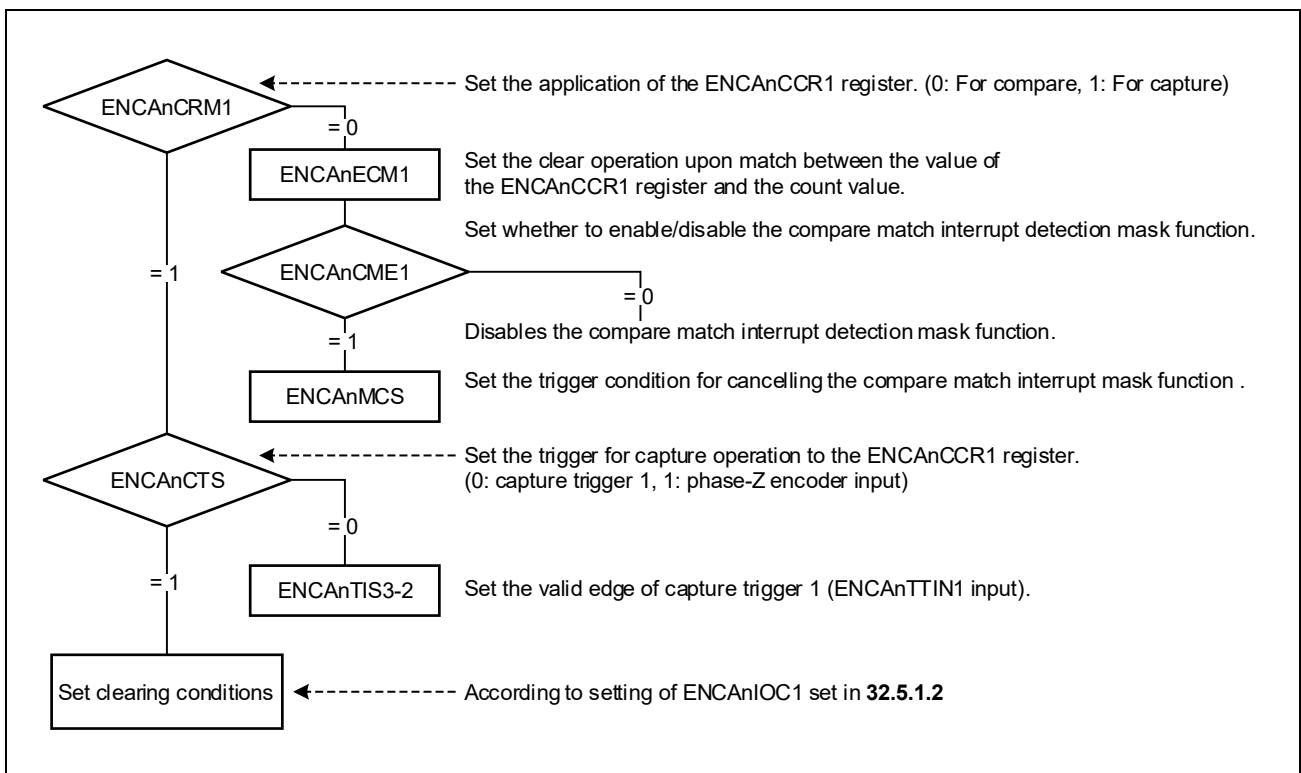


Figure 32.10 Setting Procedure for ENCA_nCCR1 Register

32.6 Timing Chart

32.6.1 Overflow Occurrence and Overflow Flag Clear Operation

An overflow occurs when up-counting is performed when the count value is $FFFF_H$. Once an overflow occurs, an overflow interrupt (ENCATIOV) is output and the overflow flag (ENCA_nOVF) is set to 1. When the overflow clear bit (ENCA_nCLOV) is set to 1, the overflow flag (ENCA_nOVF) is cleared to 0.

The operations of overflow occurrence and overflow flag clearing are described below.

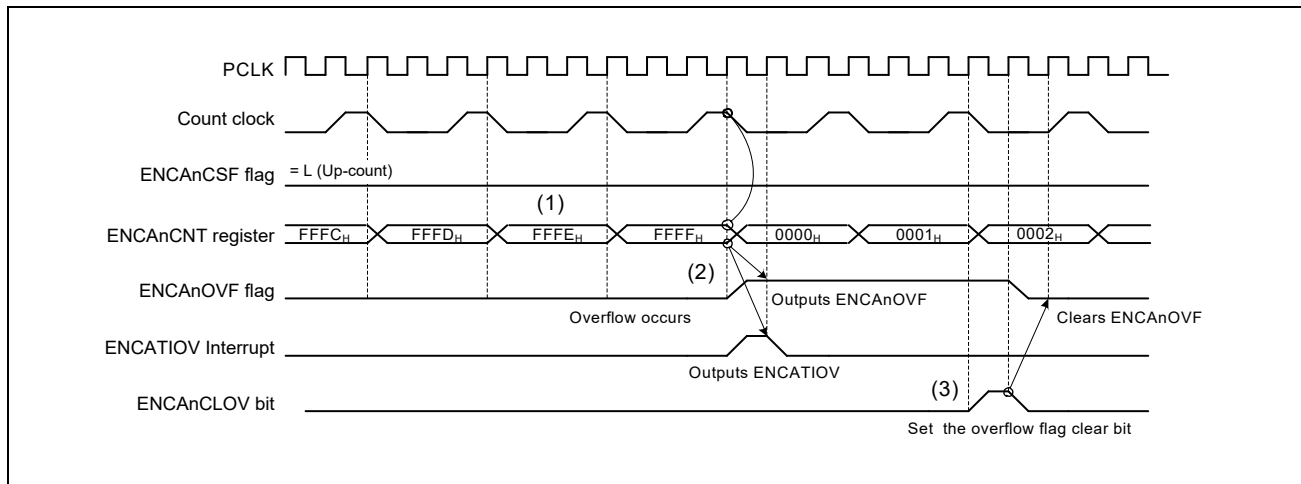


Figure 32.11 Settings of Overflow Occurrence and Overflow Flag Clear Operation

- (1) The count value is counted up from $FFFE_H$ to $FFFH$.
- (2) When the count value changes from $FFFH$ to 0000_H , an overflow occurs. At the same time, an overflow interrupt is output and the overflow flag is set to 1.
- (3) By setting the ENCA_nCLOV bit in the ENCA_nFGC register to 1 using the overflow flag clearing method, the overflow flag is cleared to 0. The overflow flag is also cleared by setting the ENCA_nTS bit in the ENCA_nTS register to 1 when the ENCA_nTE bit in the ENCA_nTE register is 0.

32.6.2 Underflow Occurrence and Underflow Flag Clear Operation

An underflow occurs when down-counting is performed when the count value is 0000_H.

Once an underflow occurs, an underflow interrupt (ENCATIUD) is output and the underflow flag (ENCAnUDF) is set to 1. When the underflow clear bit (ENCAnCLUD) is set to 1, the underflow flag (ENCAnUDF) is cleared to 0.

The operations of underflow occurrence and underflow flag clearing are described below.

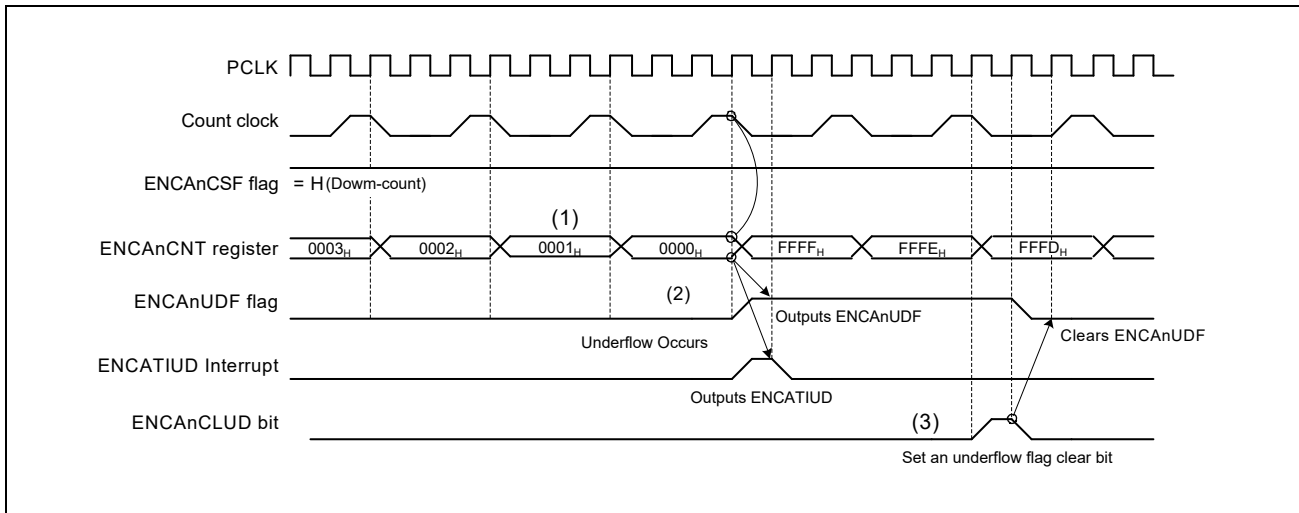


Figure 32.12 Settings of Underflow Occurrence and Underflow Flag Clear Operation

- (1) The count value is counted down from 0001_H to 0000_H.
- (2) When the count value changes from 0000_H to FFFF_H, an underflow occurs. At the same time, an underflow interrupt is output and the underflow flag is set to 1.
- (3) By setting the ENCAncCLUD bit in the ENCAncFGC register to 1 using the underflow flag clearing method, the underflow flag is cleared to 0. The underflow flag is also cleared by setting the ENCAncTS bit in the ENCAncTS register to 1 when the ENCAncTE bit in the ENCAncTE register is 0.

32.6.3 Count Clearing and Capture Operation by Encoder Clear Input (ENCAnEC Pin)

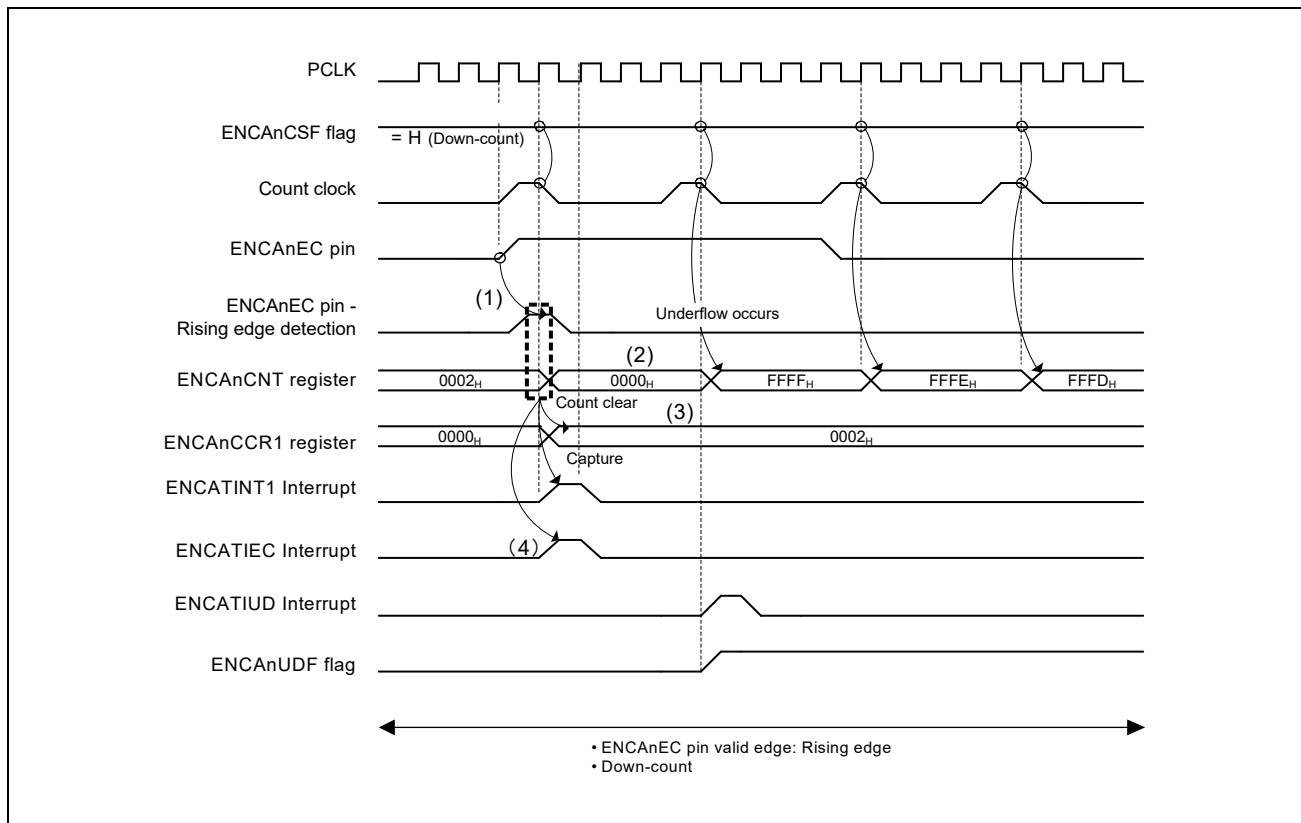


Figure 32.13 Timing Chart of Count Clearing and Capture Operation by Encoder Clear Input (ENCAnEC Pin)

Setting conditions

- ENCAAnCRM1 bit in the ENCAAnCTL register = 1
(Select the ENCAAnCCR1 register as capture.)
- ENCAAnCTS bit in the ENCAAnCTL register = 1
(Select the ENCAAnEC pin input as capture trigger input.)
- ENCAAnECS1 and ENCAAnECS0 bits in the ENCAAnIOC1 register = 01_B
(Select the ENCAAnEC pin input as rising edge detection.)

- (1) Capture operation is performed by the rising edge of the ENCAAnEC pin input trigger.
- (2) Clearing is performed by the ENCAAnEC pin input and the counter value is set to 0000_H.
- (3) The count value (0002_H) is captured in the ENCAAnCCR1 register by the rising edge of the ENCAAnEC pin input.
- (4) At the same time, a clear interrupt (ENCATIEC) and capture interrupt (ENCATINT1) due to the ENCAAnEC pin input are output.

32.6.4 Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

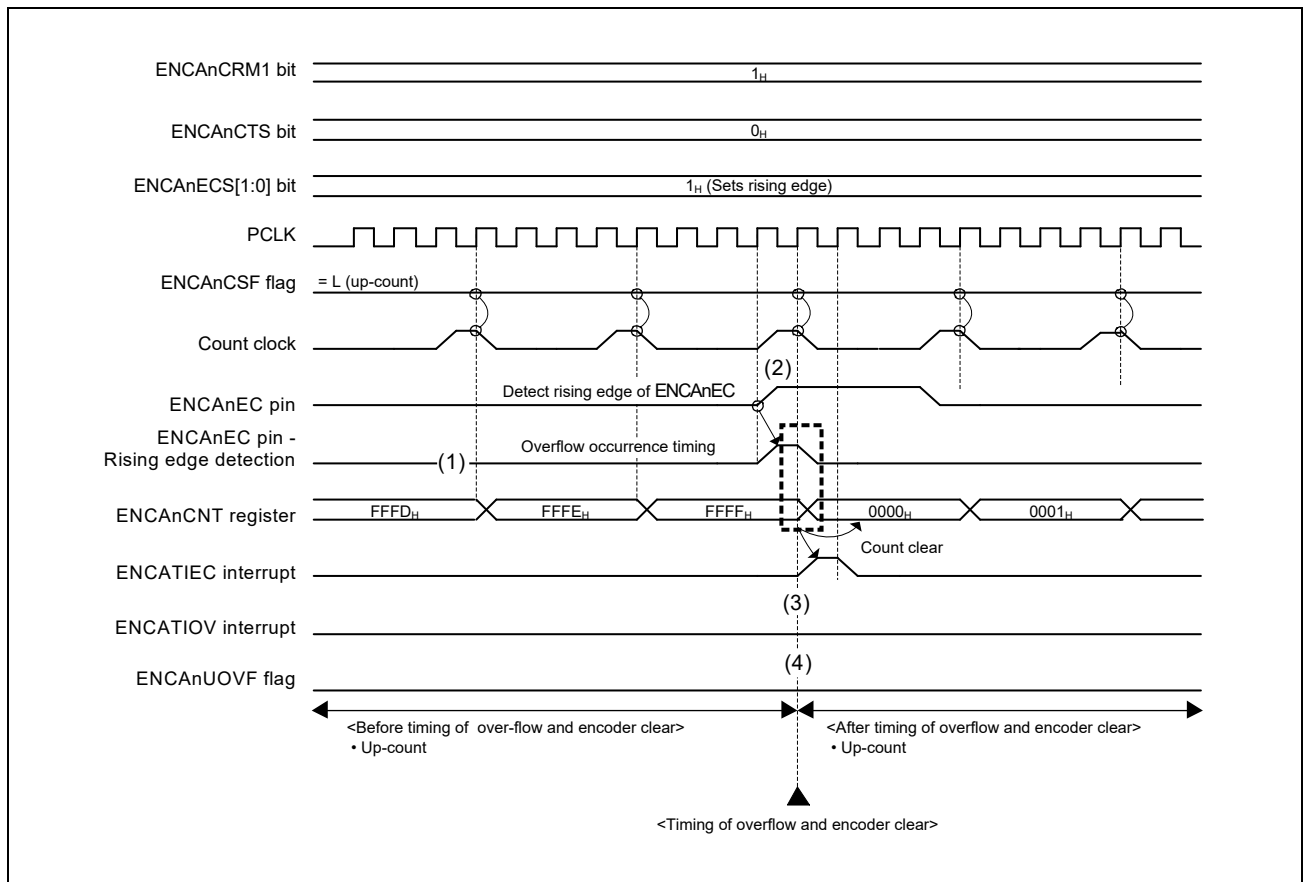


Figure 32.14 Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC pin)

- (1) An up-count from FFFD_H is continuously performed.
- (2) When an overflow occurs if the count value is FFFF_H, and the rising edge of ENCAnEC is detected simultaneously, clear operation by the encoder clear input is performed. The count value is cleared to 0000_H.
- (3) When the counter value is cleared by the encoder clear input, a clear interrupt (ENCATIEC) by encoder clear input is output simultaneously. Because a clear operation by the encoder clear input is performed simultaneously with the overflow occurrence, an overflow interrupt is not output. (An overflow does not occur. Clear operation is performed by the encoder clear input.)
- (4) Because an overflow does not occur as is the case with step 3, the overflow flag is not set.

32.6.5 Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCA_nEC Pin)

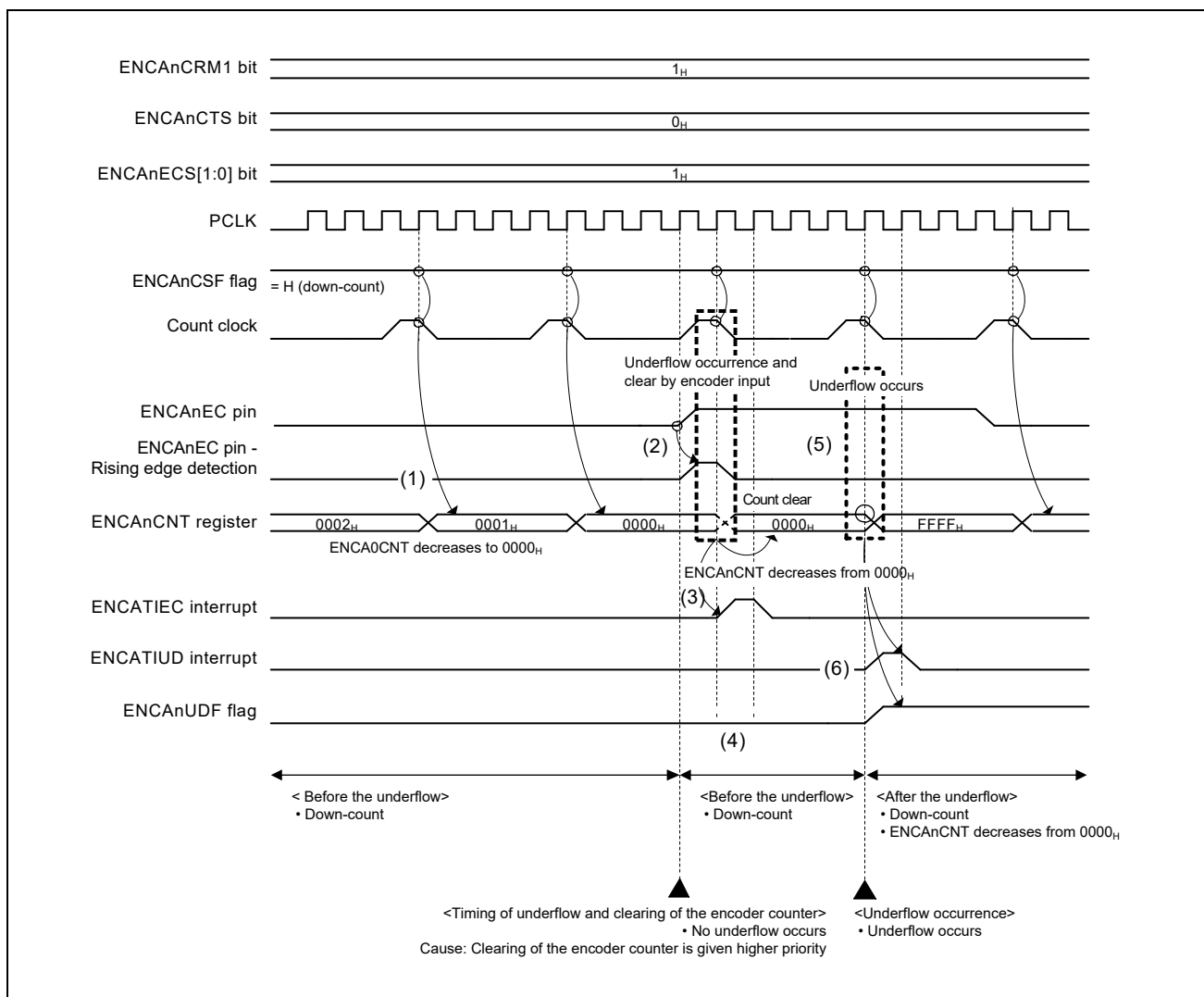


Figure 32.15 Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCA_nEC Pin)

- (1) A down-count from 0002_H is continuously performed.
- (2) When an underflow occurs if the count value is 0000_H, and the rising edge of ENCA_nEC is detected simultaneously, clear operation by the encoder clear input is performed. Even if the next clock signal is input during clear operation, the counter value remains at 0000_H.
- (3) When the counter value is cleared by the encoder clear input, an encoder clear interrupt (ENCA_nTIEC) is output simultaneously. Because a clear operation by the encoder clear input is performed simultaneously with the underflow occurrence, an underflow interrupt is not output. (An underflow does not occur. Clear operation is performed by the encoder clear input.)
- (4) Because an underflow does not occur as is the case with step 3, the underflow flag is not set.
- (5) When a further down-count is performed after the count value changes to 0000_H by clear operation of the encoder clear input, the count value changes from 0000_H to FFFF_H, and an underflow occurs.
- (6) When an underflow occurs, an underflow interrupt (ENCA_nTIU_D) is output, and the underflow flag (ENCA_nUDF) is set.

32.6.6 Overflow Operation Immediately after Startup

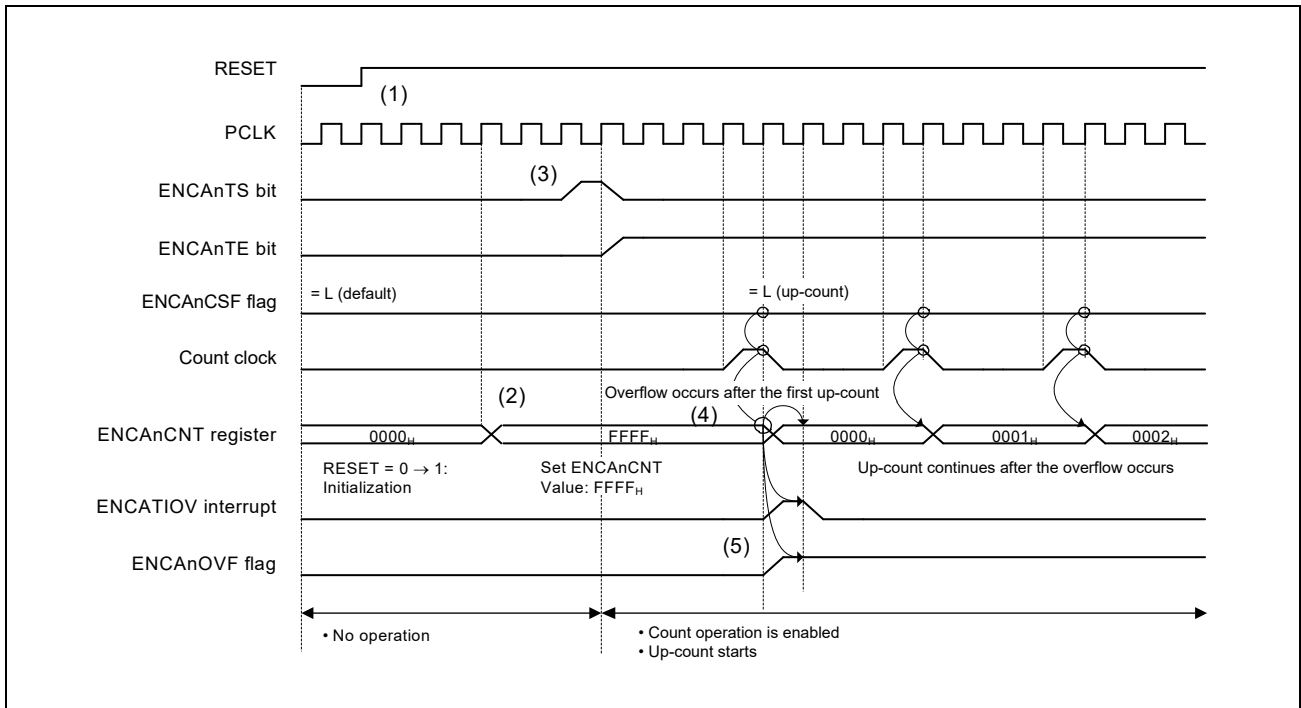


Figure 32.16 Overflow Operation Immediately after Startup

- (1) When the RESET value changes from “0” to “1”, the status changes from reset asserted to reset deasserted.
- (2) The timer counter is set to FFFF_H as the initial value.
- (3) ENCAAnTS is set to “1”, and operation starts. ENCAAnTE changes to “1”, which indicates that operation is enabled.
- (4) When an up-count is performed from FFFF_H which is the initially set count value, the count value changes from FFFF_H to 0000_H, and an overflow occurs immediately after operation starts.
- (5) At the same time, by an overflow occurrence immediately after operation starts, an overflow interrupt (ENCAAnTIOV) is output, and the overflow flag (ENCAAnOVF) is set.

32.6.7 Underflow Operation Immediately after Startup

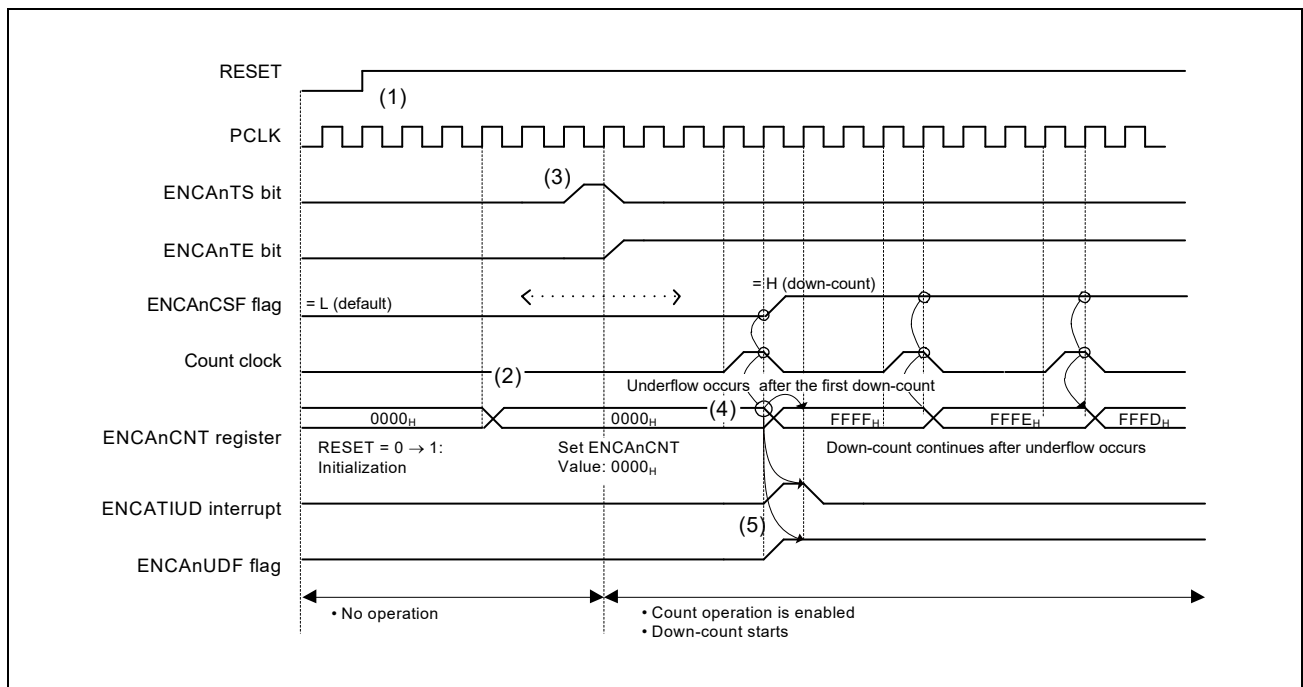


Figure 32.17 Underflow Operation Immediately after Startup

- (1) When the RESET value changes from “0” to “1”, the status changes from reset asserted to reset deasserted.
- (2) The timer counter is set to 0000_H as the initial value.
- (3) ENCAAnTS is set to “1”, and operation starts. ENCAAnTE changes to “1”, which indicates that operation is enabled.
- (4) When a down-count is performed from 0000_H which is the initially set count value, the count value changes from 0000_H to $FFFF_H$, and an underflow occurs immediately after operation starts.
- (5) At the same time, by an underflow occurrence immediately after operation starts, an underflow interrupt (ENCAAnTIUD) is output, and the underflow flag (ENCAAnUDF) is set.

32.6.8 Using the ENCANLDE Function Immediately after Startup

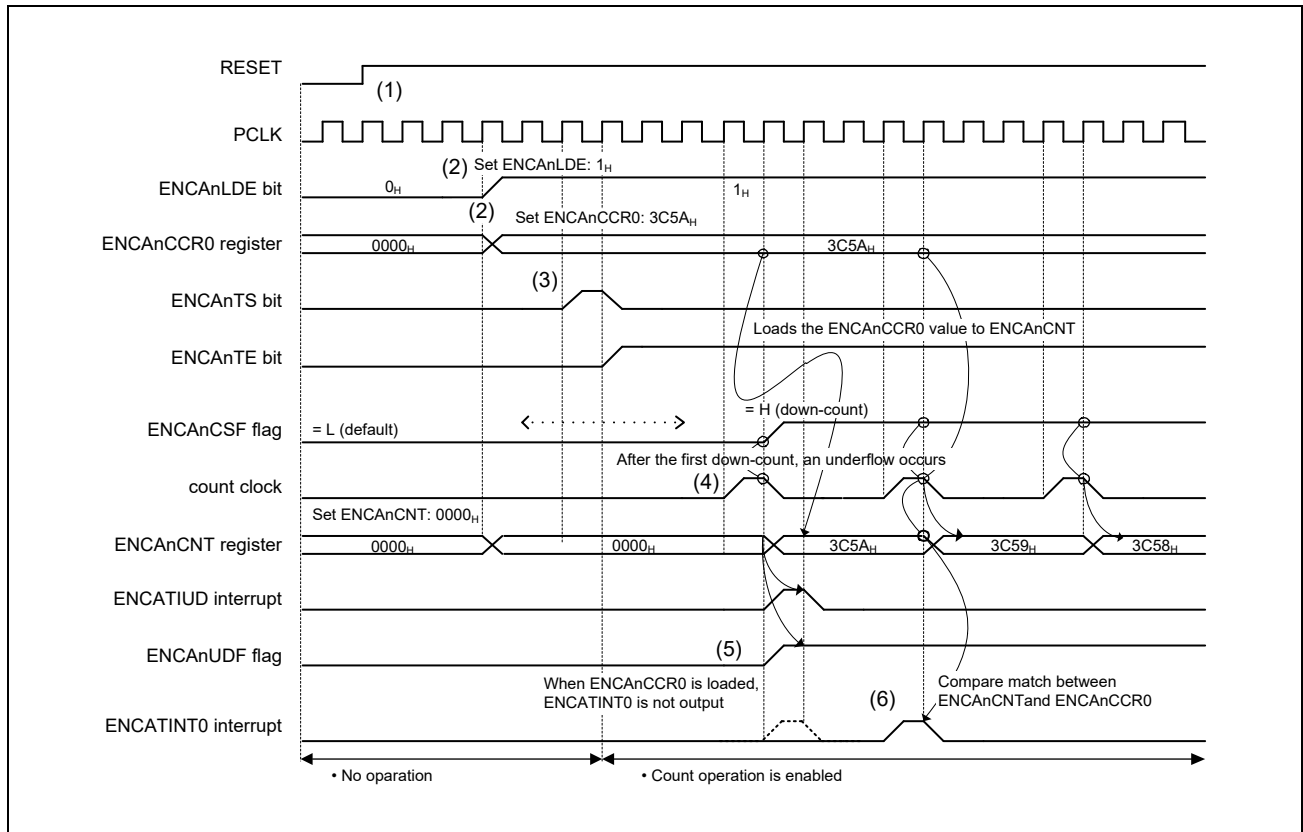


Figure 32.18 Using the ENCANLDE Function Immediately after Startup

- (1) When the RESET value changes from “0” to “1”, the status changes from reset asserted to reset deasserted”.
- (2) The load enable bit (ENCANLDE) is set to “1”, capture/compare register 0 (ENCANCCR0) is set to 3C5A_H, and the timer counter is set to the initial value 0000_H.
- (3) ENCANTS is set to “1”, and operation starts. ENCANTE changes to “1”, which indicates that operation is enabled.
- (4) When a down-count is performed from 0000_H which is the initially set count value, an underflow occurs immediately after operation starts. Because ENCANLDE is set to “1”, the ENCANCCR0 value, 3C5A_H, is loaded to the timer counter (ENCATINT0 is not output during loading).
- (5) At the same time, by an underflow occurrence immediately after operation starts, an underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCANUDF) is set (after an underflow occurs, down-count operation from the loaded value (3C5A_H) continues).
- (6) After the ENCANCCR0 value is loaded to ENCANCNT, a match with ENCANCCR0 is detected, and ENCATINT0 is output.

32.6.9 ENCANLDE Function (Loading Count Value)

(a) <When ENCANLDE = 0>

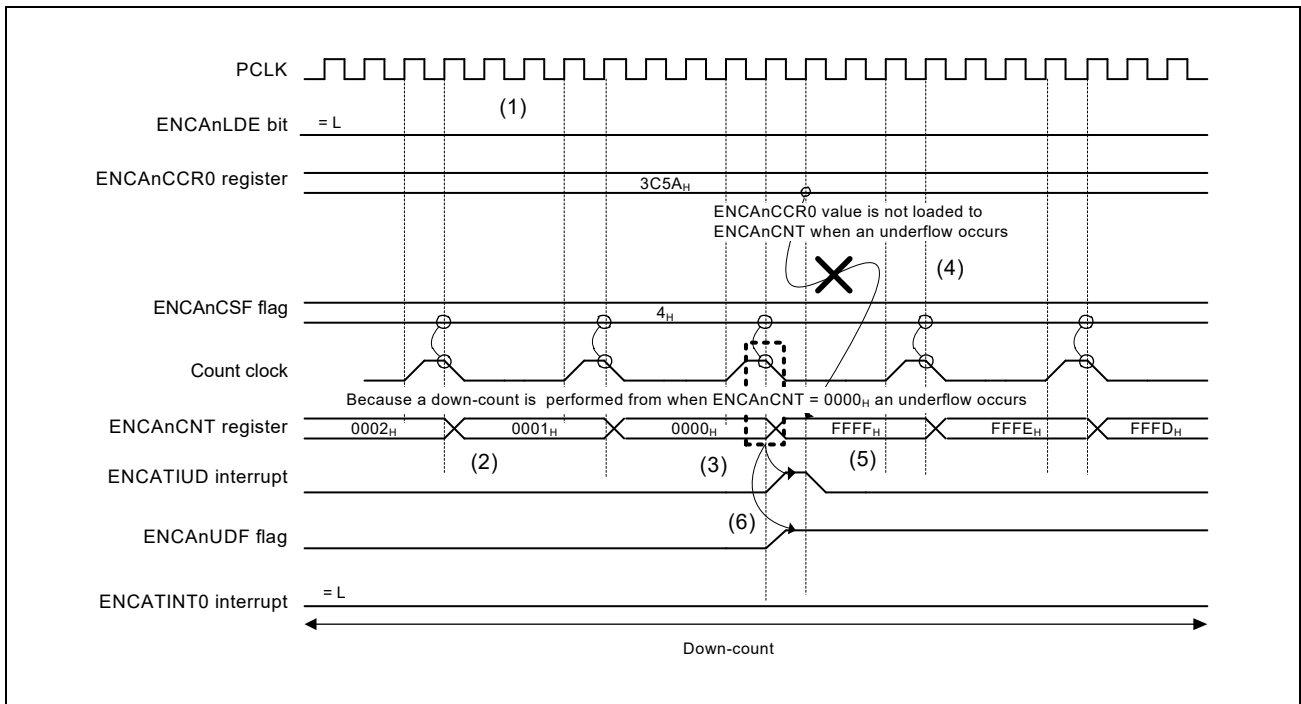
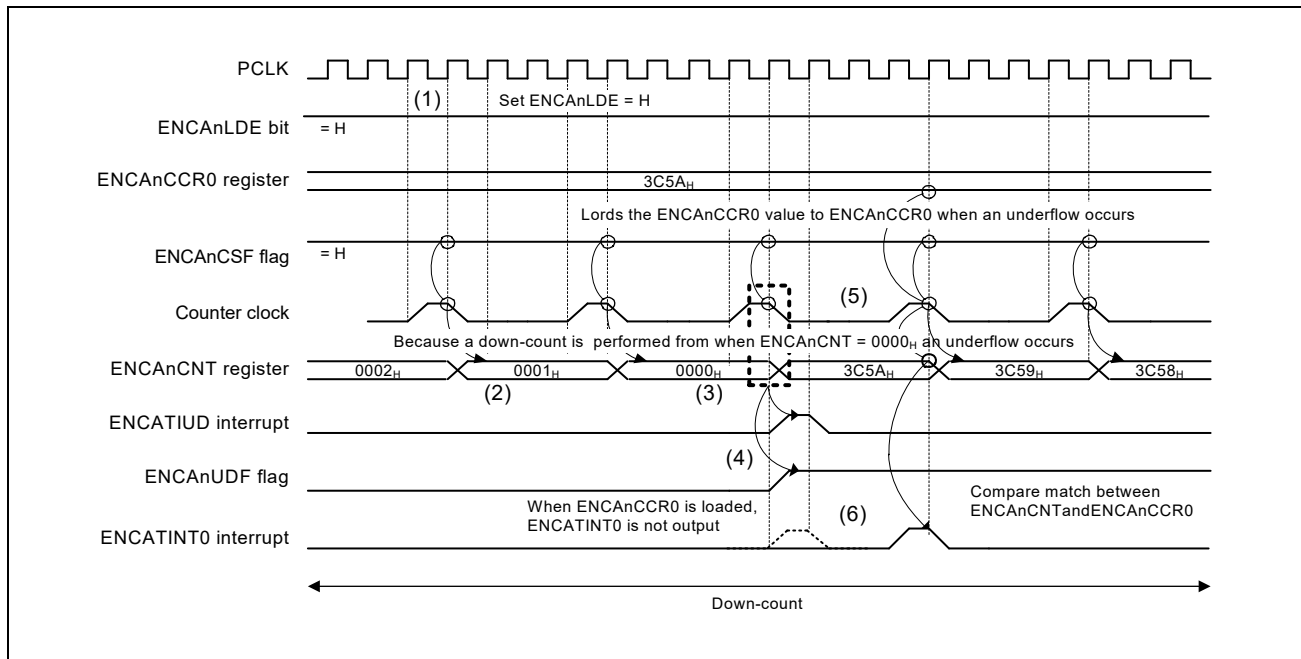


Figure 32.19 ENCANLDE Function (when ENCANLDE = 0)

- (1) ENCANLDE is set to "0" (even if an underflow occurs, the ENCANCCR0 value is not loaded).
- (2) A down-count is performed: 0002_H → 0001_H → 0000_H
- (3) When a further down-count is performed after the count value changes to 0000_H, an underflow occurs.
- (4) Because ENCANLDE is set to "0", the setting value of the ENCANCCR0 register is not loaded to the counter even if an underflow occurs.
- (5) Operation changes to underflow operation (count value: 0000_H → FFFF_H).
- (6) An underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCANUDF) is set.

(b) <When ENCA_nLDE = 1>Figure 32.20 ENCA_nLDE Function (when ENCA_nLDE = 1)

- (1) ENCA_nLDE is set to "1" (if an underflow occurs, the ENCA_nCCR0 value is loaded to the counter).
- (2) A down-count is performed: 0002_H → 0001_H → 0000_H
- (3) When a further down-count is performed after the count value changes to 0000_H, an underflow occurs.
- (4) An underflow interrupt is output, and the underflow flag is set.
- (5) Because ENCA_nLDE is set to "1", the setting value of the ENCA_nCCR0 register is loaded to the counter if an underflow occurs. ENCA_nCNT is set to 3C5A_H.
- (6) After the ENCA_nCCR0 value is set to ENCA_nCNT, if the ENCA_nCNT value matches the ENCA_nCCR0 value on a count clock, a compare match interrupt (ENCA_nTINT0) is output.

32.6.10 Conflict between ENCA_nLDE Function (Loading Count Value) and Rewrite of ENCA_nCCR0 Register

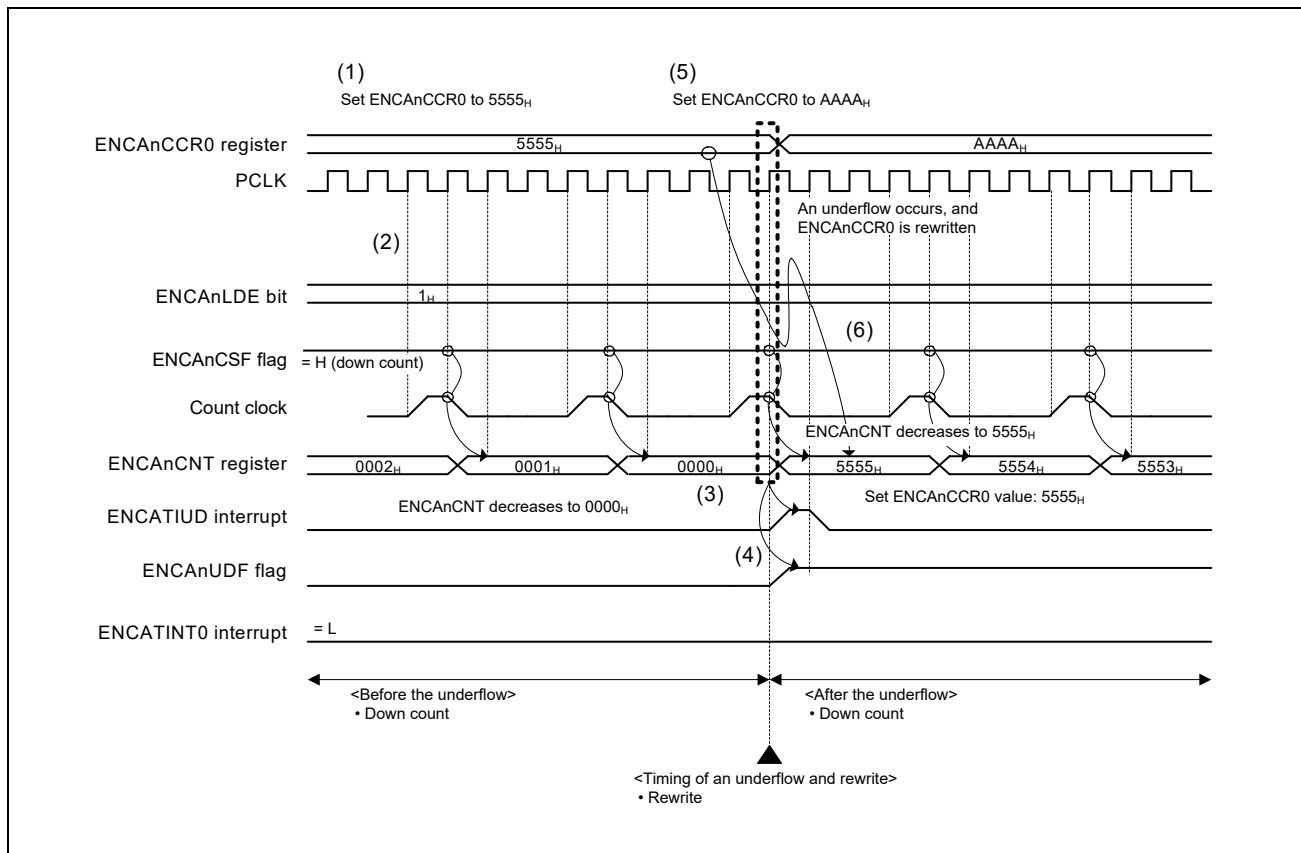


Figure 32.21 Conflict between ENCA_nLDE Function and Rewrite of ENCA_nCCR0 Register

- (1) The ENCA_nCCR0 register is currently set to 5555_H.
- (2) ENCA_nLDE is currently set to “1”.
- (3) A down-count is performed (0002_H → 0001_H → 0000_H), and an underflow occurs.
- (4) An underflow interrupt (ENCA_nTIUD) is output, and the underflow flag (ENCA_nUDF) is set.
- (5) When an underflow occurs, the ENCA_nCCR0 register value is changed from 5555_H to AAAA_H.
- (6) Additionally, when an underflow occurs, the ENCA_nCCR0 value before the rewrite is performed (5555_H) is set in ENCA_nCNT.

32.6.11 Conflict between ENCA_nLDE Function (Loading Count Value) and Clear Operation by Encoder Clear Input (ENCA_nEC Pin)

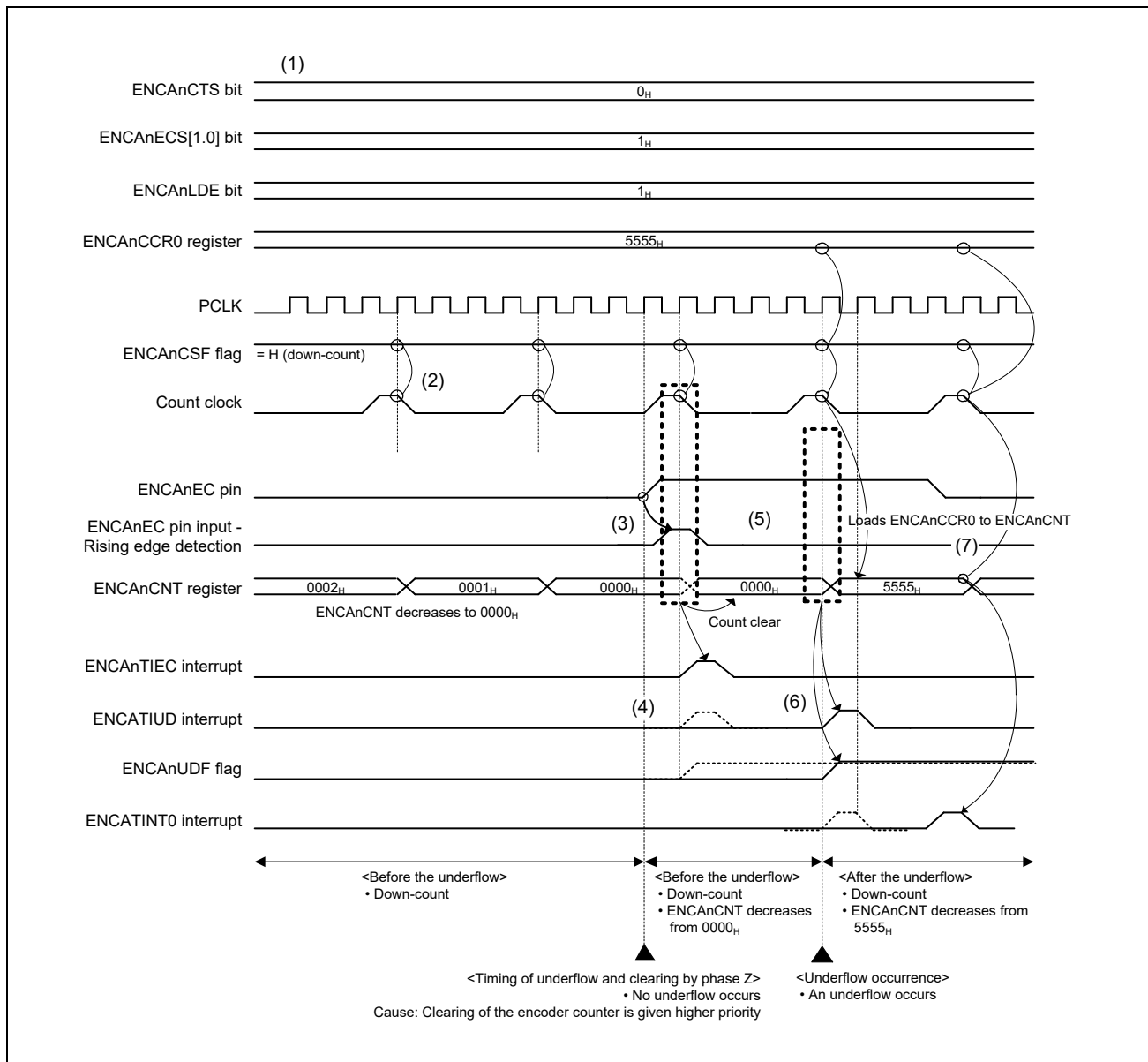


Figure 32.22 Conflict between ENCA_nLDE Function and Clear Operation by Encoder Clear Input

- (1) The values are set as follows: ENCA_nCTS = 0, ENCA_nECS[1:0] = 01_B, ENCA_nLDE = 1, and ENCA_nCCR0 = 5555_H.
- (2) A down-count is performed: 0002_H → 0001_H → 0000_H
- (3) When the count value becomes 0000_H, the rising edge of ENCA_nEC pin is detected, and clear operation by the encoder clear input is performed.
- (4) Because a count clear is performed when the count value reaches 0000_H, a counter clear interrupt (ENCA_nTIEC) by the encoder clear input is output. An underflow does not occur because a down-count is not performed when the count value is 0000_H. Therefore, an underflow interrupt (ENCA_nTIUD) is not output, and the underflow flag (ENCA_nUDF) is not set.

- (5) After the count value is cleared to 0000_H by clear operation of the encoder clear input, a down-count is performed and an underflow occurs.
- (6) An underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCA_nUDF) is set.
- (7) Because ENCA_nLDE = “1”, if an underflow occurs, the ENCA_nCCR0 value is loaded to ENCA_nCNT.
- (8) After the ENCA_nCCR0 value is set to ENCA_nCNT, a compare match is detected according to the count clock. If the ENCA_nCNT value matches the ENCA_nCCR0 value, a compare match interrupt (ENCATINT0) is output.

32.6.12 Up-count after Conflict between ENCA_nLDE Function (Loading Count Value) and Encoder Clearing

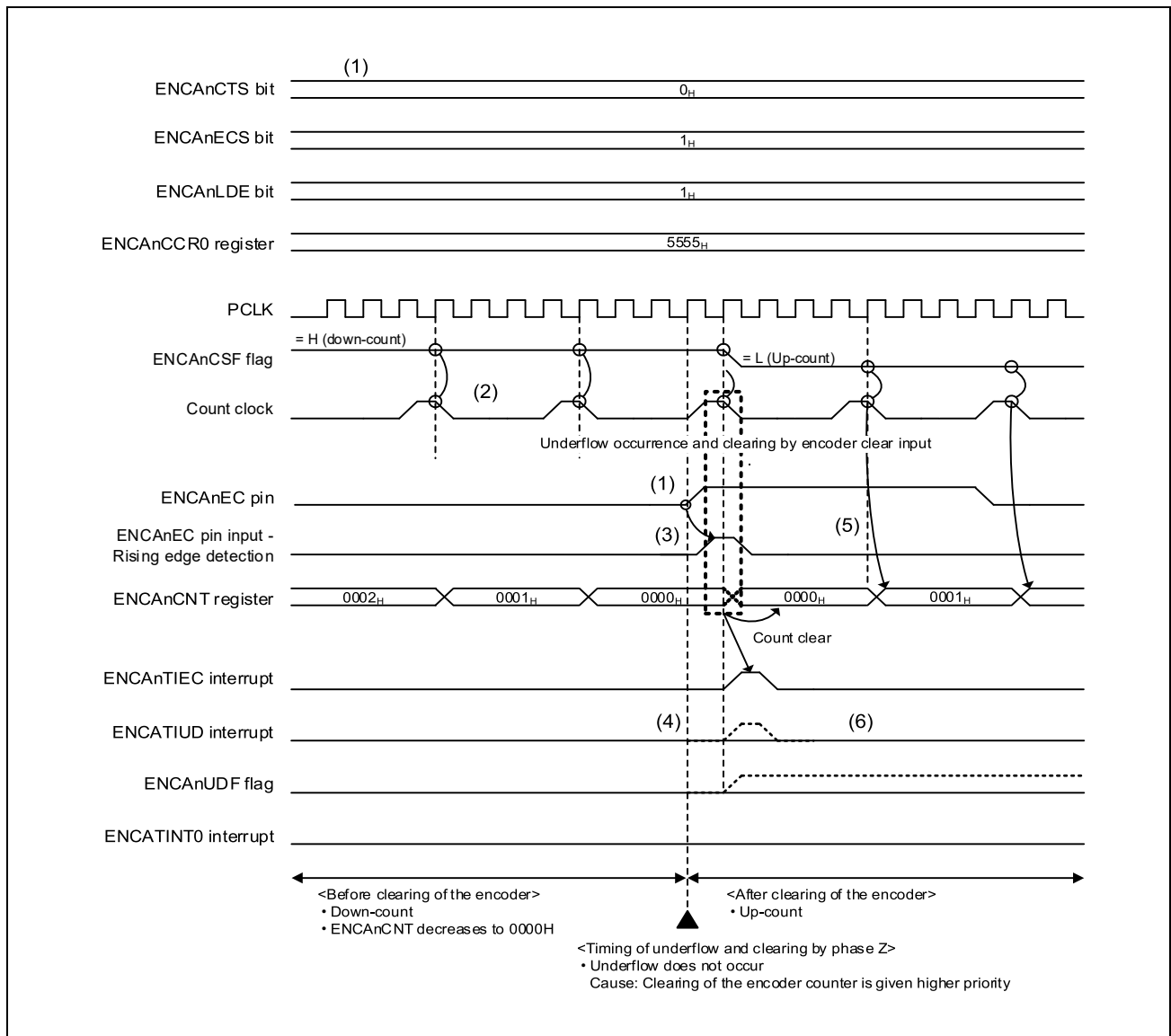


Figure 32.23 Up-count after Conflict between ENCA_nLDE Function and Encoder Clearing

- (1) The values are set as follows: ENCA_nCTS = 0, ENCA_nECS[1:0] = 01B, ENCA_nLDE = 1, and ENCA_nCCR0 = 5555_H.
- (2) A down-count is performed: 0002_H → 0001_H → 0000_H
- (3) When the count value becomes 0000_H, the rising edge of ENCA_nEC pin is detected, and clear operation by the encoder clear input is performed.
- (4) Because a count clear is performed when the count value reaches 0000_H, a counter clear interrupt (ENCA_nTIEC) by the encoder clear input is output. An underflow does not occur because a down-count is not performed when the count value is 0000_H. Therefore, an underflow interrupt (ENCA_nTIU0) is not output, and the underflow flag (ENCA_nUDF) is not set.
- (5) After the count value is cleared to 0000_H by clear operation of the encoder clear input, an up-count is performed.
- (6) An underflow interrupt (ENCA_nTIU0) is not output, and the underflow flag (ENCA_nUDF) is not set.

32.6.13 Capture Operation between Count Clocks (ENCA_nCCR1)

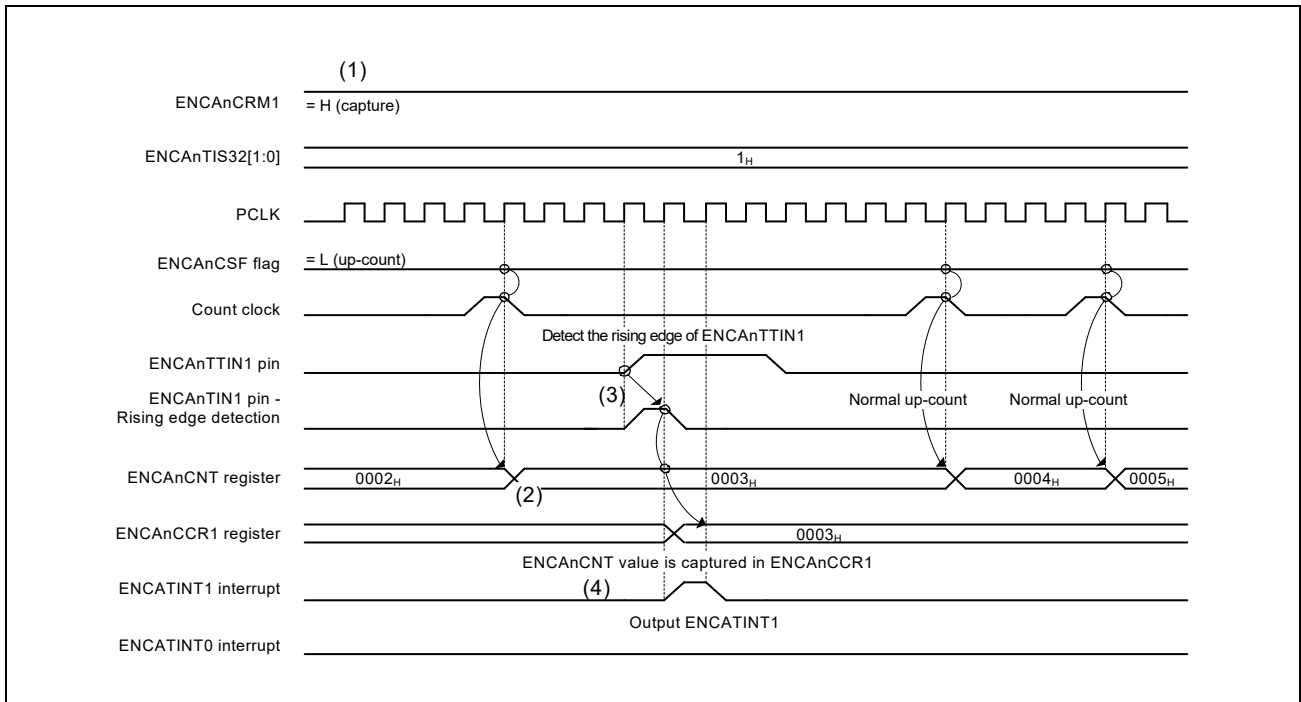


Figure 32.24 Capture Operation between Count Clocks (ENCA_nCCR1)

- (1) The values are set as follows: ENCA_nCRM1 = 1, and ENCA_nTIS[3:2] = 01_B.
- (2) An up-count is performed.
- (3) The rising edge of the ENCA_nTTIN1 input is detected, and the count value is captured in ENCA_nCCR1.
- (4) An interrupt (ENCA_nTINT1) corresponding to the capture to the ENCA_nCCR1 register is output.

32.6.14 Capture Operation between Count Clocks (ENCA_nCCR0)

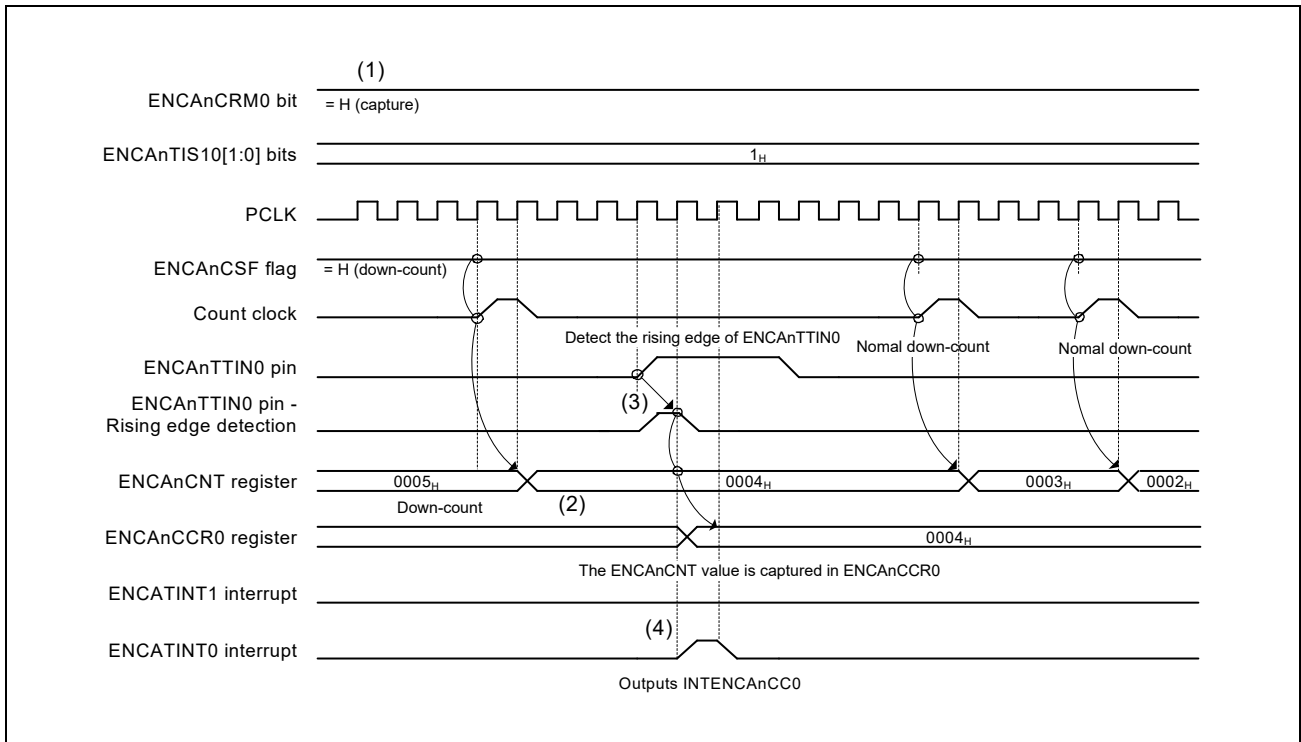


Figure 32.25 Capture Operation between Count Clocks (ENCA_nCCR0)

- (1) The values are set as follows: ENCA_nCRM0 = 1, and ENCA_nTIS[1:0] = 01_B.
- (2) A down-count is performed.
- (3) The rising edge of the ENCA_nTTIN0 input is detected, and the count value is captured in ENCA_nCCR0.
- (4) An interrupt (ENCA_nTINT0) corresponding to the capture to the ENCA_nCCR0 register is output.

32.6.15 Encoder Operation when Compare Match Clear Control is Enabled and ENCA_nCTS = 0

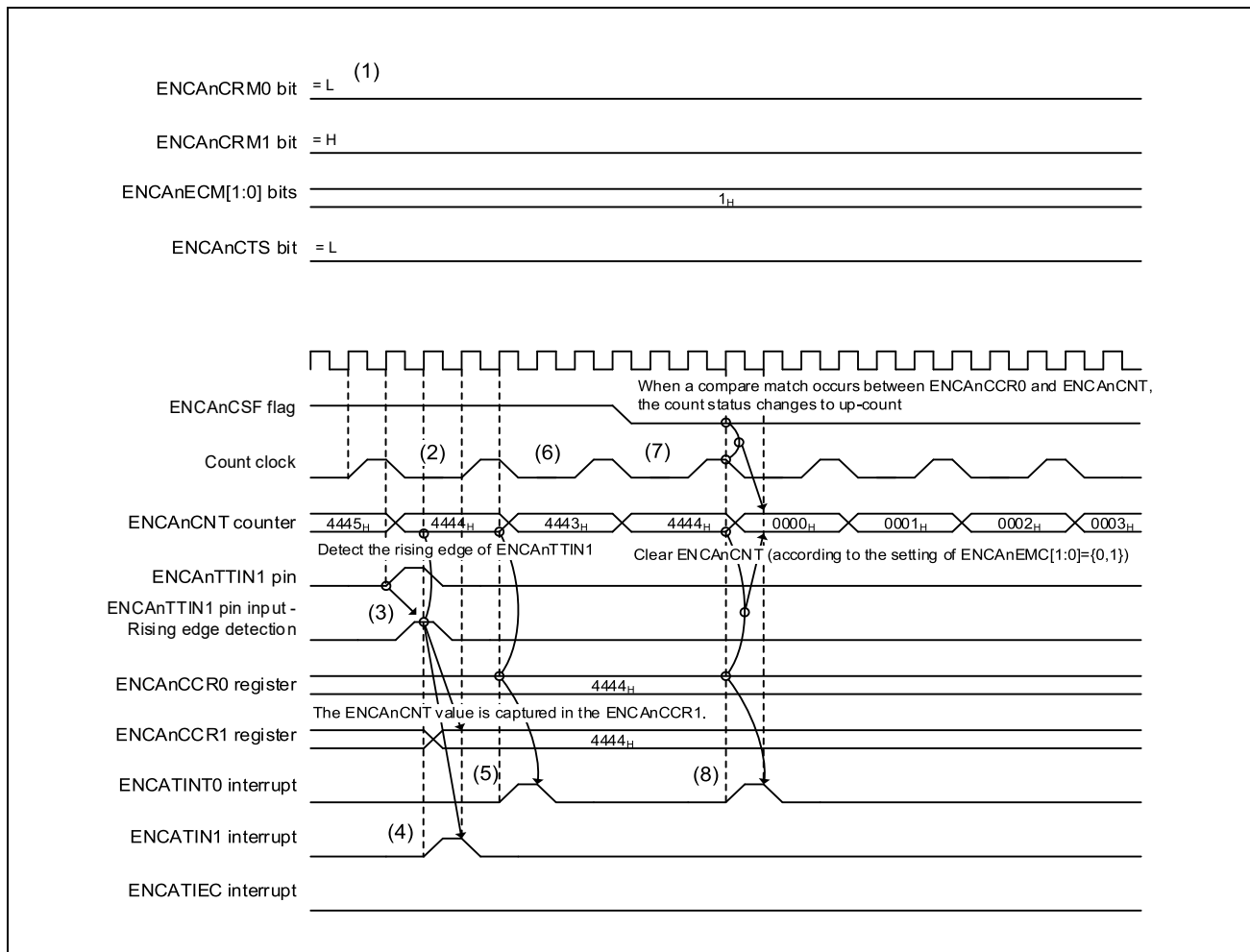


Figure 32.26 Encoder Operation when Compare Match Clear Control is Enabled and ENCA_nCTS = 0

- (1) The values are set as follows: ENCA_nCCR0 = 4444_H, ENCA_nCRM0 = 0, ENCA_nCRM1 = 1, ENCA_nECM[1:0] = 01_B, and ENCA_nCTS = 0.
- (2) A down-count is performed.
- (3) The rising edge of the ENCA_nTTIN1 input is detected, and the ENCA_nCNT value (4444_H) is captured in the ENCA_nCCR1 register.
- (4) An interrupt signal (ENCA_nTINT1) corresponding to the capture to the ENCA_nCCR1 register is output.
- (5) When a compare match occurs between ENCA_nCNT (counted down from 4445_H to 4444_H) and ENCA_nCCR0 (4444_H), a compare match interrupt (ENCA_nTINT0) with ENCA_nCCR0 is output.
- (6) The count operation changes to up-count.
- (7) When ENCA_nCNT is counted up from 4443_H to 4444_H, a compare match with ENCA_nCCR0 occurs again. Because the count operation is up-count when the compare match occurs, the count value is cleared according to the setting of ENCA_nECM1 and ENCA_nECM0 (01_B), and the ENCA_nCNT value changes to 0000_H.
- (8) When ENCA_nCNT changes to 4444_H, a compare match interrupt (ENCA_nTINT0) with ENCA_nCCR0 is output.

32.6.16 Encoder Operation when Compare Match Clear Control is Enabled and ENCA_nCTS = 1

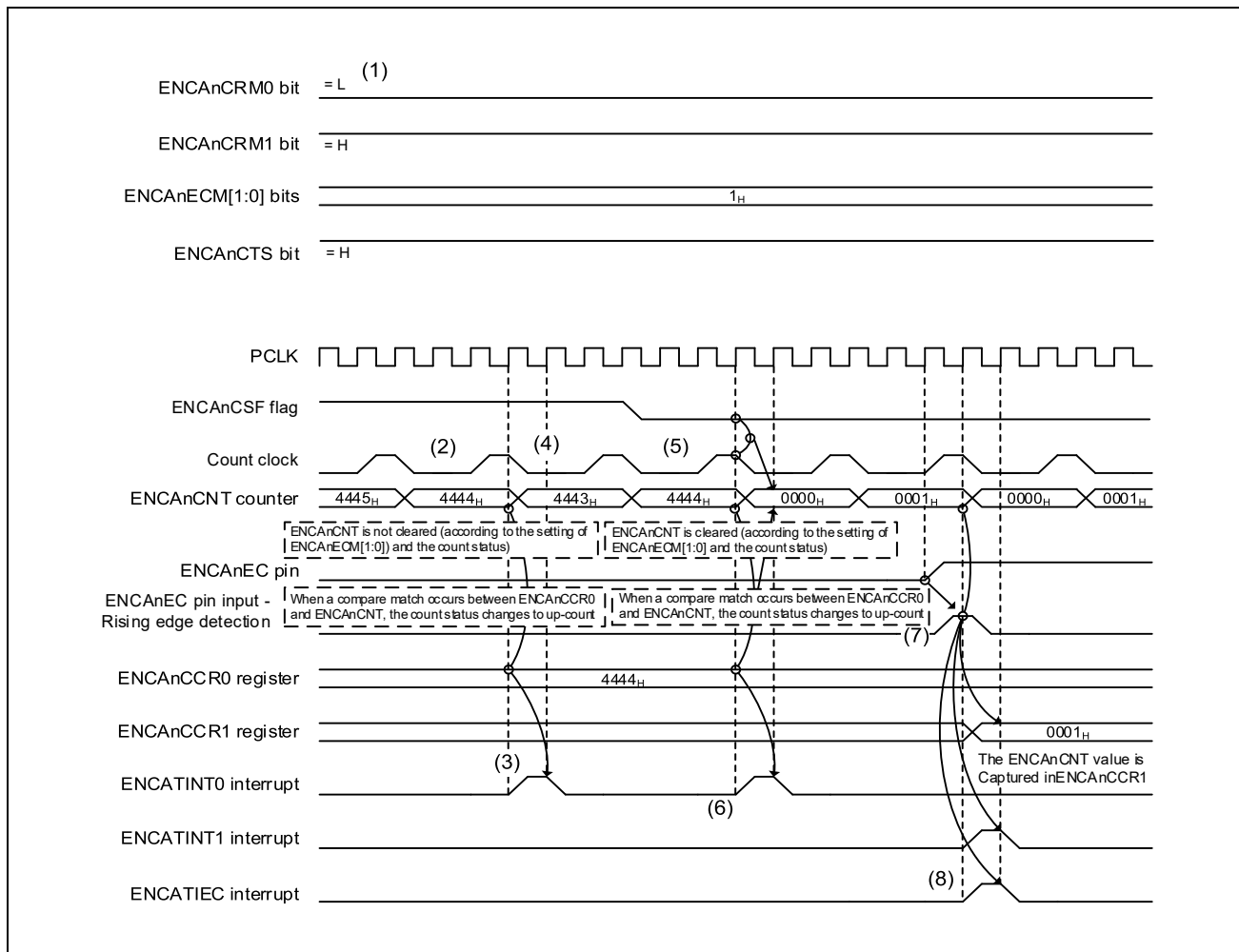


Figure 32.27 Encoder Operation when Compare Match Clear Control is Enabled and ENCA_nCTS = 1

- (1) The values are set as follows: ENCA_nCCR0 = 4444_H, ENCA_nCRM0 = 0, ENCA_nCRM1 = 1, ENCA_nECM[1:0] = 01_B, and ENCA_nCTS = 1.
- (2) A down-count is performed.
- (3) When a compare match occurs between ENCA_nCNT (counted down from 4445_H to 4444_H) and ENCA_nCCR0 (4444_H), a compare/capture interrupt (ENCA_nTINT0) is output.
- (4) The count operation changes to up-count.
- (5) When ENCA_nCNT is counted up from 4443_H to 4444_H, a compare match with ENCA_nCCR0 occurs again. Because the count operation is up-count when the compare match occurs, the count value is cleared according to the setting of ENCA_nECM1 and ENCA_nECM0 (01_B), and the ENCA_nCNT value changes to 0000_H.
- (6) When ENCA_nCNT changes to 4444_H, a compare match interrupt (ENCA_nTINT0) with ENCA_nCCR0 is output.
- (7) After the count value is cleared, an up-count is performed, and the count value changes to 0001_H. At this point, the ENCA_nCNT value (0001_H) is captured in ENCA_nCCR1 by detecting the rising edge of the ENCA_nEC signal, and the counter is cleared to 0000_H.
- (8) An interrupt (ENCA_nTINT1) corresponding to the capture to the ENCA_nCCR1 register and a clear interrupt (ENCA_nTIEC) by ENCA_nEC are output.

32.6.17 Encoder Operation when Compare Match Clear Control is Disabled

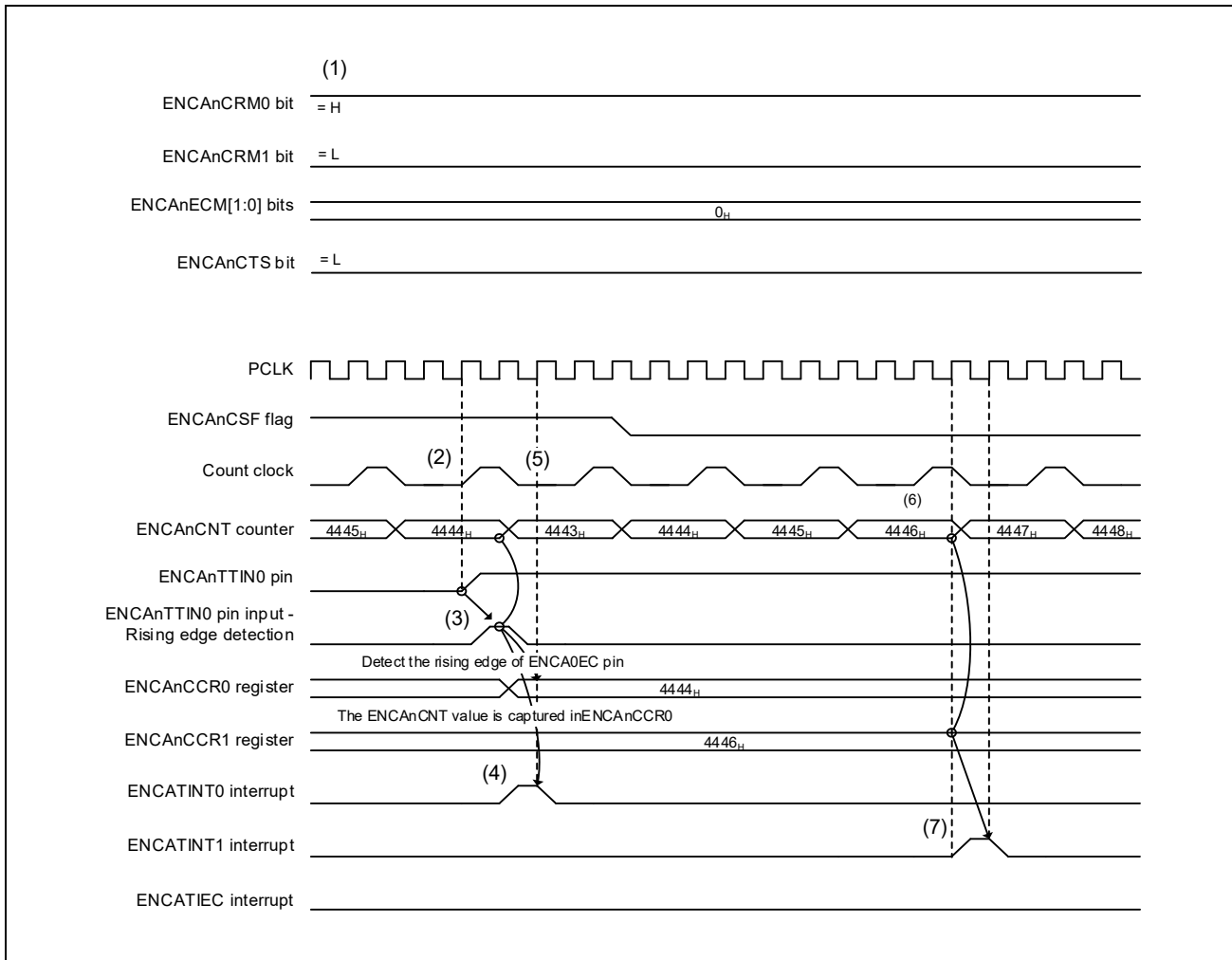


Figure 32.28 Encoder Operation when Compare Match Clear Control is Disabled

- (1) The values are set as follows: ENCAAnCCR1 = 4446_H, ENCAAnCRM0 = 1, ENCAAnCRM1 = 0, ENCAAnECM[1:0] = 00_B, and ENCAAnCTS = 0.
- (2) A down-count is performed.
- (3) When the rising edge of ENCAAnTTIN0 is detected, the ENCAAnCNT value (4444_H) is captured in ENCAAnCCR0.
- (4) An interrupt signal (ENCAAnTINT0) corresponding to the capture to the ENCAAnCCR0 register is output.
- (5) The count operation changes to up-count.
- (6) When ENCAAnCNT changes to 4446_H, a compare match with ENCAAnCCR1 is detected.
- (7) A compare match interrupt (ENCAAnTINT1) with ENCAAnCCR1 is output.

32.6.18 Capture Operation Performed upon Clearing by ENCA_nEC, ENCA_nE0, or ENCA_nE1 when ENCA_nSCE = 1

32.6.18.1 Accompanying Capture Operation

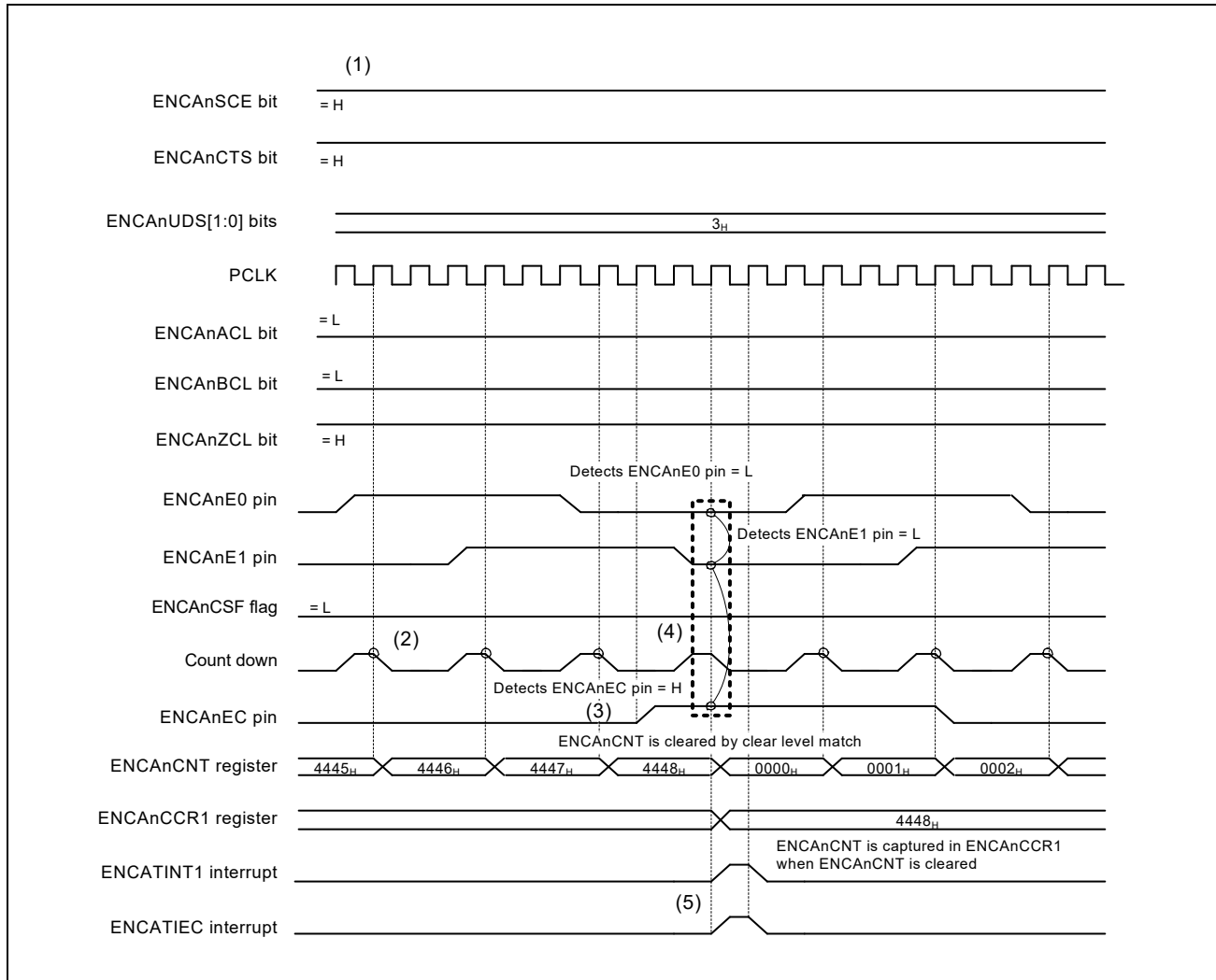


Figure 32.29 Capture Operation Performed upon Clearing by ENCA_nEC, ENCA_nE0, or ENCA_nE1 when ENCA_nSCE = 1

- (1) The values are set as follows: ENCA_nSCE = 1, ENCA_nCTS = 1, ENCA_nUDS[1:0] = 11_B, ENCA_nACL = 0, ENCA_nBCL = 0, and ENCA_nZCL = 1.
- (2) An up-count is performed.
- (3) The count value is not cleared upon the rising edge of ENCA_nEC.
- (4) When ENCA_nE0, ENCA_nE1 and ENCA_nEC reach the set clear level, the count value is cleared. The count value is captured in ENCA_nCCR1 at the time of the clearing.
- (5) At the time of the clearing, an interrupt (ENCA_nTINT1) corresponding to the capture to the ENCA_nCCR1 register and a clear interrupt (ENCA_nTIEC) by ENCA_nEC are output.

32.6.18.2 When the Timing of the ENCA_nEC Input is Later than that of the ENCA_nE1 Input during Up-count (When ENCA_nACL = 1, ENCA_nBCL = 0, ENCA_nZCL = 1, and ENCA_nUDS[1:0] = 11_B)

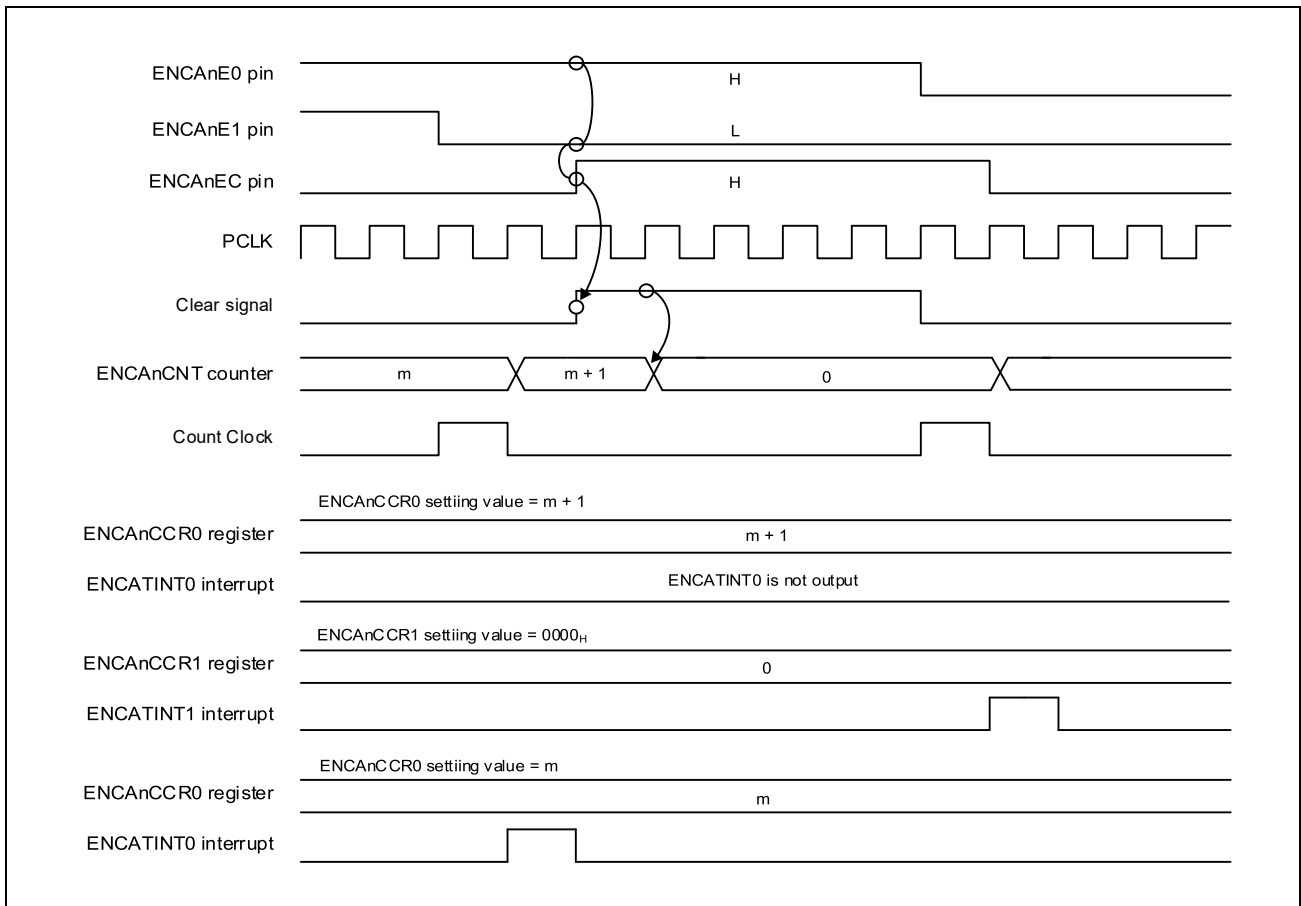


Figure 32.30 Clearing Timing for when the Timing of the ENCA_nEC Input is Later than that of the ENCA_nE1 Input during Up-count

32.6.18.3 When the Timing of the ENCA_nEC Input is the Same as that of the ENCA_nE1 Input during Up-count (When ENCA_nACL = 1, ENCA_nBCL = 0, ENCA_nZCL = 1, and ENCA_nUDS[1:0] = 11_B)

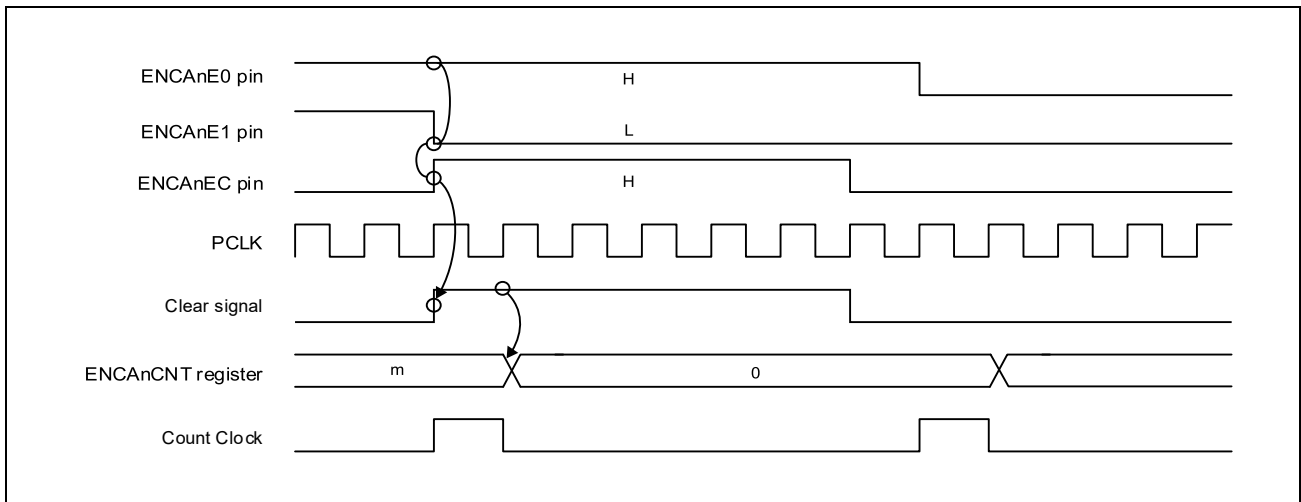


Figure 32.31 Clearing Timing for when the Timing of the ENCA_nEC Input is the Same as that of the ENCA_nE1 Input during Up-count

32.6.18.4 When the Timing of the ENCA_nEC Input is Earlier than that of the ENCA_nE1 Input during Up-count (When ENCA_nACL = 1, ENCA_nBCL = 0, ENCA_nZCL = 1, and ENCA_nUDS[1:0] = 11_B)

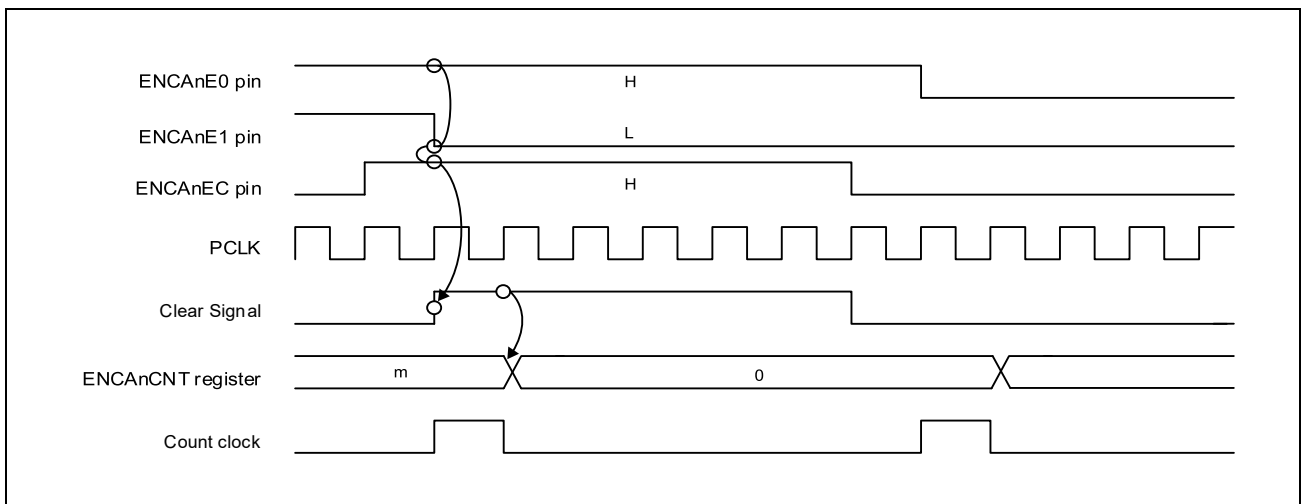


Figure 32.32 Clearing Timing for when the Timing of the ENCA_nEC Input is Earlier than that of the ENCA_nE1 Input during Up-count

32.6.18.5 When the Timing of the ENCA_nEC Input is Later than that of the ENCA_nE1 Input during Down-count (When ENCA_nACL = 1, ENCA_nBCL = 0, ENCA_nZCL = 1, and ENCA_nUDS[1:0] = 11_B)

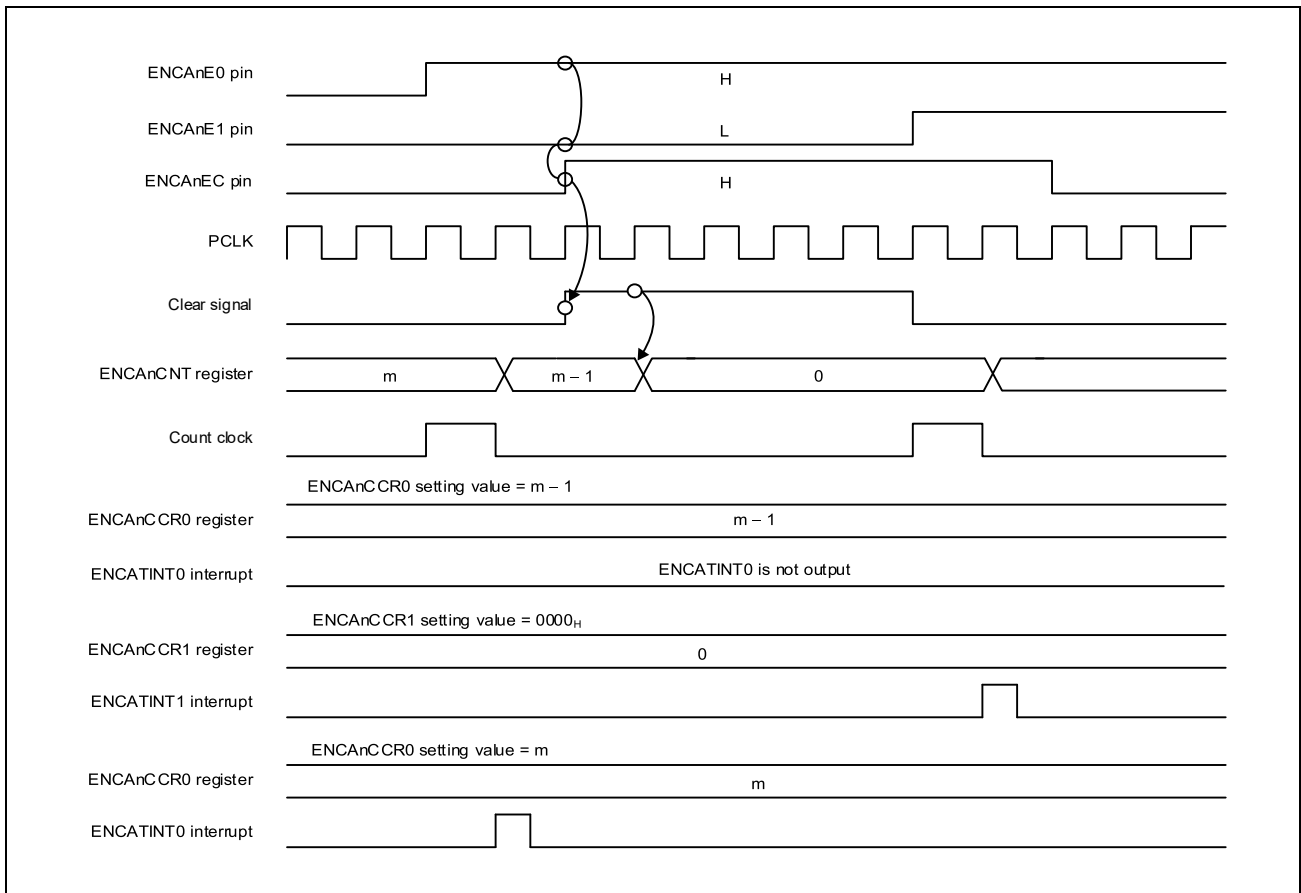


Figure 32.33 Clearing Timing for when the Timing of the ENCA_nEC Input is Later than that of the ENCA_nE1 Input during Down-count

32.6.19 Capture Operation Performed upon Clearing by ENCA_nEC when ENCA_nSCE = 0

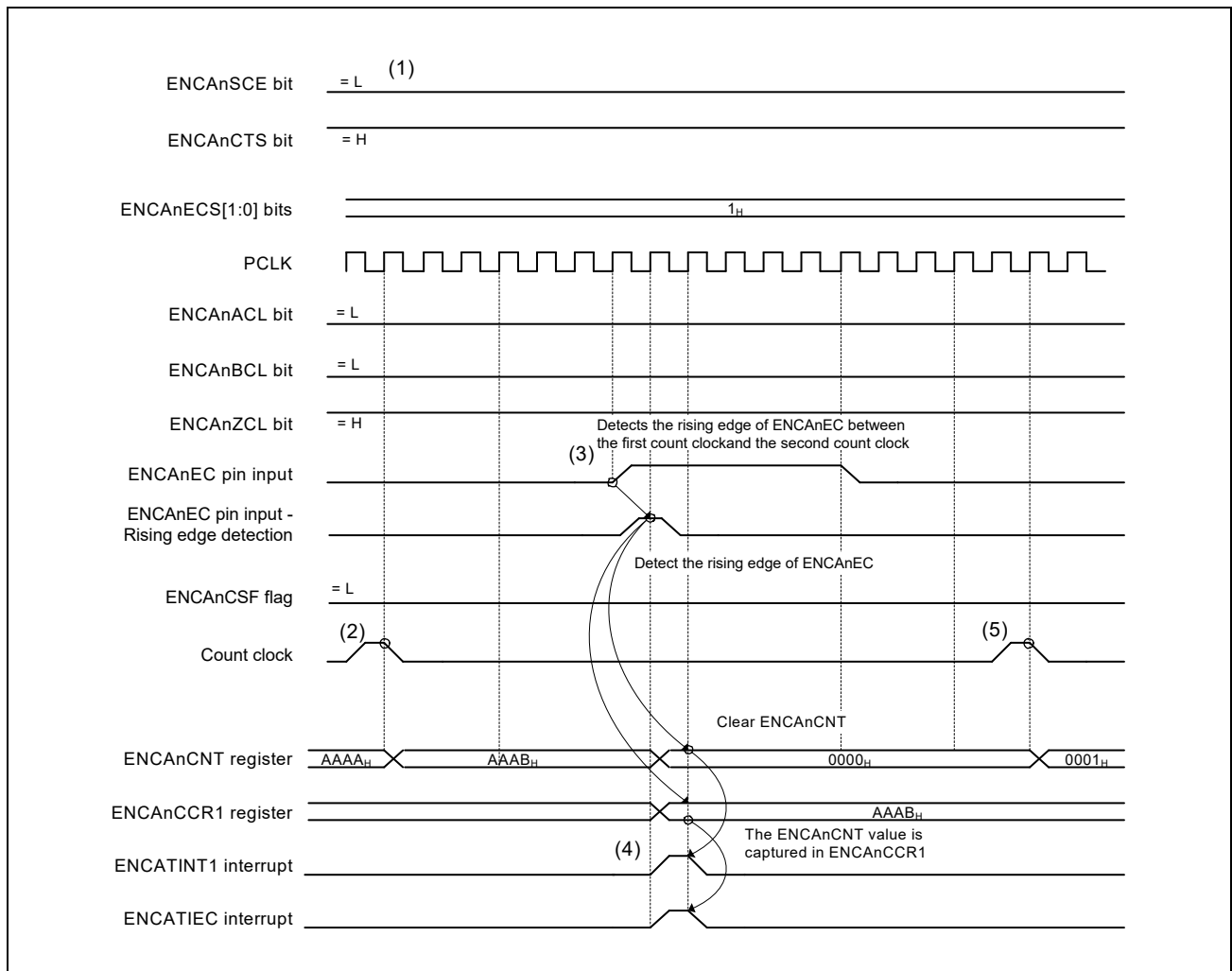


Figure 32.34 Capture Operation Performed upon Clearing by ENCA_nEC when ENCA_nSCE = 0

- (1) The values are set as follows: ENCA_nSCE = 0, ENCA_nCTS = 1, and ENCA_nECS[1:0] = 01_B.
- (2) An up-count is performed.
- (3) The rising edge of the ENCA_nEC input is detected, and the ENCA_nCNT value (AAAB_H) is captured in the ENCA_nCCR1 register. Concurrently, clear operation by ENCA_nEC is performed, and ENCA_nCNT is cleared to 0000_H.
- (4) A capture interrupt 1 (ENCA_nTINT1) to the ENCA_nCCR1 register and an encoder clear interrupt (ENCA_nTIEC) by ENCA_nEC are output.
- (5) After the count value is cleared, an up-count is performed, and the count value changes to 0001_H.

Section 33 Peripheral Interconnection (PIC)

33.1 Features of PIC

This section contains a generic description of the peripheral interconnection (PIC).

The first part of this section describes specific properties of this peripheral, such as the number of units, and register base addresses. The remainder of the section describes the functions and registers of the PIC (PIC1, PIC2).

33.1.1 Number of Units

This microcontroller has the following number of PIC units.

Table 33.1 Number of Units (PIC1)

Product Name	RH850/E2x-FCC1 Series
Number of Units	1
Name	PIC1

Table 33.2 Number of Units (PIC2)

Product Name	RH850/E2x-FCC1 Series
Number of Units	3
Name	PIC2

33.1.2 Register Base Addresses

PIC base addresses are listed in the following table.

PIC register addresses are given as offsets from the base addresses in general.

Table 33.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<PIC1_base>	FFDD 0000 _H	Peripheral Group 2H
<PIC20_base>	FFDD 1000 _H	Peripheral Group 2H
<PIC21_base>	FFDD 1800 _H	Peripheral Group 2H
<PIC22_base>	FFDD 2000 _H	Peripheral Group 2H

33.1.3 Clock Supply

PIC clock supplies are listed in the following table.

Table 33.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
PIC1	CLK_HSB	Peripheral high speed
PIC2	CLK_HSB	Peripheral high speed

33.1.4 Reset Sources

PIC reset sources are listed in the following table. PIC is initialized by these reset sources.

Table 33.5 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
PIC1	All registers	✓	✓	✓	✓	—	—
PIC2	All registers	✓	✓	✓	✓	—	—

33.1.5 External Input/Output Signals

External input/output signals of PIC are listed in the following table.

Table 33.6 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
ADTRG0	ADCH0 conversion startup trigger input	S_INEXT0Z
ADTRG1	ADCH1 conversion startup trigger input	S_INEXT1Z
ADTRG2	ADCH2 conversion startup trigger input	S_INEXT2Z
ADTRG3	ADCH3 conversion startup trigger input	S_INEXT3Z
DSADTRG00	DSADC00 conversion trigger input	D_INEXTZ0
DSADTRG10	DSADC10 conversion trigger input	D_INEXTZ1
DSADTRG20	DSADC20 conversion trigger input	D_INEXTZ2
DSADTRG12	DSADC12 conversion trigger input	D_INEXTZ3
DSADTRG13	DSADC13 conversion trigger input	D_INEXTZ4
DSADTRG11	DSADC11 conversion trigger input	D_INEXTZ5
CADTRG00	CADC00 conversion trigger input	D_INEXTZ10
ENCA0E0	ENCA0 encoder input (Phase A)	ENCA0E0
ENCA0E1	ENCA0 encoder input (Phase B)	ENCA0E1
ENCA0EC	ENCA0 encoder input (Phase Z)	ENCA0EC
ENCA1E0	ENCA1 encoder input (Phase A)	ENCA1E0
ENCA1E1	ENCA1 encoder input (Phase B)	ENCA1E1
ENCA1EC	ENCA1 encoder input (Phase Z)	ENCA1EC

33.2 Timer Synchronization Function (PIC1)

33.2.1 Overview

33.2.1.1 Functional Overview

The peripheral interconnection (PIC1) handles synchronous start of multiple OSTM timers.

The overview of PIC1 specification is shown in **Table 33.7**.

Table 33.7 PIC1 Specification Overview

Item	Description
Timer synchronization	Simultaneous start of any combination of timers (OSTMn).

33.2.1.2 Block Diagram

The following figure shows a block diagram of PIC1.

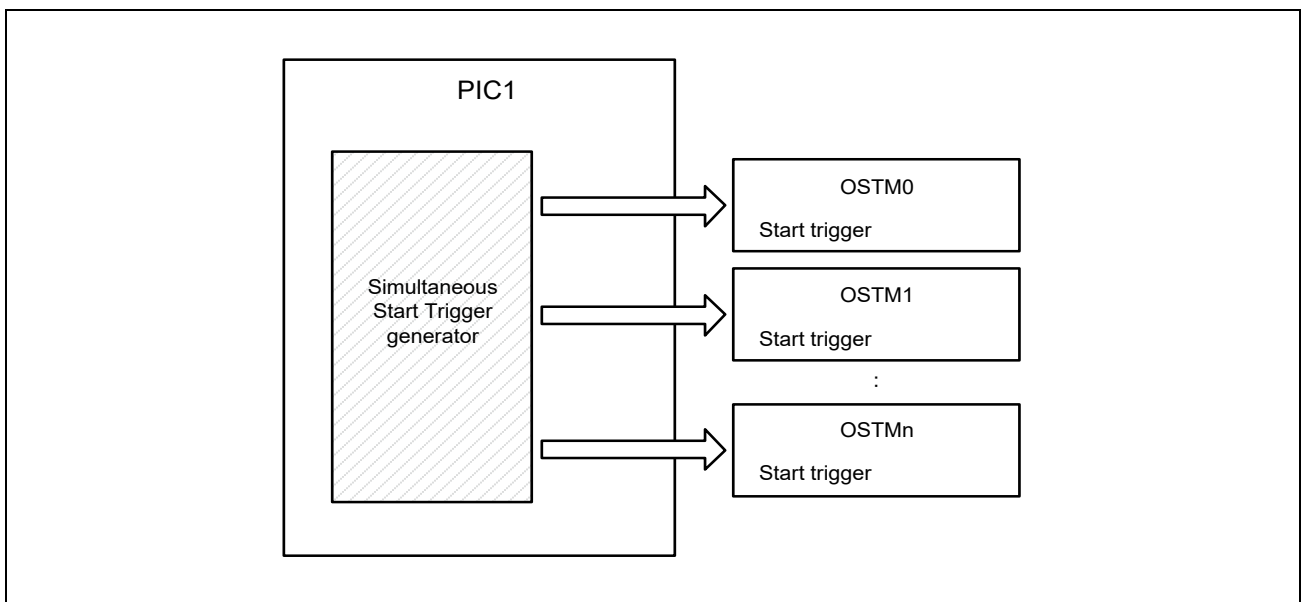


Figure 33.1 Block Diagram of PIC1

33.2.2 Registers

33.2.2.1 List of Registers

The list of PIC1 registers is shown in **Table 33.8**.

For information about restrictions on the individual registers and bits, see the register description in the following sections.

Table 33.8 List of PIC1 Registers

Register Name	Symbol	Address	Access	Access Protection
Simultaneous Start Trigger Control Register	PIC1SST	<PIC1_base> + 04 _H	8	—
Simultaneous Start Control Register 0	PIC1SSER0	<PIC1_base> + 10 _H	16	—

33.2.2.2 PIC1SST — Simultaneous Start Trigger Control Register

The PIC1SST register generates start triggers for timers for which simultaneous start is enabled.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SYNCTRG
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 33.9 PIC1SST Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When writing to these bits, write 0.
0	SYNCTRG* ¹	Generate start triggers for timers for which simultaneous start is enabled. 0: Invalid setting 1: Trigger simultaneous start (generates a trigger pulse of 1-PCLK-cycle to timers for which simultaneous start is enabled)

Note 1. When this bit is read, the result is always 0.

33.2.2.3 PIC1SSER0 — Simultaneous Start Control Register 0

The PIC1SSER0 register defines OSTMn timers (n = 0 to 2) for which simultaneous start is enabled.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PIC1SSER002	PIC1SSER001	PIC1SSER000
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 33.10 PIC1SSER0 Register Contents

Bit Position	Bit Name	Function
15 to 3	—	Reserved When writing to these bits, write 0.
2	PIC1SSER002	Enable simultaneous start for OSTM2. 0: Simultaneous start not enabled for OSTM2. 1: Simultaneous start is enabled for OSTM2.
1	PIC1SSER001	Enable simultaneous start for OSTM1. 0: Simultaneous start not enabled for OSTM1. 1: Simultaneous start is enabled for OSTM1.
0	PIC1SSER000	Enable simultaneous start for OSTM0. 0: Simultaneous start not enabled for OSTM0. 1: Simultaneous start is enabled for OSTM0.

33.2.3 Operation

33.2.3.1 Timer Synchronization

- (1) Configure timers (OSTMn). For more details about the configuration of the OSTMn timers see **Section 29, OS Timer (OSTM)**.
- (2) For the timers that should be started simultaneously, write 1 to their corresponding PIC1SSER00n bits of PIC1SSER0.
- (3) Write 1 to the SYNCTRG bit of PIC1SST0 to generate a start trigger.
- (4) Repeat (2) and (3) for timers that are not started yet to allow different timer groups to operate in a different timing.

33.3 Trigger Selection Function (PIC2)

33.3.1 Overview

33.3.1.1 Functional Overview

Peripheral interconnection (PIC2) handles configuration of triggers to ADCH, DSADC, ENCA, DFE, PSI5-S, GTM and CADC from various peripheral modules.

The overview of PIC2 specifications with ATU is shown in **Table 33.11**.

Table 33.11 Overview of PIC2 Specifications with ATU

Item	Description
ADC trigger selection	Selects sources from ATU for ADC
(selection from ATU timers, peripheral IPs, and above described sources)	Selects an AD trigger from the above selected sources and internal signals from peripheral IPs.
DSADC trigger selection	Selects an AD trigger for DSADC from ATU and external pins.
CADC trigger selection	Selects an AD trigger for CADC from ATU and external pins.

The overview of PIC2 specifications with GTM is shown in **Table 33.12**.

Table 33.12 Overview of PIC2 Specifications with GTM

Item	Description
ADC trigger selection	Selects sources from GTM for ADC
(selection from GTM timers, peripheral IPs, and above described sources)	Selects an AD trigger from the above selected sources and internal signals from peripheral IPs.
DSADC trigger selection	Selects an AD trigger for DSADC from GTM and external pins.
	Selects an AD read gate for DSADC from GTM.
CADC trigger selection	Selects an AD trigger for CADC from GTM and external pins.
	Selects an AD read gate for CADC from GTM.
ENCA trigger selection	Selects capture timing trigger from GTM for ENCA.
	Selects an ENCA capture timing trigger from the above selected sources and internal signals from peripheral IPs.
DFE trigger selection	Selects compare timer trigger and capture trigger, angle 1 deg trigger from GTM for DFE.
PSI5-S time stamp and sync signal selection	Selects time stamp and sync pulse signal from GTM for PSI5-S.
GTM timer input selection	
–PORT	Selects a timer input signal for GTM from GPIO.
–ADCH	Selects an AD scan end interrupt for GTM from ADCH.
–DSADC	Selects an AD conversion end timing interrupt for GTM from DSADC.
–CADC	Selects an AD conversion end timing interrupt for GTM from CADC.
–ENCA	Selects an ENCA count clock output and direction of rotation notice from encoder input (phases A/B/Z) and compare capture interrupt and phase Z edge detection for GTM from ENCA.
–DFE	Selects a DFE peak hold update notification for GTM from DFE.
ENCA encoder input selection	Selects an encoder input signal for ENCA from GPIO.

33.3.1.2 Block Diagrams

The following figure shows a block diagram of PIC2 configured with ATU.

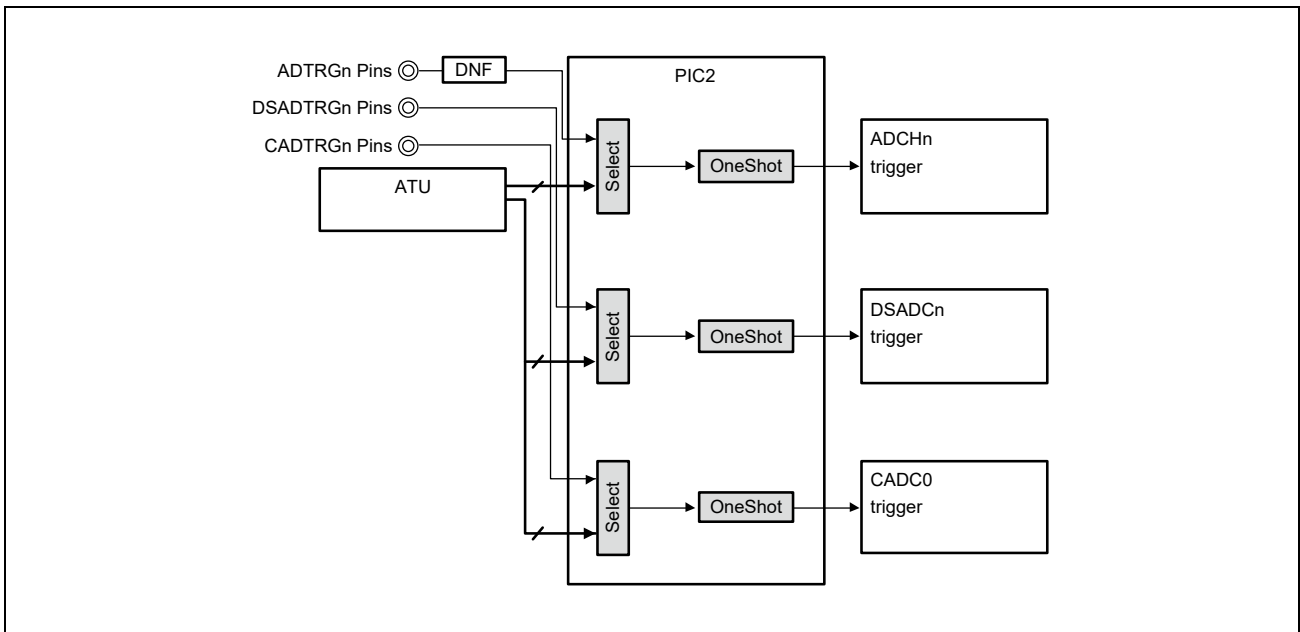


Figure 33.2 Block Diagram of PIC2 Configured with ATU

The following figure shows a block diagram of PIC2 configured with GTM.

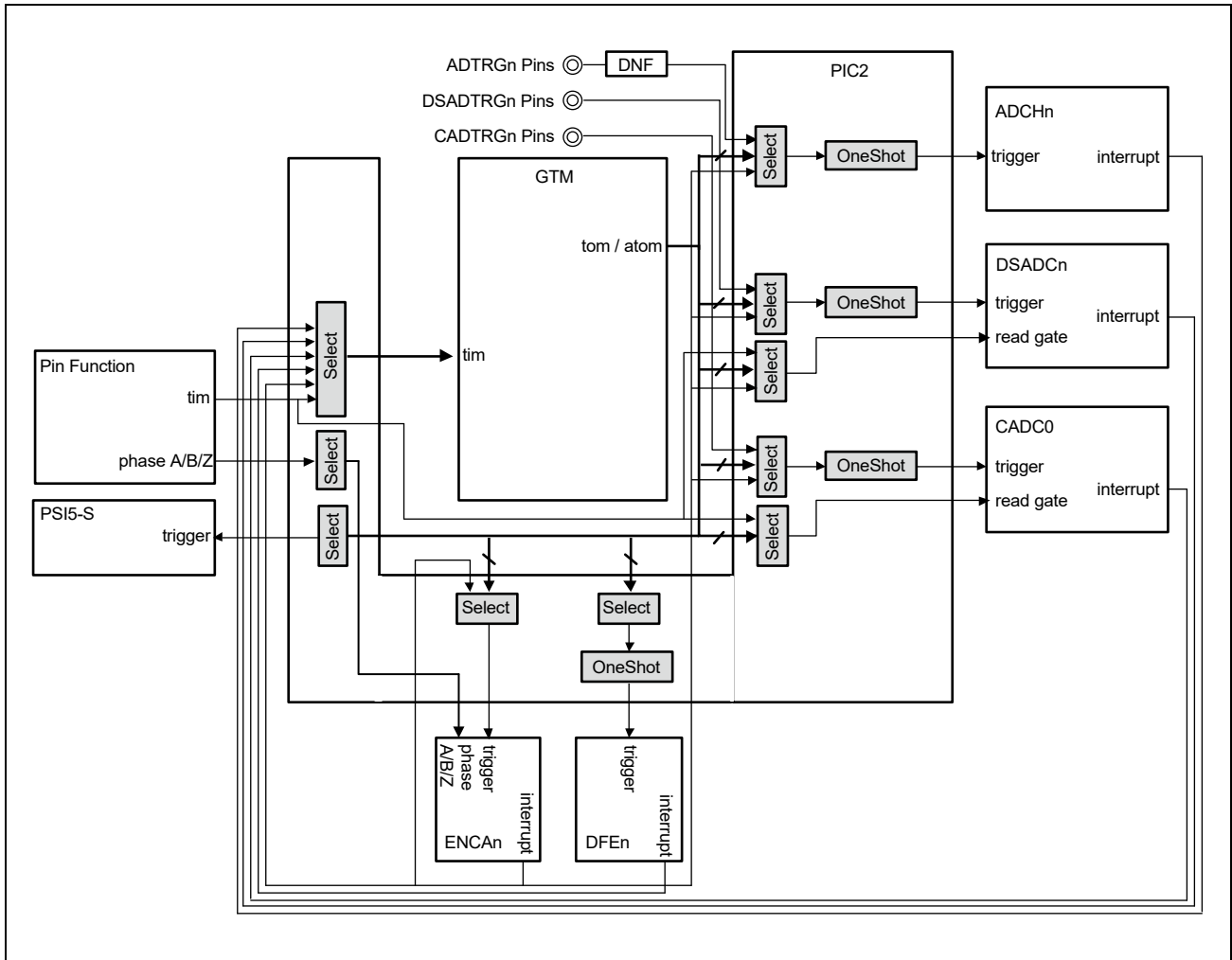


Figure 33.3 Block Diagram of PIC2 Configured with GTM

33.3.2 Registers

33.3.2.1 List of Registers

The list of PIC2 registers is shown in **Table 33.13**.

For information about restrictions on the individual registers and bits, see the register description in the following sections.

Table 33.13 List of PIC2 Registers (1/6)

Register Name	Symbol	Address	Access	Access Protection
ADC Trigger Selection Control Register				
AD Converter 0 Trigger Selection Control Register 0	PIC2ADCH0TSEL0	<PIC20_base> + 00 _H	16, 32	—
AD Converter 0 Trigger Selection Control Register 1	PIC2ADCH0TSEL1	<PIC20_base> + 04 _H	16, 32	—
AD Converter 0 Trigger Selection Control Register 2	PIC2ADCH0TSEL2	<PIC20_base> + 08 _H	16, 32	—
AD Converter 0 Trigger Selection Control Register 3	PIC2ADCH0TSEL3	<PIC20_base> + 0C _H	16, 32	—
AD Converter 0 Trigger Selection Control Register 4	PIC2ADCH0TSEL4	<PIC20_base> + 10 _H	16, 32	—
AD Converter 0 Trigger Edge Selection Control Register	PIC2ADCH0EDGSEL	<PIC20_base> + 1C _H	16	—
AD Converter 1 Trigger Selection Control Register 0	PIC2ADCH1TSEL0	<PIC20_base> + 20 _H	16, 32	—
AD Converter 1 Trigger Selection Control Register 1	PIC2ADCH1TSEL1	<PIC20_base> + 24 _H	16, 32	—
AD Converter 1 Trigger Selection Control Register 2	PIC2ADCH1TSEL2	<PIC20_base> + 28 _H	16, 32	—
AD Converter 1 Trigger Selection Control Register 3	PIC2ADCH1TSEL3	<PIC20_base> + 2C _H	16, 32	—
AD Converter 1 Trigger Selection Control Register 4	PIC2ADCH1TSEL4	<PIC20_base> + 30 _H	16, 32	—
AD Converter 1 Trigger Edge Selection Control Register	PIC2ADCH1EDGSEL	<PIC20_base> + 3C _H	16	—
AD Converter 2 Trigger Selection Control Register 0	PIC2ADCH2TSEL0	<PIC20_base> + 40 _H	16, 32	—
AD Converter 2 Trigger Selection Control Register 1	PIC2ADCH2TSEL1	<PIC20_base> + 44 _H	16, 32	—
AD Converter 2 Trigger Selection Control Register 2	PIC2ADCH2TSEL2	<PIC20_base> + 48 _H	16, 32	—
AD Converter 2 Trigger Selection Control Register 3	PIC2ADCH2TSEL3	<PIC20_base> + 4C _H	16, 32	—
AD Converter 2 Trigger Selection Control Register 4	PIC2ADCH2TSEL4	<PIC20_base> + 50 _H	16, 32	—
AD Converter 2 Trigger Edge Selection Control Register	PIC2ADCH2EDGSEL	<PIC20_base> + 5C _H	16	—
AD Converter 3 Trigger Selection Control Register 0	PIC2ADCH3TSEL0	<PIC20_base> + 60 _H	16, 32	—
AD Converter 3 Trigger Selection Control Register 1	PIC2ADCH3TSEL1	<PIC20_base> + 64 _H	16, 32	—
AD Converter 3 Trigger Selection Control Register 2	PIC2ADCH3TSEL2	<PIC20_base> + 68 _H	16, 32	—
AD Converter 3 Trigger Selection Control Register 3	PIC2ADCH3TSEL3	<PIC20_base> + 6C _H	16, 32	—
AD Converter 3 Trigger Selection Control Register 4	PIC2ADCH3TSEL4	<PIC20_base> + 70 _H	16, 32	—
AD Converter 3 Trigger Edge Selection Control Register	PIC2ADCH3EDGSEL	<PIC20_base> + 7C _H	16	—
AD Converter Trigger Output Control Register 500	PIC2ADTEN500	<PIC20_base> + 180 _H	16, 32	—
AD Converter Trigger Output Control Register 501	PIC2ADTEN501	<PIC20_base> + 184 _H	16, 32	—
AD Converter Trigger Output Control Register 502	PIC2ADTEN502	<PIC20_base> + 188 _H	16, 32	—
AD Converter Trigger Output Control Register 503	PIC2ADTEN503	<PIC20_base> + 18C _H	16, 32	—
AD Converter Trigger Output Control Register 504	PIC2ADTEN504	<PIC20_base> + 190 _H	16, 32	—
AD Converter Trigger Output Control Register 510	PIC2ADTEN510	<PIC20_base> + 1A0 _H	16, 32	—
AD Converter Trigger Output Control Register 511	PIC2ADTEN511	<PIC20_base> + 1A4 _H	16, 32	—
AD Converter Trigger Output Control Register 512	PIC2ADTEN512	<PIC20_base> + 1A8 _H	16, 32	—
AD Converter Trigger Output Control Register 513	PIC2ADTEN513	<PIC20_base> + 1AC _H	16, 32	—
AD Converter Trigger Output Control Register 514	PIC2ADTEN514	<PIC20_base> + 1B0 _H	16, 32	—
AD Converter Trigger Output Control Register 520	PIC2ADTEN520	<PIC20_base> + 1C0 _H	16, 32	—
AD Converter Trigger Output Control Register 521	PIC2ADTEN521	<PIC20_base> + 1C4 _H	16, 32	—

Table 33.13 List of PIC2 Registers (2/6)

Register Name	Symbol	Address	Access	Access Protection
AD Converter Trigger Output Control Register 522	PIC2ADTEN522	<PIC20_base> + 1C8 _H	16, 32	—
AD Converter Trigger Output Control Register 523	PIC2ADTEN523	<PIC20_base> + 1CC _H	16, 32	—
AD Converter Trigger Output Control Register 524	PIC2ADTEN524	<PIC20_base> + 1D0 _H	16, 32	—
AD Converter Trigger Output Control Register 530	PIC2ADTEN530	<PIC20_base> + 1E0 _H	16, 32	—
AD Converter Trigger Output Control Register 531	PIC2ADTEN531	<PIC20_base> + 1E4 _H	16, 32	—
AD Converter Trigger Output Control Register 532	PIC2ADTEN532	<PIC20_base> + 1E8 _H	16, 32	—
AD Converter Trigger Output Control Register 533	PIC2ADTEN533	<PIC20_base> + 1EC _H	16, 32	—
AD Converter Trigger Output Control Register 534	PIC2ADTEN534	<PIC20_base> + 1F0 _H	16, 32	—
AD Converter Trigger Output Control Register 600	PIC2ADTEN600	<PIC20_base> + 200 _H	32	—
AD Converter Trigger Output Control Register 601	PIC2ADTEN601	<PIC20_base> + 204 _H	32	—
AD Converter Trigger Output Control Register 602	PIC2ADTEN602	<PIC20_base> + 208 _H	32	—
AD Converter Trigger Output Control Register 603	PIC2ADTEN603	<PIC20_base> + 20C _H	32	—
AD Converter Trigger Output Control Register 604	PIC2ADTEN604	<PIC20_base> + 210 _H	32	—
AD Converter Trigger Output Control Register 610	PIC2ADTEN610	<PIC20_base> + 220 _H	32	—
AD Converter Trigger Output Control Register 611	PIC2ADTEN611	<PIC20_base> + 224 _H	32	—
AD Converter Trigger Output Control Register 612	PIC2ADTEN612	<PIC20_base> + 228 _H	32	—
AD Converter Trigger Output Control Register 613	PIC2ADTEN613	<PIC20_base> + 22C _H	32	—
AD Converter Trigger Output Control Register 614	PIC2ADTEN614	<PIC20_base> + 230 _H	32	—
AD Converter Trigger Output Control Register 620	PIC2ADTEN620	<PIC20_base> + 240 _H	32	—
AD Converter Trigger Output Control Register 621	PIC2ADTEN621	<PIC20_base> + 244 _H	32	—
AD Converter Trigger Output Control Register 622	PIC2ADTEN622	<PIC20_base> + 248 _H	32	—
AD Converter Trigger Output Control Register 623	PIC2ADTEN623	<PIC20_base> + 24C _H	32	—
AD Converter Trigger Output Control Register 624	PIC2ADTEN624	<PIC20_base> + 250 _H	32	—
AD Converter Trigger Output Control Register 630	PIC2ADTEN630	<PIC20_base> + 260 _H	32	—
AD Converter Trigger Output Control Register 631	PIC2ADTEN631	<PIC20_base> + 264 _H	32	—
AD Converter Trigger Output Control Register 632	PIC2ADTEN632	<PIC20_base> + 268 _H	32	—
AD Converter Trigger Output Control Register 633	PIC2ADTEN633	<PIC20_base> + 26C _H	32	—
AD Converter Trigger Output Control Register 634	PIC2ADTEN634	<PIC20_base> + 270 _H	32	—
AD Converter Trigger Output Control Register 700	PIC2ADTEN700	<PIC20_base> + 280 _H	32	—
AD Converter Trigger Output Control Register 701	PIC2ADTEN701	<PIC20_base> + 284 _H	32	—
AD Converter Trigger Output Control Register 702	PIC2ADTEN702	<PIC20_base> + 288 _H	32	—
AD Converter Trigger Output Control Register 703	PIC2ADTEN703	<PIC20_base> + 28C _H	32	—
AD Converter Trigger Output Control Register 704	PIC2ADTEN704	<PIC20_base> + 290 _H	32	—
AD Converter Trigger Output Control Register 710	PIC2ADTEN710	<PIC20_base> + 2A0 _H	32	—
AD Converter Trigger Output Control Register 711	PIC2ADTEN711	<PIC20_base> + 2A4 _H	32	—
AD Converter Trigger Output Control Register 712	PIC2ADTEN712	<PIC20_base> + 2A8 _H	32	—
AD Converter Trigger Output Control Register 713	PIC2ADTEN713	<PIC20_base> + 2AC _H	32	—
AD Converter Trigger Output Control Register 714	PIC2ADTEN714	<PIC20_base> + 2B0 _H	32	—
AD Converter Trigger Output Control Register 720	PIC2ADTEN720	<PIC20_base> + 2C0 _H	32	—
AD Converter Trigger Output Control Register 721	PIC2ADTEN721	<PIC20_base> + 2C4 _H	32	—
AD Converter Trigger Output Control Register 722	PIC2ADTEN722	<PIC20_base> + 2C8 _H	32	—
AD Converter Trigger Output Control Register 723	PIC2ADTEN723	<PIC20_base> + 2CC _H	32	—
AD Converter Trigger Output Control Register 724	PIC2ADTEN724	<PIC20_base> + 2D0 _H	32	—
AD Converter Trigger Output Control Register 730	PIC2ADTEN730	<PIC20_base> + 2E0 _H	32	—

Table 33.13 List of PIC2 Registers (3/6)

Register Name	Symbol	Address	Access	Access Protection
AD Converter Trigger Output Control Register 731	PIC2ADTEN731	<PIC20_base> + 2E4 _H	32	—
AD Converter Trigger Output Control Register 732	PIC2ADTEN732	<PIC20_base> + 2E8 _H	32	—
AD Converter Trigger Output Control Register 733	PIC2ADTEN733	<PIC20_base> + 2EC _H	32	—
AD Converter Trigger Output Control Register 734	PIC2ADTEN734	<PIC20_base> + 2F0 _H	32	—
AD Converter Trigger Output Control Register 800	PIC2ADTEN800	<PIC20_base> + 300 _H	32	—
AD Converter Trigger Output Control Register 801	PIC2ADTEN801	<PIC20_base> + 304 _H	32	—
AD Converter Trigger Output Control Register 802	PIC2ADTEN802	<PIC20_base> + 308 _H	32	—
AD Converter Trigger Output Control Register 803	PIC2ADTEN803	<PIC20_base> + 30C _H	32	—
AD Converter Trigger Output Control Register 804	PIC2ADTEN804	<PIC20_base> + 310 _H	32	—
AD Converter Trigger Output Control Register 810	PIC2ADTEN810	<PIC20_base> + 320 _H	32	—
AD Converter Trigger Output Control Register 811	PIC2ADTEN811	<PIC20_base> + 324 _H	32	—
AD Converter Trigger Output Control Register 812	PIC2ADTEN812	<PIC20_base> + 328 _H	32	—
AD Converter Trigger Output Control Register 813	PIC2ADTEN813	<PIC20_base> + 32C _H	32	—
AD Converter Trigger Output Control Register 814	PIC2ADTEN814	<PIC20_base> + 330 _H	32	—
AD Converter Trigger Output Control Register 820	PIC2ADTEN820	<PIC20_base> + 340 _H	32	—
AD Converter Trigger Output Control Register 821	PIC2ADTEN821	<PIC20_base> + 344 _H	32	—
AD Converter Trigger Output Control Register 822	PIC2ADTEN822	<PIC20_base> + 348 _H	32	—
AD Converter Trigger Output Control Register 823	PIC2ADTEN823	<PIC20_base> + 34C _H	32	—
AD Converter Trigger Output Control Register 824	PIC2ADTEN824	<PIC20_base> + 350 _H	32	—
AD Converter Trigger Output Control Register 830	PIC2ADTEN830	<PIC20_base> + 360 _H	32	—
AD Converter Trigger Output Control Register 831	PIC2ADTEN831	<PIC20_base> + 364 _H	32	—
AD Converter Trigger Output Control Register 832	PIC2ADTEN832	<PIC20_base> + 368 _H	32	—
AD Converter Trigger Output Control Register 833	PIC2ADTEN833	<PIC20_base> + 36C _H	32	—
AD Converter Trigger Output Control Register 834	PIC2ADTEN834	<PIC20_base> + 370 _H	32	—
DSADC Trigger Selection Control Register				
DSADC Start Trigger Output Control Register 00	PIC2DSADTEN000	<PIC20_base> + 400 _H	16	—
DSADC Start Trigger Output Control Register 01	PIC2DSADTEN001	<PIC20_base> + 404 _H	32	—
DSADC Start Trigger Output Control Register 02	PIC2DSADTEN002	<PIC20_base> + 408 _H	32	—
DSADC Start Trigger Output Control Register 03	PIC2DSADTEN003	<PIC20_base> + 40C _H	32	—
DSADC Stop Trigger Output Control Register 00	PIC2DSADTEN100	<PIC20_base> + 410 _H	16	—
DSADC Stop Trigger Output Control Register 01	PIC2DSADTEN101	<PIC20_base> + 414 _H	32	—
DSADC Stop Trigger Output Control Register 02	PIC2DSADTEN102	<PIC20_base> + 418 _H	32	—
DSADC Stop Trigger Output Control Register 03	PIC2DSADTEN103	<PIC20_base> + 41C _H	32	—
DSADC Start Trigger Output Control Register 10	PIC2DSADTEN010	<PIC20_base> + 420 _H	16	—
DSADC Start Trigger Output Control Register 11	PIC2DSADTEN011	<PIC20_base> + 424 _H	32	—
DSADC Start Trigger Output Control Register 12	PIC2DSADTEN012	<PIC20_base> + 428 _H	32	—
DSADC Start Trigger Output Control Register 13	PIC2DSADTEN013	<PIC20_base> + 42C _H	32	—
DSADC Stop Trigger Output Control Register 10	PIC2DSADTEN110	<PIC20_base> + 430 _H	16	—
DSADC Stop Trigger Output Control Register 11	PIC2DSADTEN111	<PIC20_base> + 434 _H	32	—
DSADC Stop Trigger Output Control Register 12	PIC2DSADTEN112	<PIC20_base> + 438 _H	32	—
DSADC Stop Trigger Output Control Register 13	PIC2DSADTEN113	<PIC20_base> + 43C _H	32	—
DSADC Start Trigger Output Control Register 20	PIC2DSADTEN020	<PIC20_base> + 440 _H	16	—
DSADC Start Trigger Output Control Register 21	PIC2DSADTEN021	<PIC20_base> + 444 _H	32	—
DSADC Start Trigger Output Control Register 22	PIC2DSADTEN022	<PIC20_base> + 448 _H	32	—

Table 33.13 List of PIC2 Registers (4/6)

Register Name	Symbol	Address	Access	Access Protection
DSADC Start Trigger Output Control Register 23	PIC2DSADTEN023	<PIC20_base> + 44C _H	32	—
DSADC Stop Trigger Output Control Register 20	PIC2DSADTEN120	<PIC20_base> + 450 _H	16	—
DSADC Stop Trigger Output Control Register 21	PIC2DSADTEN121	<PIC20_base> + 454 _H	32	—
DSADC Stop Trigger Output Control Register 22	PIC2DSADTEN122	<PIC20_base> + 458 _H	32	—
DSADC Stop Trigger Output Control Register 23	PIC2DSADTEN123	<PIC20_base> + 45C _H	32	—
DSADC Start Trigger Output Control Register 30	PIC2DSADTEN030	<PIC20_base> + 460 _H	16	—
DSADC Start Trigger Output Control Register 31	PIC2DSADTEN031	<PIC20_base> + 464 _H	32	—
DSADC Start Trigger Output Control Register 32	PIC2DSADTEN032	<PIC20_base> + 468 _H	32	—
DSADC Start Trigger Output Control Register 33	PIC2DSADTEN033	<PIC20_base> + 46C _H	32	—
DSADC Stop Trigger Output Control Register 30	PIC2DSADTEN130	<PIC20_base> + 470 _H	16	—
DSADC Stop Trigger Output Control Register 31	PIC2DSADTEN131	<PIC20_base> + 474 _H	32	—
DSADC Stop Trigger Output Control Register 32	PIC2DSADTEN132	<PIC20_base> + 478 _H	32	—
DSADC Stop Trigger Output Control Register 33	PIC2DSADTEN133	<PIC20_base> + 47C _H	32	—
DSADC Start Trigger Output Control Register 40	PIC2DSADTEN040	<PIC20_base> + 480 _H	16	—
DSADC Start Trigger Output Control Register 41	PIC2DSADTEN041	<PIC20_base> + 484 _H	32	—
DSADC Start Trigger Output Control Register 42	PIC2DSADTEN042	<PIC20_base> + 488 _H	32	—
DSADC Start Trigger Output Control Register 43	PIC2DSADTEN043	<PIC20_base> + 48C _H	32	—
DSADC Stop Trigger Output Control Register 40	PIC2DSADTEN140	<PIC20_base> + 490 _H	16	—
DSADC Stop Trigger Output Control Register 41	PIC2DSADTEN141	<PIC20_base> + 494 _H	32	—
DSADC Stop Trigger Output Control Register 42	PIC2DSADTEN142	<PIC20_base> + 498 _H	32	—
DSADC Stop Trigger Output Control Register 43	PIC2DSADTEN143	<PIC20_base> + 49C _H	32	—
DSADC Start Trigger Output Control Register 50	PIC2DSADTEN050	<PIC20_base> + 4A0 _H	16	—
DSADC Start Trigger Output Control Register 51	PIC2DSADTEN051	<PIC20_base> + 4A4 _H	32	—
DSADC Start Trigger Output Control Register 52	PIC2DSADTEN052	<PIC20_base> + 4A8 _H	32	—
DSADC Start Trigger Output Control Register 53	PIC2DSADTEN053	<PIC20_base> + 4AC _H	32	—
DSADC Stop Trigger Output Control Register 50	PIC2DSADTEN150	<PIC20_base> + 4B0 _H	16	—
DSADC Stop Trigger Output Control Register 51	PIC2DSADTEN151	<PIC20_base> + 4B4 _H	32	—
DSADC Stop Trigger Output Control Register 52	PIC2DSADTEN152	<PIC20_base> + 4B8 _H	32	—
DSADC Stop Trigger Output Control Register 53	PIC2DSADTEN153	<PIC20_base> + 4BC _H	32	—
DSADC Trigger Selection Control Register 0	PIC2DSADCATSEL0	<PIC20_base> + 600 _H	16, 32	—
DSADC Trigger Selection Control Register 1	PIC2DSADCATSEL1	<PIC20_base> + 604 _H	16, 32	—
Encoder Timer Trigger Selection Control Register				
Encoder Timer 0 Trigger Selection Control Register 0	PIC2ENCA0TSEL0	<PIC21_base> + 00 _H	16, 32	—
Encoder Timer 0 Trigger Selection Control Register 1	PIC2ENCA0TSEL1	<PIC21_base> + 04 _H	16, 32	—
Encoder Timer 1 Trigger Selection Control Register 0	PIC2ENCA1TSEL0	<PIC21_base> + 20 _H	16, 32	—
Encoder Timer 1 Trigger Selection Control Register 1	PIC2ENCA1TSEL1	<PIC21_base> + 24 _H	16, 32	—
CADC Trigger Selection Control Register				
CADC Start Trigger Output Control Register 00	PIC2CADTEN000	<PIC21_base> + 400 _H	16	—
CADC Start Trigger Output Control Register 01	PIC2CADTEN001	<PIC21_base> + 404 _H	32	—
CADC Start Trigger Output Control Register 02	PIC2CADTEN002	<PIC21_base> + 408 _H	32	—
CADC Start Trigger Output Control Register 03	PIC2CADTEN003	<PIC21_base> + 40C _H	32	—
CADC Stop Trigger Output Control Register 00	PIC2CADTEN100	<PIC21_base> + 410 _H	16	—
CADC Stop Trigger Output Control Register 01	PIC2CADTEN101	<PIC21_base> + 414 _H	32	—
CADC Stop Trigger Output Control Register 02	PIC2CADTEN102	<PIC21_base> + 418 _H	32	—

Table 33.13 List of PIC2 Registers (5/6)

Register Name	Symbol	Address	Access	Access Protection
CADC Stop Trigger Output Control Register 03	PIC2CADTEN103	<PIC21_base> + 41C _H	32	—
CADC Trigger Selection Control Register 0	PIC2CADCATSELO	<PIC21_base> + 600 _H	16, 32	—
CADC Trigger Selection Control Register 1	PIC2CADCATSEL1	<PIC21_base> + 604 _H	16, 32	—
AD Converter Trigger Output Configuration Register				
AD Converter Trigger Output Configuration Register 0	PIC2ADTCFG0	<PIC22_base> + 00 _H	16, 32	—
AD Converter Trigger Output Configuration Register 1	PIC2ADTCFG1	<PIC22_base> + 04 _H	16, 32	—
AD Converter Trigger Output Configuration Register 2	PIC2ADTCFG2	<PIC22_base> + 08 _H	16, 32	—
AD Converter Trigger Output Configuration Register 3	PIC2ADTCFG3	<PIC22_base> + 0C _H	16, 32	—
AD Converter Trigger Output Configuration Register 4	PIC2ADTCFG4	<PIC22_base> + 10 _H	16, 32	—
DSADC Start/Stop Trigger Output Configuration Register				
DSADC Start/Stop Trigger Output Configuration Register 0	PIC2DSADTCFG0	<PIC22_base> + 20 _H	16, 32	—
DSADC Start/Stop Trigger Output Configuration Register 1	PIC2DSADTCFG1	<PIC22_base> + 24 _H	16, 32	—
DSADC Start/Stop Trigger Output Configuration Register 2	PIC2DSADTCFG2	<PIC22_base> + 28 _H	16, 32	—
CADC Start/Stop Trigger Output Configuration Register				
CADC Start/Stop Trigger Output Configuration Register 0	PIC2CADTCFG0	<PIC22_base> + 40 _H	16, 32	—
DFE Timer Trigger Edge Selection Control Register				
DFE Timer Trigger Edge Selection Control Register 0	PIC2DFETEDGSELO	<PIC22_base> + 50 _H	16, 32	—
DFE Timer Trigger Edge Selection Control Register 1	PIC2DFETEDGSEL1	<PIC22_base> + 54 _H	16, 32	—
DFE Timer Trigger Edge Selection Control Register 2	PIC2DFETEDGSEL2	<PIC22_base> + 58 _H	16, 32	—
DSADC Read Gate Trigger Selection Control Register				
DSADC Read Gate Trigger Selection Control Register 0	PIC2DSADTSEN0	<PIC22_base> + 60 _H	16, 32	—
DSADC Read Gate Trigger Selection Control Register 1	PIC2DSADTSEN1	<PIC22_base> + 64 _H	16, 32	—
CADC Read Gate Trigger Selection Control Register 0	PIC2CADTSEN0	<PIC22_base> + 6C _H	16, 32	—
DFE Trigger Selection Control Register				
DFE Timer Trigger Selection Control Register 0	PIC2DFETSEN0	<PIC22_base> + 70 _H	16, 32	—
DFE Timer Trigger Selection Control Register 1	PIC2DFETSEN1	<PIC22_base> + 74 _H	16, 32	—
DFE Timer Trigger Selection Control Register 2	PIC2DFETSEN2	<PIC22_base> + 78 _H	16, 32	—
DFE Timer Trigger Selection Control Register 3	PIC2DFETSEN3	<PIC22_base> + 7C _H	16, 32	—
DFE Timer Trigger Selection Control Register 4	PIC2DFETSEN4	<PIC22_base> + 80 _H	16, 32	—
DFE Timer Trigger Selection Control Register 5	PIC2DFETSEN5	<PIC22_base> + 84 _H	16, 32	—
DFE Timer Trigger Selection Control Register 6	PIC2DFETSEN6	<PIC22_base> + 88 _H	16, 32	—
DFE Timer Trigger Selection Control Register 7	PIC2DFETSEN7	<PIC22_base> + 8C _H	16, 32	—
DFE Timer Trigger Selection Control Register 8	PIC2DFETSEN8	<PIC22_base> + 90 _H	16, 32	—
DFE Timer Trigger Selection Control Register 9	PIC2DFETSEN9	<PIC22_base> + 94 _H	16, 32	—
DFE Timer Trigger Selection Control Register 10	PIC2DFETSEN10	<PIC22_base> + 98 _H	16, 32	—
DFE Timer Trigger Selection Control Register 11	PIC2DFETSEN11	<PIC22_base> + 9C _H	16, 32	—
PSI5S Sync Output Control Register				
PSI5S Sync Output Control Register 0	PIC2PSI5SEN0	<PIC22_base> + A0 _H	16, 32	—
PSI5S Sync Output Control Register 1	PIC2PSI5SEN1	<PIC22_base> + A4 _H	16, 32	—
PSI5S Sync Output Control Register 2	PIC2PSI5SEN2	<PIC22_base> + A8 _H	16, 32	—

Table 33.13 List of PIC2 Registers (6/6)

Register Name	Symbol	Address	Access	Access Protection
PSI5S Sync Output Control Register 3	PIC2PSI5SEN3	<PIC22_base> + AC _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register				
GTM Timer Input Module (TIM) Source Select Register 0	PIC2GTMINEN0	<PIC22_base> + B0 _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register 1	PIC2GTMINEN1	<PIC22_base> + B4 _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register 2	PIC2GTMINEN2	<PIC22_base> + B8 _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register 3	PIC2GTMINEN3	<PIC22_base> + BC _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register 4	PIC2GTMINEN4	<PIC22_base> + C0 _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register 5	PIC2GTMINEN5	<PIC22_base> + C4 _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register 6	PIC2GTMINEN6	<PIC22_base> + C8 _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register 7	PIC2GTMINEN7	<PIC22_base> + CC _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register 8	PIC2GTMINEN8	<PIC22_base> + D0 _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register 9	PIC2GTMINEN9	<PIC22_base> + D4 _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register 10	PIC2GTMINEN10	<PIC22_base> + D8 _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register 11	PIC2GTMINEN11	<PIC22_base> + DC _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register 12	PIC2GTMINEN12	<PIC22_base> + E0 _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register 13	PIC2GTMINEN13	<PIC22_base> + E4 _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register 14	PIC2GTMINEN14	<PIC22_base> + E8 _H	16, 32	—
GTM Timer Input Module (TIM) Source Select Register 15	PIC2GTMINEN15	<PIC22_base> + EC _H	16, 32	—
Encoder Timer Trigger Output Configuration Register 0	PIC2ENCATCFG0	<PIC22_base> + F0 _H	16, 32	—
Encoder Timer Trigger Output Configuration Register 1	PIC2ENCATCFG1	<PIC22_base> + F4 _H	16, 32	—
Encoder Input Selection Control Register 0	PIC2ENCAISEN0	<PIC22_base> + F8 _H	8	—

33.3.2.2 PIC2ADCHnTSELj — AD Converter n Trigger Selection Control Register j (n = 0 to 3^{*}; j = 0 to 4)

The PIC2ADCHnTSELj register selects a trigger for ADCHn scan group j (n = 0 to 3^{*}; j = 0 to 4).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2AD CHnTS ELj15	—	—	—	—	—	—	PIC2AD CHnTS ELj08	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Table 33.14 PIC2ADCHnTSELj Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing to these bits, write 0.
15	PIC2ADCHnTSELj15	Selects the trigger selected by the “PIC2ADTEN5nj PIC2ADTEN6nj PIC2ADTEN7nj PIC2ADTEN8nj” register as a trigger source of ADCHn scan group i. 0: Does not select the trigger selected by register “PIC2ADTEN5nj,6nj,7nj,8nj”. 1: Selects the trigger selected by register “PIC2ADTEN5nj,6nj,7nj,8nj”.
14 to 9	—	Reserved When writing to these bits, write 0.
8	PIC2ADCHnTSELj08	Selects the ADTRGn pin as a trigger source of ADCHn scan group j. 0: The ADTRGn pin is not selected. 1: The ADTRGn pin is selected.
7 to 0	—	Reserved When writing to these bits, write 0.

Note 1. ADCH1 and ADCH3 are implemented only on E2x-FCC1 and E2M(373 pins), they are not supported for E2M(292 pins).

33.3.2.3 PIC2ADCHnEDGSEL — AD Converter n*1 Trigger Edge Selection Control Register (n = 0 to 3*1)

The PIC2ADCHnEDGSEL register selects an effective edge for the one-shot pulse generation circuit which generates an ADCH trigger.

The ADCH external pin trigger is input in negative logic, but it is converted to positive logic for trigger source selection. Since edge detection is made for a trigger source after selection, note that the definition of an edge is reversed for an ADCH external pin signal. (Setting 00 enables selection of a falling edge of ADCH external pin triggers ADTRG0, ADTRG1, ADTRG2, and ADTRG3.)

Value after reset: 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIC2ADCHnEDGSEL98[9:8]	PIC2ADCHnEDGSEL76[7:6]	PIC2ADCHnEDGSEL54[5:4]	PIC2ADCHnEDGSEL32[3:2]	PIC2ADCHnEDGSEL10[1:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.15 PIC2ADCHnEDGSEL Register Contents

Bit Position	Bit Name	Function
15 to 10	—	Reserved When writing to these bits, write 0.
9, 8	PIC2ADCHnEDGSEL98[9:8]	Select an effective edge of ADCHn scan group 4. 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited)
7, 6	PIC2ADCHnEDGSEL76[7:6]	Select an effective edge of ADCHn scan group 3. 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited)
5, 4	PIC2ADCHnEDGSEL54[5:4]	Select an effective edge of ADCHn scan group 2. 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited)
3, 2	PIC2ADCHnEDGSEL32[3:2]	Select an effective edge of ADCHn scan group 1. 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited)
1, 0	PIC2ADCHnEDGSEL10[1:0]	Select an effective edge of ADCHn scan group 0. 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited)

Note 1. ADCH1 and ADCH3 are implemented only on E2x-FCC1 and E2M(373 pins), they are not supported for E2M(292 pins).

33.3.2.4 PIC2ADTEN5nj — AD Converter Trigger Output Control Register 5nj (n = 0 to 3*1; j = 0 to 4)

The PIC2ADTEN5nj register enables a trigger source from ATU timer C compare match, ATU timer G compare match, and ATU DMA request switch function to be selected as an ADCH trigger (n = 0 to 3*1; j = 0 to 4).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2ADTEN5nj31	PIC2ADTEN5nj30	PIC2ADTEN5nj29	PIC2ADTEN5nj28	PIC2ADTEN5nj27	PIC2ADTEN5nj26	PIC2ADTEN5nj25	PIC2ADTEN5nj24	PIC2ADTEN5nj23	PIC2ADTEN5nj22	PIC2ADTEN5nj21	PIC2ADTEN5nj20	PIC2ADTEN5nj19	PIC2ADTEN5nj18	PIC2ADTEN5nj17	PIC2ADTEN5nj16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2ADTEN5nj15	PIC2ADTEN5nj14	PIC2ADTEN5nj13	PIC2ADTEN5nj12	PIC2ADTEN5nj11	PIC2ADTEN5nj10	PIC2ADTEN5nj9	PIC2ADTEN5nj8	—	—	—	—	—	—	—	PIC2ADTEN5nj0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 33.16 PIC2ADTEN5nj Register Contents (1/3)

Bit Position	Bit Name	Function
31	PIC2ADTEN5nj31	GRC73 input capture / compare match interrupt (ATU5 timer C subblock 7 channel 3 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
30	PIC2ADTEN5nj30	GRC72 input capture / compare match interrupt (ATU5 timer C subblock 7 channel 2 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
29	PIC2ADTEN5nj29	GRC71 input capture / compare match interrupt (ATU5 timer C subblock 7 channel 1 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
28	PIC2ADTEN5nj28	GRC70 input capture / compare match interrupt (ATU5 timer C subblock 7 channel 0 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
27	PIC2ADTEN5nj27	GRC63 input capture / compare match interrupt (ATU5 timer C subblock 6 channel 3 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.

Note 1. ADCH1 and ADCH3 are implemented only on E2x-FCC1 and E2M(373 pins), they are not supported for E2M(292 pins).

Table 33.16 PIC2ADTEN5nj Register Contents (2/3)

Bit Position	Bit Name	Function
26	PIC2ADTEN5nj26	GRC62 input capture / compare match interrupt (ATU5 timer C subblock 6 channel 2 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
25	PIC2ADTEN5nj25	GRC61 input capture / compare match interrupt (ATU5 timer C subblock 6 channel 1 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
24	PIC2ADTEN5nj24	GRC60 input capture / compare match interrupt (ATU5 timer C subblock 6 channel 0 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
23	PIC2ADTEN5nj23	GRC43 input capture / compare match interrupt (ATU5 timer C subblock 4 channel 3 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
22	PIC2ADTEN5nj22	GRC42 input capture / compare match interrupt (ATU5 timer C subblock 4 channel 2 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
21	PIC2ADTEN5nj21	GRC41 input capture / compare match interrupt (ATU5 timer C subblock 4 channel 1 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
20	PIC2ADTEN5nj20	GRC40 input capture / compare match interrupt (ATU5 timer C subblock 4 channel 0 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
19	PIC2ADTEN5nj19	GRC03 input capture / compare match interrupt (ATU5 timer C subblock 0 channel 3 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
18	PIC2ADTEN5nj18	GRC02 input capture / compare match interrupt (ATU5 timer C subblock 0 channel 2 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
17	PIC2ADTEN5nj17	GRC01 input capture / compare match interrupt (ATU5 timer C subblock 0 channel 1 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
16	PIC2ADTEN5nj16	GRC00 input capture / compare match interrupt (ATU5 timer C subblock 0 channel 0 input capture / compare match interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
15	PIC2ADTEN5nj15	OCRG7 compare match interrupt (ATU5 timer G subblock 7) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
14	PIC2ADTEN5nj14	OCRG6 compare match interrupt (ATU5 timer G subblock 6) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.

Table 33.16 PIC2ADTEN5nj Register Contents (3/3)

Bit Position	Bit Name	Function
13	PIC2ADTEN5nj13	OCRG5 compare match interrupt (ATU5 timer G subblock 5) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
12	PIC2ADTEN5nj12	OCRG4 compare match interrupt (ATU5 timer G subblock 4) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
11	PIC2ADTEN5nj11	OCRG3 compare match interrupt (ATU5 timer G subblock 3) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
10	PIC2ADTEN5nj10	OCRG2 compare match interrupt (ATU5 timer G subblock 2) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
9	PIC2ADTEN5nj9	OCRG1 compare match interrupt (ATU5 timer G subblock 1) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
8	PIC2ADTEN5nj8	OCRG0 compare match interrupt (ATU5 timer G subblock 0) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
7 to 1	—	Reserved When writing to these bits, write 0.
0	PIC2ADTEN5nj0	Output signal from the ADCH trigger selection circuit for ATU DMA / AD request auto-switching 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.

33.3.2.5 PIC2ADTEN6nj — AD Converter Trigger Output Control Register 6nj (n = 0 to 3*1; j = 0 to 4)

The PIC2ADTEN6nj register enables a trigger source from ATU timer D compare match A to be selected as an ADCH trigger (n = 0 to 3*1; j = 0 to 4).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2AD TEN6nj 15	PIC2AD TEN6nj 14	PIC2AD TEN6nj 13	PIC2AD TEN6nj 12	PIC2AD TEN6nj 11	PIC2AD TEN6nj 10	PIC2AD TEN6nj 9	PIC2AD TEN6nj 8	PIC2AD TEN6nj 7	PIC2AD TEN6nj 6	PIC2AD TEN6nj 5	PIC2AD TEN6nj 4	PIC2AD TEN6nj 3	PIC2AD TEN6nj 2	PIC2AD TEN6nj 1	PIC2AD TEN6nj 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.17 PIC2ADTEN6nj Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing, always write 0.
15	PIC2ADTEN6nj15	OCR1D63 compare match interrupt (ATU5 timer D subblock 6 channel 3 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
14	PIC2ADTEN6nj14	OCR1D62 compare match interrupt (ATU5 timer D subblock 6 channel 2 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
13	PIC2ADTEN6nj13	OCR1D61 compare match interrupt (ATU5 timer D subblock 6 channel 1 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
12	PIC2ADTEN6nj12	OCR1D60 compare match interrupt (ATU5 timer D subblock 6 channel 0 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
11	PIC2ADTEN6nj11	OCR1D43 compare match interrupt (ATU5 timer D subblock 4 channel 3 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.

Note 1. ADCH1 and ADCH3 are implemented only on E2x-FCC1 and E2M(373 pins), they are not supported for E2M(292 pins).

Table 33.17 PIC2ADTEN6nj Register Contents (2/2)

Bit Position	Bit Name	Function
10	PIC2ADTEN6nj10	OCR1D42 compare match interrupt (ATU5 timer D subblock 4 channel 2 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
9	PIC2ADTEN6nj9	OCR1D41 compare match interrupt (ATU5 timer D subblock 4 channel 1 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
8	PIC2ADTEN6nj8	OCR1D40 compare match interrupt (ATU5 timer D subblock 4 channel 0 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
7	PIC2ADTEN6nj7	OCR1D23 compare match interrupt (ATU5 timer D subblock 2 channel 3 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
6	PIC2ADTEN6nj6	OCR1D22 compare match interrupt (ATU5 timer D subblock 2 channel 2 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
5	PIC2ADTEN6nj5	OCR1D21 compare match interrupt (ATU5 timer D subblock 2 channel 1 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
4	PIC2ADTEN6nj4	OCR1D20 compare match interrupt (ATU5 timer D subblock 2 channel 0 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
3	PIC2ADTEN6nj3	OCR1D03 compare match interrupt (ATU5 timer D subblock 0 channel 3 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
2	PIC2ADTEN6nj2	OCR1D02 compare match interrupt (ATU5 timer D subblock 0 channel 2 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
1	PIC2ADTEN6nj1	OCR1D01 compare match interrupt (ATU5 timer D subblock 0 channel 1 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
0	PIC2ADTEN6nj0	OCR1D00 compare match interrupt (ATU5 timer D subblock 0 channel 0 compare match A interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.

33.3.2.6 PIC2ADTEN7nj — AD Converter Trigger Output Control Register 7nj (n = 0 to 3*1; j = 0 to 4)

The PIC2ADTEN7nj register enables a trigger source from ATU timer D compare match B to be selected as an ADCH trigger (n = 0 to 3*1; j = 0 to 4).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2AD TEN7nj 15	PIC2AD TEN7nj 14	PIC2AD TEN7nj 13	PIC2AD TEN7nj 12	PIC2AD TEN7nj 11	PIC2AD TEN7nj 10	PIC2AD TEN7nj 9	PIC2AD TEN7nj 8	PIC2AD TEN7nj 7	PIC2AD TEN7nj 6	PIC2AD TEN7nj 5	PIC2AD TEN7nj 4	PIC2AD TEN7nj 3	PIC2AD TEN7nj 2	PIC2AD TEN7nj 1	PIC2AD TEN7nj 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.18 PIC2ADTEN7nj Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing to these bits, write 0.
15	PIC2ADTEN7nj15	OCR2D63 compare match interrupt (ATU5 timer D subblock 6 channel 3 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
14	PIC2ADTEN7nj14	OCR2D62 compare match interrupt (ATU5 timer D subblock 6 channel 2 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
13	PIC2ADTEN7nj13	OCR2D61 compare match interrupt (ATU5 timer D subblock 6 channel 1 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
12	PIC2ADTEN7nj12	OCR2D60 compare match interrupt (ATU5 timer D subblock 6 channel 0 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
11	PIC2ADTEN7nj11	OCR2D43 compare match interrupt (ATU5 timer D subblock 4 channel 3 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.

Note 1. ADCH1 and ADCH3 are implemented only on E2x-FCC1 and E2M(373 pins), they are not supported for E2M(292 pins).

Table 33.18 PIC2ADTEN7nj Register Contents (2/2)

Bit Position	Bit Name	Function
10	PIC2ADTEN7nj10	OCR2D42 compare match interrupt (ATU5 timer D subblock 4 channel 2 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
9	PIC2ADTEN7nj9	OCR2D41 compare match interrupt (ATU5 timer D subblock 4 channel 1 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
8	PIC2ADTEN7nj8	OCR2D40 compare match interrupt (ATU5 timer D subblock 4 channel 0 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
7	PIC2ADTEN7nj7	OCR2D23 compare match interrupt (ATU5 timer D subblock 2 channel 3 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
6	PIC2ADTEN7nj6	OCR2D22 compare match interrupt (ATU5 timer D subblock 2 channel 2 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
5	PIC2ADTEN7nj5	OCR2D21 compare match interrupt (ATU5 timer D subblock 2 channel 1 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
4	PIC2ADTEN7nj4	OCR2D20 compare match interrupt (ATU5 timer D subblock 2 channel 0 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
3	PIC2ADTEN7nj3	OCR2D03 compare match interrupt (ATU5 timer D subblock 0 channel 3 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
2	PIC2ADTEN7nj2	OCR2D02 compare match interrupt (ATU5 timer D subblock 0 channel 2 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
1	PIC2ADTEN7nj1	OCR2D01 compare match interrupt (ATU5 timer D subblock 0 channel 1 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
0	PIC2ADTEN7nj0	OCR2D00 compare match interrupt (ATU5 timer D subblock 0 channel 0 compare match B interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.

33.3.2.7 PIC2ADTEN8nj — AD Converter Trigger Output Control Register 8nj (n = 0 to 3^{*1}; j = 0 to 4)

The PIC2ADTEN8nj register enables a trigger source from GTM (by set output in A/D converter trigger output configuration register) and ENCA interrupts to be selected as an ADCH trigger (n = 0 to 3^{*1}; j = 0 to 4).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2ADTEN8nj31	PIC2ADTEN8nj30	PIC2ADTEN8nj29	PIC2ADTEN8nj28	PIC2ADTEN8nj27	PIC2ADTEN8nj26	—	—	—	—	—	—	PIC2ADTEN8nj19	PIC2ADTEN8nj18	PIC2ADTEN8nj17	PIC2ADTEN8nj16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2ADTEN8nj15	PIC2ADTEN8nj14	PIC2ADTEN8nj13	PIC2ADTEN8nj12	PIC2ADTEN8nj11	PIC2ADTEN8nj10	PIC2ADTEN8nj9	PIC2ADTEN8nj8	PIC2ADTEN8nj7	PIC2ADTEN8nj6	PIC2ADTEN8nj5	PIC2ADTEN8nj4	PIC2ADTEN8nj3	PIC2ADTEN8nj2	PIC2ADTEN8nj1	PIC2ADTEN8nj0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.19 PIC2ADTEN8nj Register Contents (1/3)

Bit Position	Bit Name	Function
31	PIC2ADTEN8nj31	INTENCA1IEC Clear interrupt signal by encoder. (ENCA1 ENCATIEC Clear interrupt signal by encoder) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
30	PIC2ADTEN8nj30	INTENCA1I1 Compare 1 match or Capture 1 interrupt. (ENCA1 ENCATINT1 Compare 1 match or Capture 1 interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
29	PIC2ADTEN8nj29	INTENCA1I0 Compare 0 match or Capture 0 interrupt. (ENCA1 ENCATINT0 Compare 0 match or Capture 0 interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
28	PIC2ADTEN8nj28	INTENCA0IEC Clear interrupt signal by encoder. (ENCA0 ENCATIEC Clear interrupt signal by encoder) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
27	PIC2ADTEN8nj27	INTENCA0I1 compare 1 match or Capture 1 interrupt. (ENCA0 ENCATINT1 Compare 1 match or Capture 1 interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
26	PIC2ADTEN8nj26	INTENCA0I0 compare 0 match or Capture 0 interrupt. (ENCA0 ENCATINT0 Compare 0 match or Capture 0 interrupt) 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.

Note 1. ADCH1 and ADCH3 are implemented only on E2x-FCC1 and E2M(373 pins), they are not supported for E2M(292 pins).

Table 33.19 PIC2ADTEN8nj Register Contents (2/3)

Bit Position	Bit Name	Function
25 to 20	—	Reserved When writing to these bits, write 0.
19	PIC2ADTEN8nj19	Output value of GTM timer output selected by PIC2ADTCFG4 bits 31 to 24. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
18	PIC2ADTEN8nj18	Output value of GTM timer output selected by PIC2ADTCFG4 bits 23 to 16. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
17	PIC2ADTEN8nj17	Output value of GTM timer output selected by PIC2ADTCFG4 bits 15 to 8. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
16	PIC2ADTEN8nj16	Output value of GTM timer output selected by PIC2ADTCFG4 bits 7 to 0. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
15	PIC2ADTEN8nj15	Output value of GTM timer output selected by PIC2ADTCFG3 bits 31 to 24. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
14	PIC2ADTEN8nj14	Output value of GTM timer output selected by PIC2ADTCFG3 bits 23 to 16. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
13	PIC2ADTEN8nj13	Output value of GTM timer output selected by PIC2ADTCFG3 bits 15 to 8. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
12	PIC2ADTEN8nj12	Output value of GTM timer output selected by PIC2ADTCFG3 bits 7 to 0. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
11	PIC2ADTEN8nj11	Output value of GTM timer output selected by PIC2ADTCFG2 bits 31 to 24. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
10	PIC2ADTEN8nj10	Output value of GTM timer output selected by PIC2ADTCFG2 bits 23 to 16. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
9	PIC2ADTEN8nj9	Output value of GTM timer output selected by PIC2ADTCFG2 bits 15 to 8. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
8	PIC2ADTEN8nj8	Output value of GTM timer output selected by PIC2ADTCFG2 bits 7 to 0. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
7	PIC2ADTEN8nj7	Output value of GTM timer output selected by PIC2ADTCFG1 bits 31 to 24. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
6	PIC2ADTEN8nj6	Output value of GTM timer output selected by PIC2ADTCFG1 bits 23 to 16. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.

Table 33.19 PIC2ADTEN8nj Register Contents (3/3)

Bit Position	Bit Name	Function
5	PIC2ADTEN8nj5	Output value of GTM timer output selected by PIC2ADTCFG1 bits 15 to 8. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
4	PIC2ADTEN8nj4	Output value of GTM timer output selected by PIC2ADTCFG1 bits 7 to 0. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
3	PIC2ADTEN8nj3	Output value of GTM timer output selected by PIC2ADTCFG0 bits 31 to 24. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
2	PIC2ADTEN8nj2	Output value of GTM timer output selected by PIC2ADTCFG0 bits 23 to 16. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
1	PIC2ADTEN8nj1	Output value of GTM timer output selected by PIC2ADTCFG0 bits 15 to 8. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.
0	PIC2ADTEN8nj0	Output value of GTM timer output selected by PIC2ADTCFG0 bits 7 to 0. 0: Selection of the above signal as an ADCH trigger is disabled. 1: Selection of the above signal as an ADCH trigger is enabled.

33.3.2.8 PIC2DSADTEN0n0 — DSADC Start Trigger Output Control Register n0 (n = 0 to 5)

DSADC start trigger output control register n0 enables a trigger source from ATU timer G compare match to be selected as a trigger for starting DSADC.

(n=0 DSADC00, n=1 DSADC10, n=2 DSADC20, n=3 DSADC12, n=4 DSADC13, n=5 DSADC11)

Value after reset: 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2DSADTEN0n015	PIC2DSADTEN0n014	PIC2DSADTEN0n013	PIC2DSADTEN0n012	PIC2DSADTEN0n011	PIC2DSADTEN0n010	PIC2DSADTEN0n009	PIC2DSADTEN0n008	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 33.20 PIC2DSADTEN0n0 Register Contents

Bit Position	Bit Name	Function
15	PIC2DSADTEN0n015	OCRG7 compare match interrupt (ATU5 timer G subblock 7) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
14	PIC2DSADTEN0n014	OCRG6 compare match interrupt (ATU5 timer G subblock 6) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
13	PIC2DSADTEN0n013	OCRG5 compare match interrupt (ATU5 timer G subblock 5) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
12	PIC2DSADTEN0n012	OCRG4 compare match interrupt (ATU5 timer G subblock 4) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
11	PIC2DSADTEN0n011	OCRG3 compare match interrupt (ATU5 timer G subblock 3) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
10	PIC2DSADTEN0n010	OCRG2 compare match interrupt (ATU5 timer G subblock 2) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
9	PIC2DSADTEN0n009	OCRG1 compare match interrupt (ATU5 timer G subblock 1) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
8	PIC2DSADTEN0n008	OCRG0 compare match interrupt (ATU5 timer G subblock 0) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
7 to 0	—	Reserved When writing to these bits, write 0.

33.3.2.9 PIC2DSADTEN0n1 — DSADC Start Trigger Output Control Register n1 (n = 0 to 5)

DSADC start trigger output control register n1 enables a trigger source from ATU timer D compare match A to be selected as a trigger for starting DSADC.

(n=0 DSADC00, n=1 DSADC10, n=2 DSADC20, n=3 DSADC12, n=4 DSADC13, n=5 DSADC11)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2DSADTEN0n115	PIC2DSADTEN0n114	PIC2DSADTEN0n113	PIC2DSADTEN0n112	PIC2DSADTEN0n111	PIC2DSADTEN0n110	PIC2DSADTEN0n119	PIC2DSADTEN0n118	PIC2DSADTEN0n117	PIC2DSADTEN0n116	PIC2DSADTEN0n115	PIC2DSADTEN0n114	PIC2DSADTEN0n113	PIC2DSADTEN0n112	PIC2DSADTEN0n111	PIC2DSADTEN0n110
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.21 PIC2DSADTEN0n1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing to these bits, write 0.
15	PIC2DSADTEN0n115	OCR1D63 compare match interrupt (ATU5 timer D subblock 6 channel 3 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
14	PIC2DSADTEN0n114	OCR1D62 compare match interrupt (ATU5 timer D subblock 6 channel 2 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
13	PIC2DSADTEN0n113	OCR1D61 compare match interrupt (ATU5 timer D subblock 6 channel 1 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
12	PIC2DSADTEN0n112	OCR1D60 compare match interrupt (ATU5 timer D subblock 6 channel 0 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
11	PIC2DSADTEN0n111	OCR1D43 compare match interrupt (ATU5 timer D subblock 4 channel 3 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
10	PIC2DSADTEN0n110	OCR1D42 compare match interrupt (ATU5 timer D subblock 4 channel 2 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
9	PIC2DSADTEN0n119	OCR1D41 compare match interrupt (ATU5 timer D subblock 4 channel 1 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.

Table 33.21 PIC2DSADTEN0n1 Register Contents (2/2)

Bit Position	Bit Name	Function
8	PIC2DSADTEN0n18	OCR1D40 compare match interrupt (ATU5 timer D subblock 4 channel 0 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
7	PIC2DSADTEN0n17	OCR1D23 compare match interrupt (ATU5 timer D subblock 2 channel 3 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
6	PIC2DSADTEN0n16	OCR1D22 compare match interrupt (ATU5 timer D subblock 2 channel 2 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
5	PIC2DSADTEN0n15	OCR1D21 compare match interrupt (ATU5 timer D subblock 2 channel 1 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
4	PIC2DSADTEN0n14	OCR1D20 compare match interrupt (ATU5 timer D subblock 2 channel 0 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
3	PIC2DSADTEN0n13	OCR1D03 compare match interrupt (ATU5 timer D subblock 0 channel 3 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
2	PIC2DSADTEN0n12	OCR1D02 compare match interrupt (ATU5 timer D subblock 0 channel 2 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
1	PIC2DSADTEN0n11	OCR1D01 compare match interrupt (ATU5 timer D subblock 0 channel 1 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
0	PIC2DSADTEN0n10	OCR1D00 compare match interrupt (ATU5 timer D subblock 0 channel 0 compare match A interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.

33.3.2.10 PIC2DSADTEN0n2 — DSADC Start Trigger Output Control Register n2 (n = 0 to 5)

DSADC start trigger output control register n2 enables a trigger source from ATU timer D compare match B to be selected as a trigger for starting DSADC.

(n=0 DSADC00, n=1 DSADC10, n=2 DSADC20, n=3 DSADC12, n=4 DSADC13, n=5 DSADC11)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2DSADTEN0n215	PIC2DSADTEN0n214	PIC2DSADTEN0n213	PIC2DSADTEN0n212	PIC2DSADTEN0n211	PIC2DSADTEN0n210	PIC2DSADTEN0n209	PIC2DSADTEN0n208	PIC2DSADTEN0n207	PIC2DSADTEN0n206	PIC2DSADTEN0n205	PIC2DSADTEN0n204	PIC2DSADTEN0n203	PIC2DSADTEN0n202	PIC2DSADTEN0n201	PIC2DSADTEN0n200
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.22 PIC2DSADTEN0n2 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing to these bits, write 0.
15	PIC2DSADTEN0n215	OCR2D63 compare match interrupt (ATU5 timer D subblock 6 channel 3 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
14	PIC2DSADTEN0n214	OCR2D62 compare match interrupt (ATU5 timer D subblock 6 channel 2 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
13	PIC2DSADTEN0n213	OCR2D61 compare match interrupt (ATU5 timer D subblock 6 channel 1 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
12	PIC2DSADTEN0n212	OCR2D60 compare match interrupt (ATU5 timer D subblock 6 channel 0 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
11	PIC2DSADTEN0n211	OCR2D43 compare match interrupt (ATU5 timer D subblock 4 channel 3 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
10	PIC2DSADTEN0n210	OCR2D42 compare match interrupt (ATU5 timer D subblock 4 channel 2 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
9	PIC2DSADTEN0n209	OCR2D41 compare match interrupt (ATU5 timer D subblock 4 channel 1 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.

Table 33.22 PIC2DSADTEN0n2 Register Contents (2/2)

Bit Position	Bit Name	Function
8	PIC2DSADTEN0n28	OCR2D40 compare match interrupt (ATU5 timer D subblock 4 channel 0 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
7	PIC2DSADTEN0n27	OCR2D23 compare match interrupt (ATU5 timer D subblock 2 channel 3 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
6	PIC2DSADTEN0n26	OCR2D22 compare match interrupt (ATU5 timer D subblock 2 channel 2 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
5	PIC2DSADTEN0n25	OCR2D21 compare match interrupt (ATU5 timer D subblock 2 channel 1 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
4	PIC2DSADTEN0n24	OCR2D20 compare match interrupt (ATU5 timer D subblock 2 channel 0 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
3	PIC2DSADTEN0n23	OCR2D03 compare match interrupt (ATU5 timer D subblock 0 channel 3 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
2	PIC2DSADTEN0n22	OCR2D02 compare match interrupt (ATU5 timer D subblock 0 channel 2 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
1	PIC2DSADTEN0n21	OCR2D01 compare match interrupt (ATU5 timer D subblock 0 channel 1 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
0	PIC2DSADTEN0n20	OCR2D00 compare match interrupt (ATU5 timer D subblock 0 channel 0 compare match B interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.

33.3.2.11 PIC2DSADTEN0n3 — DSADC Start Trigger Output Control Register n3 (n = 0 to 5)

DSADC start trigger output control register n3 enables a trigger source from GTM and ENCA to be selected as a trigger for starting DSADC. Whether enabled trigger source is used or not is defined by set timer output in DSADC Start/Stop trigger output configuration register.

(n=0 DSADC00, n=1 DSADC10, n=2 DSADC20, n=3 DSADC12, n=4 DSADC13, n=5 DSADC11)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2DSADTEN0n331	PIC2DSADTEN0n330	PIC2DSADTEN0n329	PIC2DSADTEN0n328	PIC2DSADTEN0n327	PIC2DSADTEN0n326	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PIC2DSADTEN0n311	PIC2DSADTEN0n310	PIC2DSADTEN0n309	PIC2DSADTEN0n308	PIC2DSADTEN0n307	PIC2DSADTEN0n306	PIC2DSADTEN0n305	PIC2DSADTEN0n304	PIC2DSADTEN0n303	PIC2DSADTEN0n302	PIC2DSADTEN0n301	PIC2DSADTEN0n300
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.23 PIC2DSADTEN0n3 Register Contents (1/2)

Bit Position	Bit Name	Function
31	PIC2DSADTEN0n331	Selects the INTENCA1IEC as a trigger source of DSADCn. (ENCA1 ENCATIEC Clear interrupt signal by encoder) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
30	PIC2DSADTEN0n330	Selects the INTENCA1I1 as a trigger source of DSADCn. (ENCA1 ENCATINT1 Compare 1 match or Capture 1 interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
29	PIC2DSADTEN0n329	Selects the INTENCA1I0 as a trigger source of DSADCn. (ENCA1 ENCATINT0 Compare 0 match or Capture 0 interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
28	PIC2DSADTEN0n328	Selects the INTENCA0IEC as a trigger source of DSADCn. (ENCA0 ENCATIEC Clear interrupt signal by encoder) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
27	PIC2DSADTEN0n327	Selects the INTENCA0I1 as a trigger source of DSADCn. (ENCA0 ENCATINT1 Compare 1 match or Capture 1 interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
26	PIC2DSADTEN0n326	Selects the INTENCA0I0 as a trigger source of DSADCn. (ENCA0 ENCATINT0 Compare 0 match or Capture 0 interrupt) 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
25 to 12	—	Reserved When writing to these bits, write 0.
11	PIC2DSADTEN0n311	Output value of GTM timer output selected by PIC2DSADTCFG2 bist 31 to 24 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.

Table 33.23 PIC2DSADTEN0n3 Register Contents (2/2)

Bit Position	Bit Name	Function
10	PIC2DSADTEN0n310	Output value of GTM timer output selected by PIC2DSADTCFG2 bits 23 to 16 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
9	PIC2DSADTEN0n39	Output value of GTM timer output selected by PIC2DSADTCFG2 bits 15 to 8 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
8	PIC2DSADTEN0n38	Output value of GTM timer output selected by PIC2DSADTCFG2 bits 7 to 0 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
7	PIC2DSADTEN0n37	Output value of GTM timer output selected by PIC2DSADTCFG1 bits 31 to 24 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
6	PIC2DSADTEN0n36	Output value of GTM timer output selected by PIC2DSADTCFG1 bits 23 to 16 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
5	PIC2DSADTEN0n35	Output value of GTM timer output selected by PIC2DSADTCFG1 bit 15 to 8 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
4	PIC2DSADTEN0n34	Output value of GTM timer output selected by PIC2DSADTCFG1 bit 7 to 0 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
3	PIC2DSADTEN0n33	Output value of GTM timer output selected by PIC2DSADTCFG0 bits 31 to 24 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
2	PIC2DSADTEN0n32	Output value of GTM timer output selected by PIC2DSADTCFG0 bits 23 to 16 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
1	PIC2DSADTEN0n31	Output value of GTM timer output selected by PIC2DSADTCFG0 bits 15 to 8 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.
0	PIC2DSADTEN0n30	Output value of GTM timer output selected by PIC2DSADTCFG0 bits 7 to 0 0: Selection of the above signal as a DSADCn start trigger is disabled. 1: Selection of the above signal as a DSADCn start trigger is enabled.

33.3.2.12 PIC2DSADTEN1n0 — DSADC Stop Trigger Output Control Register n0 (n = 0 to 5)

DSADC stop trigger output control register n0 enables a trigger source from ATU timer G compare match to be selected as a trigger for stopping DSADC.

(n=0 DSADC00, n=1 DSADC10, n=2 DSADC20, n=3 DSADC12, n=4 DSADC13, n=5 DSADC11)

Value after reset: 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2DSADTEN1n015	PIC2DSADTEN1n014	PIC2DSADTEN1n013	PIC2DSADTEN1n012	PIC2DSADTEN1n011	PIC2DSADTEN1n010	PIC2DSADTEN1n009	PIC2DSADTEN1n008	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 33.24 PIC2DSADTEN1n0 Register Contents

Bit Position	Bit Name	Function
15	PIC2DSADTEN1n015	OCRG7 compare match interrupt (ATU5 timer G subblock 7) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
14	PIC2DSADTEN1n014	OCRG6 compare match interrupt (ATU5 timer G subblock 6) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
13	PIC2DSADTEN1n013	OCRG5 compare match interrupt (ATU5 timer G subblock 5) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
12	PIC2DSADTEN1n012	OCRG4 compare match interrupt (ATU5 timer G subblock 4) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
11	PIC2DSADTEN1n011	OCRG3 compare match interrupt (ATU5 timer G subblock 3) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
10	PIC2DSADTEN1n010	OCRG2 compare match interrupt (ATU5 timer G subblock 2) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
9	PIC2DSADTEN1n009	OCRG1 compare match interrupt (ATU5 timer G subblock 1) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
8	PIC2DSADTEN1n008	OCRG0 compare match interrupt (ATU5 timer G subblock 0) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
7 to 0	—	Reserved When writing to these bits, write 0.

33.3.2.13 PIC2DSADTEN1n1 — DSADC Stop Trigger Output Control Register n1 (n = 0 to 5)

DSADC stop trigger output control register n1 enables a trigger source from ATU timer D compare match A to be selected as a trigger for stopping DSADC.

(n=0 DSADC00, n=1 DSADC10, n=2 DSADC20, n=3 DSADC12, n=4 DSADC13, n=5 DSADC11)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2DSADTEN1n115	PIC2DSADTEN1n114	PIC2DSADTEN1n113	PIC2DSADTEN1n112	PIC2DSADTEN1n111	PIC2DSADTEN1n110	PIC2DSADTEN1n119	PIC2DSADTEN1n118	PIC2DSADTEN1n117	PIC2DSADTEN1n116	PIC2DSADTEN1n115	PIC2DSADTEN1n114	PIC2DSADTEN1n113	PIC2DSADTEN1n112	PIC2DSADTEN1n111	PIC2DSADTEN1n110
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.25 PIC2DSADTEN1n1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing to these bits, write 0.
15	PIC2DSADTEN1n115	OCR1D63 compare match interrupt (ATU5 timer D subblock 6 channel 3 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
14	PIC2DSADTEN1n114	OCR1D62 compare match interrupt (ATU5 timer D subblock 6 channel 2 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
13	PIC2DSADTEN1n113	OCR1D61 compare match interrupt (ATU5 timer D subblock 6 channel 1 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
12	PIC2DSADTEN1n112	OCR1D60 compare match interrupt (ATU5 timer D subblock 6 channel 0 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
11	PIC2DSADTEN1n111	OCR1D43 compare match interrupt (ATU5 timer D subblock 4 channel 3 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
10	PIC2DSADTEN1n110	OCR1D42 compare match interrupt (ATU5 timer D subblock 4 channel 2 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
9	PIC2DSADTEN1n119	OCR1D41 compare match interrupt (ATU5 timer D subblock 4 channel 1 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.

Table 33.25 PIC2DSADTEN1n1 Register Contents (2/2)

Bit Position	Bit Name	Function
8	PIC2DSADTEN1n18	OCR1D40 compare match interrupt (ATU5 timer D subblock 4 channel 0 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
7	PIC2DSADTEN1n17	OCR1D23 compare match interrupt (ATU5 timer D subblock 2 channel 3 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
6	PIC2DSADTEN1n16	OCR1D22 compare match interrupt (ATU5 timer D subblock 2 channel 2 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
5	PIC2DSADTEN1n15	OCR1D21 compare match interrupt (ATU5 timer D subblock 2 channel 1 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
4	PIC2DSADTEN1n14	OCR1D20 compare match interrupt (ATU5 timer D subblock 2 channel 0 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
3	PIC2DSADTEN1n13	OCR1D03 compare match interrupt (ATU5 timer D subblock 0 channel 3 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
2	PIC2DSADTEN1n12	OCR1D02 compare match interrupt (ATU5 timer D subblock 0 channel 2 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
1	PIC2DSADTEN1n11	OCR1D01 compare match interrupt (ATU5 timer D subblock 0 channel 1 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
0	PIC2DSADTEN1n10	OCR1D00 compare match interrupt (ATU5 timer D subblock 0 channel 0 compare match A interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.

33.3.2.14 PIC2DSADTEN1n2 — DSADC Stop Trigger Output Control Register n2 (n = 0 to 5)

DSADC stop trigger output control register n2 enables a trigger source from ATU timer D compare match B to be selected as a trigger for stopping DSADC.

(n=0 DSADC00, n=1 DSADC10, n=2 DSADC20, n=3 DSADC12, n=4 DSADC13, n=5 DSADC11)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2DSADTEN1n215	PIC2DSADTEN1n214	PIC2DSADTEN1n213	PIC2DSADTEN1n212	PIC2DSADTEN1n211	PIC2DSADTEN1n210	PIC2DSADTEN1n29	PIC2DSADTEN1n28	PIC2DSADTEN1n27	PIC2DSADTEN1n26	PIC2DSADTEN1n25	PIC2DSADTEN1n24	PIC2DSADTEN1n23	PIC2DSADTEN1n22	PIC2DSADTEN1n21	PIC2DSADTEN1n20
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.26 PIC2DSADTEN1n2 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing to these bits, write 0.
15	PIC2DSADTEN1n215	OCR2D63 compare match interrupt (ATU5 timer D subblock 6 channel 3 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
14	PIC2DSADTEN1n214	OCR2D62 compare match interrupt (ATU5 timer D subblock 6 channel 2 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
13	PIC2DSADTEN1n213	OCR2D61 compare match interrupt (ATU5 timer D subblock 6 channel 1 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
12	PIC2DSADTEN1n212	OCR2D60 compare match interrupt (ATU5 timer D subblock 6 channel 0 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
11	PIC2DSADTEN1n211	OCR2D43 compare match interrupt (ATU5 timer D subblock 4 channel 3 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
10	PIC2DSADTEN1n210	OCR2D42 compare match interrupt (ATU5 timer D subblock 4 channel 2 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
9	PIC2DSADTEN1n29	OCR2D41 compare match interrupt (ATU5 timer D subblock 4 channel 1 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.

Table 33.26 PIC2DSADTEN1n2 Register Contents (2/2)

Bit Position	Bit Name	Function
8	PIC2DSADTEN1n28	OCR2D40 compare match interrupt (ATU5 timer D subblock 4 channel 0 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
7	PIC2DSADTEN1n27	OCR2D23 compare match interrupt (ATU5 timer D subblock 2 channel 3 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
6	PIC2DSADTEN1n26	OCR2D22 compare match interrupt (ATU5 timer D subblock 2 channel 2 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
5	PIC2DSADTEN1n25	OCR2D21 compare match interrupt (ATU5 timer D subblock 2 channel 1 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
4	PIC2DSADTEN1n24	OCR2D20 compare match interrupt (ATU5 timer D subblock 2 channel 0 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
3	PIC2DSADTEN1n23	OCR2D03 compare match interrupt (ATU5 timer D subblock 0 channel 3 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
2	PIC2DSADTEN1n22	OCR2D02 compare match interrupt (ATU5 timer D subblock 0 channel 2 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
1	PIC2DSADTEN1n21	OCR2D01 compare match interrupt (ATU5 timer D subblock 0 channel 1 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
0	PIC2DSADTEN1n20	OCR2D00 compare match interrupt (ATU5 timer D subblock 0 channel 0 compare match B interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.

33.3.2.15 PIC2DSADTEN1n3 — DSADC Stop Trigger Output Control Register n3 (n = 0 to 5)

DSADC stop trigger output control register n3 enables a trigger source from GTM and ENCA encoder to be selected as a trigger for stopping DSADC. Whether enabled trigger source is used or not is defined by set timer output in DSADC Start / Stop trigger output configuration register.

(n=0 DSADC00, n=1 DSADC10, n=2 DSADC20, n=3 DSADC12, n=4 DSADC13, n=5 DSADC11)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2DSADTEN1n331	PIC2DSADTEN1n330	PIC2DSADTEN1n329	PIC2DSADTEN1n328	PIC2DSADTEN1n327	PIC2DSADTEN1n326	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PIC2DSADTEN1n311	PIC2DSADTEN1n310	PIC2DSADTEN1n309	PIC2DSADTEN1n308	PIC2DSADTEN1n307	PIC2DSADTEN1n306	PIC2DSADTEN1n305	PIC2DSADTEN1n304	PIC2DSADTEN1n303	PIC2DSADTEN1n302	PIC2DSADTEN1n301	PIC2DSADTEN1n300
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.27 PIC2DSADTEN1n3 Register Contents (1/2)

Bit Position	Bit Name	Function
31	PIC2DSADTEN1n331	Selects the INTENCA1IEC as a trigger source of DSADCn. (ENCA1 ENCATIEC Clear interrupt signal by encoder) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
30	PIC2DSADTEN1n330	Selects the INTENCA1I1 as a trigger source of DSADCn. (ENCA1 ENCATINT1 Compare 1 match or Capture 1 interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
29	PIC2DSADTEN1n329	Selects the INTENCA1I0 as a trigger source of DSADCn. (ENCA1 ENCATINT0 Compare 0 match or Capture 0 interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
28	PIC2DSADTEN1n328	Selects the INTENCA0IEC as a trigger source of DSADCn. (ENCA0 ENCATIEC Clear interrupt signal by encoder) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
27	PIC2DSADTEN1n327	Selects the INTENCA0I1 as a trigger source of DSADCn. (ENCA0 ENCATINT1 Compare 1 match or Capture 1 interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
26	PIC2DSADTEN1n326	Selects the INTENCA0I0 as a trigger source of DSADCn. (ENCA0 ENCATINT0 Compare 0 match or Capture 0 interrupt) 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
25 to 12	—	Reserved When writing to these bits, write 0.
11	PIC2DSADTEN1n311	Output value of GTM timer output selected by PIC2DSADTCFG2 bits 31 to 24 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.

Table 33.27 PIC2DSADTEN1n3 Register Contents (2/2)

Bit Position	Bit Name	Function
10	PIC2DSADTEN1n310	Output value of GTM timer output selected by PIC2DSADTCFG2 bits 23 to 16 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
9	PIC2DSADTEN1n39	Output value of GTM timer output selected by PIC2DSADTCFG2 bits 15 to 8 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
8	PIC2DSADTEN1n38	Output value of GTM timer output selected by PIC2DSADTCFG2 bits 7 to 0 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
7	PIC2DSADTEN1n37	Output value of GTM timer output selected by PIC2DSADTCFG1 bits 31 to 24 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
6	PIC2DSADTEN1n36	Output value of GTM timer output selected by PIC2DSADTCFG1 bits 23 to 16 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
5	PIC2DSADTEN1n35	Output value of GTM timer output selected by PIC2DSADTCFG1 bits 15 to 8 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
4	PIC2DSADTEN1n34	Output value of GTM timer output selected by PIC2DSADTCFG1 bits 7 to 0 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
3	PIC2DSADTEN1n33	Output value of GTM timer output selected by PIC2DSADTCFG0 bits 31 to 24 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
2	PIC2DSADTEN1n32	Output value of GTM timer output selected by PIC2DSADTCFG0 bits 23 to 16 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
1	PIC2DSADTEN1n31	Output value of GTM timer output selected by PIC2DSADTCFG0 bits 15 to 8 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.
0	PIC2DSADTEN1n30	Output value of GTM timer output selected by PIC2DSADTCFG0 bits 7 to 0 0: Selection of the above signal as a DSADCn end trigger is disabled. 1: Selection of the above signal as a DSADCn end trigger is enabled.

33.3.2.16 PIC2DSADCATSEL0 — DSADC Trigger Selection Control Register 0

DSADC trigger selection control register 0 selects a trigger for starting DSADC_j (j = 00, 10, 20, 12, 13, 11).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DSADC ATSEL011	DSADC ATSEL010	DSADC ATSEL009	DSADC ATSEL008	DSADC ATSEL007	DSADC ATSEL006	DSADC ATSEL005	DSADC ATSEL004	DSADC ATSEL003	DSADC ATSEL002	DSADC ATSEL001	DSADC ATSEL000
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.28 PIC2DSADCATSEL0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 12	—	Reserved When writing to these bits, write 0.
11	DSADCATSEL011	Selects the DSADTRG11 pin as a trigger source for starting DSADC11. 0: DSADTRG11 is not selected. 1: DSADTRG11 is selected.
10	DSADCATSEL010	Selects the trigger selected by DSADC start trigger output control register 5 _j (j = 0 to 2) as a trigger source for starting DSADC11. 0: Does not select the trigger selected by register PIC2DSADTEN05 _j . 1: Selects the trigger selected by register PIC2DSADTEN05 _j .
9	DSADCATSEL009	Selects the DSADTRG13 pin as a trigger source for starting DSADC13. 0: DSADTRG13 is not selected. 1: DSADTRG13 is selected.
8	DSADCATSEL008	Selects the trigger selected by DSADC start trigger output control register 4 _j (j = 0 to 2) as a trigger source for starting DSADC13. 0: Does not select the trigger selected by register PIC2DSADTEN04 _j . 1: Selects the trigger selected by register PIC2DSADTEN04 _j .
7	DSADCATSEL007	Selects the DSADTRG12 pin as a trigger source for starting DSADC12. 0: DSADTRG12 is not selected. 1: DSADTRG12 is selected.
6	DSADCATSEL006	Selects the trigger selected by DSADC start trigger output control register 3 _j (j = 0 to 2) as a trigger source for starting DSADC12. 0: Does not select the trigger selected by register PIC2DSADTEN03 _j . 1: Selects the trigger selected by register PIC2DSADTEN03 _j .
5	DSADCATSEL005	Selects the DSADTRG20 pin as a trigger source for starting DSADC20. 0: DSADTRG20 is not selected. 1: DSADTRG20 is selected.
4	DSADCATSEL004	Selects the trigger selected by DSADC start trigger output control register 2 _j (j = 0 to 2) as a trigger source for starting DSADC20. 0: Does not select the trigger selected by register PIC2DSADTEN02 _j . 1: Selects the trigger selected by register PIC2DSADTEN02 _j .
3	DSADCATSEL003	Selects the DSADTRG10 pin as a trigger source for starting DSADC10. 0: DSADTRG10 is not selected. 1: DSADTRG10 is selected.

Table 33.28 PIC2DSADCATSEL0 Register Contents (2/2)

Bit Position	Bit Name	Function
2	DSADCATSEL002	Selects the trigger selected by DSADC start trigger output control register 1j (j = 0 to 2) as a trigger source for starting DSADC10. 0: Does not select the trigger selected by register PIC2DSADTEN01j. 1: Selects the trigger selected by register PIC2DSADTEN01j.
1	DSADCATSEL001	Selects the DSADTRG00 pin as a trigger source for starting DSADC00. 0: DSADTRG00 is not selected. 1: DSADTRG00 is selected.
0	DSADCATSEL000	Selects the trigger selected by DSADC start trigger output control register 0j (j = 0 to 2) as a trigger source for starting DSADC00. 0: Does not select the trigger selected by register PIC2DSADTEN00j. 1: Selects the trigger selected by register PIC2DSADTEN00j.

33.3.2.17 PIC2DSADCATSEL1 — DSADC Trigger Selection Control Register 1

DSADC trigger selection control register 1 selects a trigger for stopping DSADC_j (j = 00, 10, 20, 12, 13, 11).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DSADC ATSEL1 11	DSADC ATSEL1 10	DSADC ATSEL1 09	DSADC ATSEL1 08	DSADC ATSEL1 07	DSADC ATSEL1 06	DSADC ATSEL1 05	DSADC ATSEL1 04	DSADC ATSEL1 03	DSADC ATSEL1 02	DSADC ATSEL1 01	DSADC ATSEL1 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.29 PIC2DSADCATSEL1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 12	—	Reserved When writing to these bits, write 0.
11	DSADCATSEL111	Selects the DSADTRG11 pin as a trigger source for stopping DSADC11. 0: DSADTRG11 is not selected. 1: DSADTRG11 is selected.
10	DSADCATSEL110	Selects the trigger selected by DSADC stop trigger output control register 5j (j = 0 to 2) as a trigger source for stopping DSADC11. 0: Does not select the trigger selected by register PIC2DSADTEN15j. 1: Selects the trigger selected by register PIC2DSADTEN15j.
9	DSADCATSEL109	Selects the DSADTRG13 pin as a trigger source for stopping DSADC13. 0: DSADTRG13 is not selected. 1: DSADTRG13 is selected.
8	DSADCATSEL108	Selects the trigger selected by DSADC stop trigger output control register 4j (j = 0 to 2) as a trigger source for stopping DSADC13. 0: Does not select the trigger selected by register PIC2DSADTEN14j. 1: Selects the trigger selected by register PIC2DSADTEN14j.
7	DSADCATSEL107	Selects the DSADTRG12 pin as a trigger source for stopping DSADC12. 0: DSADTRG12 is not selected. 1: DSADTRG12 is selected.
6	DSADCATSEL106	Selects the trigger selected by DSADC stop trigger output control register 3j (j = 0 to 2) as a trigger source for stopping DSADC12. 0: Does not select the trigger selected by register PIC2DSADTEN13j. 1: Selects the trigger selected by register PIC2DSADTEN13j.
5	DSADCATSEL105	Selects the DSADTRG20 pin as a trigger source for stopping DSADC20. 0: DSADTRG20 is not selected. 1: DSADTRG20 is selected.
4	DSADCATSEL104	Selects the trigger selected by DSADC stop trigger output control register 2j (j = 0 to 2) as a trigger source for stopping DSADC20. 0: Does not select the trigger selected by register PIC2DSADTEN12j. 1: Selects the trigger selected by register PIC2DSADTEN12j.
3	DSADCATSEL103	Selects the DSADTRG10 pin as a trigger source for stopping DSADC10. 0: DSADTRG10 is not selected. 1: DSADTRG10 is selected.

Table 33.29 PIC2DSADCATSEL1 Register Contents (2/2)

Bit Position	Bit Name	Function
2	DSADCATSEL102	Selects the trigger selected by DSADC stop trigger output control register 1j (j = 0 to 2) as a trigger source for stopping DSADC10. 0: Does not select the trigger selected by register PIC2DSADTEN11j. 1: Selects the trigger selected by register PIC2DSADTEN11j.
1	DSADCATSEL101	Selects the DSADTRG00 pin as a trigger source for stopping DSADC00. 0: DSADTRG00 is not selected. 1: DSADTRG00 is selected.
0	DSADCATSEL100	Selects the trigger selected by DSADC stop trigger output control register 0j (j = 0 to 2) as a trigger source for stopping DSADC00. 0: Does not select the trigger selected by register PIC2DSADTEN10j. 1: Selects the trigger selected by register PIC2DSADTEN10j.

33.3.2.18 PIC2ENCAnTSELj — Encoder Timer n Trigger Selection Control Register j (n = 0 to 1; j = 0 to 1)

The PIC2ENCAnTSELj register selects a trigger for ENCA_n ENCATTN_j (n = 0 to 1; j = 0 to 1).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2ENCAnTS ELj31	PIC2ENCAnTS ELj30	PIC2ENCAnTS ELj29	PIC2ENCAnTS ELj28	PIC2ENCAnTS ELj27	PIC2ENCAnTS ELj26	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	PIC2ENCAnTS ELj03	PIC2ENCAnTS ELj02	PIC2ENCAnTS ELj01	PIC2ENCAnTS ELj00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 33.30 PIC2ENCAnTSELj Register Contents (1/2)

Bit Position	Bit Name	Function
31	PIC2ENCAnTSELj31	Selects the INTENCA1IEC as a trigger source of ENCA _n ENCATTN _j . (ENCA1 ENCATIEC Clear interrupt signal by encoder) 0: INTENCA1IEC is not selected. 1: INTENCA1IEC is selected.
30	PIC2ENCAnTSELj30	Selects the INTENCA111 as a trigger source of ENCA _n ENCATTN _j . (ENCA1 ENCATINT1 Compare 1 match or Capture 1 interrupt) 0: INTENCA111 is not selected. 1: INTENCA111 is selected.
29	PIC2ENCAnTSELj29	Selects the INTENCA110 as a trigger source of ENCA _n ENCATTN _j . (ENCA1 ENCATINT0 Compare 0 match or Capture 0 interrupt) 0: INTENCA110 is not selected. 1: INTENCA110 is selected.
28	PIC2ENCAnTSELj28	Selects the INTENCA0IEC as a trigger source of ENCA _n ENCATTN _j . (ENCA0 ENCATIEC Clear interrupt signal by encoder) 0: INTENCA0IEC is not selected. 1: INTENCA0IEC is selected.
27	PIC2ENCAnTSELj27	Selects the INTENCA011 as a trigger source of ENCA _n ENCATTN _j . (ENCA0 ENCATINT1 Compare 1 match or Capture 1 interrupt) 0: INTENCA011 is not selected. 1: INTENCA011 is selected.
26	PIC2ENCAnTSELj26	Selects the INTENCA010 as a trigger source of ENCA _n ENCATTN _j . (ENCA0 ENCATINT0 Compare 0 match or Capture 0 interrupt) 0: INTENCA010 is not selected. 1: INTENCA010 is selected.
25 to 4	—	Reserved When writing to these bits, write 0.
3	PIC2ENCAnTSELj03	Output value of GTM timer output selected by PIC2ENCTCFG1 bit 15 to 8. 0: Selection of the above signal as an ENCA trigger is disabled. 1: Selection of the above signal as an ENCA trigger is enabled.

Table 33.30 PIC2ENCAnTSELj Register Contents (2/2)

Bit Position	Bit Name	Function
2	PIC2ENCAnTSELj02	Output value of GTM timer output selected by PIC2ENCTCFG1 bit 7 to 0. 0: Selection of the above signal as an ENCA trigger is disabled. 1: Selection of the above signal as an ENCA trigger is enabled.
1	PIC2ENCAnTSELj01	Output value of GTM timer output selected by PIC2ENCTCFG0 bit 15 to 8. 0: Selection of the above signal as an ENCA trigger is disabled. 1: Selection of the above signal as an ENCA trigger is enabled.
0	PIC2ENCAnTSELj00	Output value of GTM timer output selected by PIC2ENCTCFG0 bit 7 to 0. 0: Selection of the above signal as an ENCA trigger is disabled. 1: Selection of the above signal as an ENCA trigger is enabled.

33.3.2.19 PIC2CADTEN000 — CADC Start Trigger Output Control Register 00

CADC start trigger output control register 00 enables a trigger source from ATU timer G compare match to be selected as a trigger for starting CADC00.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2CADTEN0015	PIC2CADTEN0014	PIC2CADTEN0013	PIC2CADTEN0012	PIC2CADTEN0011	PIC2CADTEN0010	PIC2CADTEN0009	PIC2CADTEN0008	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 33.31 PIC2CADTEN000 Register Contents

Bit Position	Bit Name	Function
15	PIC2CADTEN00015	OCRG7 compare match interrupt (ATU5 timer G subblock 7) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
14	PIC2CADTEN00014	OCRG6 compare match interrupt (ATU5 timer G subblock 6) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
13	PIC2CADTEN00013	OCRG5 compare match interrupt (ATU5 timer G subblock 5) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
12	PIC2CADTEN00012	OCRG4 compare match interrupt (ATU5 timer G subblock 4) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
11	PIC2CADTEN00011	OCRG3 compare match interrupt (ATU5 timer G subblock 3) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
10	PIC2CADTEN00010	OCRG2 compare match interrupt (ATU5 timer G subblock 2) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
9	PIC2CADTEN0009	OCRG1 compare match interrupt (ATU5 timer G subblock 1) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
8	PIC2CADTEN0008	OCRG0 compare match interrupt (ATU5 timer G subblock 0) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
7 to 0	—	Reserved When writing to these bits, write 0.

33.3.2.20 PIC2CADTEN001 — CADC Start Trigger Output Control Register 01

CADC start trigger output control register 01 enables a trigger source from ATU timer D compare match A to be selected as a trigger for starting CADC00.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2CADTEN0115	PIC2CADTEN0114	PIC2CADTEN0113	PIC2CADTEN0112	PIC2CADTEN0111	PIC2CADTEN0110	PIC2CADTEN0109	PIC2CADTEN0108	PIC2CADTEN0107	PIC2CADTEN0106	PIC2CADTEN0105	PIC2CADTEN0104	PIC2CADTEN0103	PIC2CADTEN0102	PIC2CADTEN0101	PIC2CADTEN0100
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.32 PIC2CADTEN001 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing to these bits, write 0.
15	PIC2CADTEN00115	OCR1D63 compare match interrupt (ATU5 timer D subblock 6 channel 3 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
14	PIC2CADTEN00114	OCR1D62 compare match interrupt (ATU5 timer D subblock 6 channel 2 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
13	PIC2CADTEN00113	OCR1D61 compare match interrupt (ATU5 timer D subblock 6 channel 1 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
12	PIC2CADTEN00112	OCR1D60 compare match interrupt (ATU5 timer D subblock 6 channel 0 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
11	PIC2CADTEN00111	OCR1D43 compare match interrupt (ATU5 timer D subblock 4 channel 3 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
10	PIC2CADTEN00110	OCR1D42 compare match interrupt (ATU5 timer D subblock 4 channel 2 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
9	PIC2CADTEN00109	OCR1D41 compare match interrupt (ATU5 timer D subblock 4 channel 1 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
8	PIC2CADTEN00108	OCR1D40 compare match interrupt (ATU5 timer D subblock 4 channel 0 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.

Table 33.32 PIC2CADTEN001 Register Contents (2/2)

Bit Position	Bit Name	Function
7	PIC2CADTEN0017	OCR1D23 compare match interrupt (ATU5 timer D subblock 2 channel 3 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
6	PIC2CADTEN0016	OCR1D22 compare match interrupt (ATU5 timer D subblock 2 channel 2 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
5	PIC2CADTEN0015	OCR1D21 compare match interrupt (ATU5 timer D subblock 2 channel 1 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
4	PIC2CADTEN0014	OCR1D20 compare match interrupt (ATU5 timer D subblock 2 channel 0 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
3	PIC2CADTEN0013	OCR1D03 compare match interrupt (ATU5 timer D subblock 0 channel 3 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
2	PIC2CADTEN0012	OCR1D02 compare match interrupt (ATU5 timer D subblock 0 channel 2 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
1	PIC2CADTEN0011	OCR1D01 compare match interrupt (ATU5 timer D subblock 0 channel 1 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
0	PIC2CADTEN0010	OCR1D00 compare match interrupt (ATU5 timer D subblock 0 channel 0 compare match A interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.

33.3.2.21 PIC2CADTEN002 — CADC Start Trigger Output Control Register 02

CADC start trigger output control register 02 enables a trigger source from ATU timer D compare match B to be selected as a trigger for starting CADC00.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2CADTEN00215	PIC2CADTEN00214	PIC2CADTEN00213	PIC2CADTEN00212	PIC2CADTEN00211	PIC2CADTEN00210	PIC2CADTEN00209	PIC2CADTEN00208	PIC2CADTEN00207	PIC2CADTEN00206	PIC2CADTEN00205	PIC2CADTEN00204	PIC2CADTEN00203	PIC2CADTEN00202	PIC2CADTEN00201	PIC2CADTEN00200
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.33 PIC2CADTEN002 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing to these bits, write 0.
15	PIC2CADTEN00215	OCR2D63 compare match interrupt (ATU5 timer D subblock 6 channel 3 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
14	PIC2CADTEN00214	OCR2D62 compare match interrupt (ATU5 timer D subblock 6 channel 2 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
13	PIC2CADTEN00213	OCR2D61 compare match interrupt (ATU5 timer D subblock 6 channel 1 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
12	PIC2CADTEN00212	OCR2D60 compare match interrupt (ATU5 timer D subblock 6 channel 0 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
11	PIC2CADTEN00211	OCR2D43 compare match interrupt (ATU5 timer D subblock 4 channel 3 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
10	PIC2CADTEN00210	OCR2D42 compare match interrupt (ATU5 timer D subblock 4 channel 2 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
9	PIC2CADTEN00209	OCR2D41 compare match interrupt (ATU5 timer D subblock 4 channel 1 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
8	PIC2CADTEN00208	OCR2D40 compare match interrupt (ATU5 timer D subblock 4 channel 0 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.

Table 33.33 PIC2CADTEN002 Register Contents (2/2)

Bit Position	Bit Name	Function
7	PIC2CADTEN0027	OCR2D23 compare match interrupt (ATU5 timer D subblock 2 channel 3 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
6	PIC2CADTEN0026	OCR2D22 compare match interrupt (ATU5 timer D subblock 2 channel 2 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
5	PIC2CADTEN0025	OCR2D21 compare match interrupt (ATU5 timer D subblock 2 channel 1 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
4	PIC2CADTEN0024	OCR2D20 compare match interrupt (ATU5 timer D subblock 2 channel 0 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
3	PIC2CADTEN0023	OCR2D03 compare match interrupt (ATU5 timer D subblock 0 channel 3 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
2	PIC2CADTEN0022	OCR2D02 compare match interrupt (ATU5 timer D subblock 0 channel 2 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
1	PIC2CADTEN0021	OCR2D01 compare match interrupt (ATU5 timer D subblock 0 channel 1 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
0	PIC2CADTEN0020	OCR2D00 compare match interrupt (ATU5 timer D subblock 0 channel 0 compare match B interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.

33.3.2.22 PIC2CADTEN003 — CADC Start Trigger Output Control Register 03

CADC start trigger output control register 03 enables a trigger source from GTM and ENCA to be selected as a trigger for starting CADC00. Whether enabled trigger source is used or not is defined by set timer output in CADC Start / Stop trigger output configuration register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2CADTEN00331	PIC2CADTEN00330	PIC2CADTEN00329	PIC2CADTEN00328	PIC2CADTEN00327	PIC2CADTEN00326	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PIC2CADTEN00301	PIC2CADTEN00300
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 33.34 PIC2CADTEN003 Register Contents (1/2)

Bit Position	Bit Name	Function
31	PIC2CADTEN00331	Selects the INTENCA1IEC as a trigger source of CADC00. (ENCA1 ENCATIEC Clear interrupt signal by encoder) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
30	PIC2CADTEN00330	Selects the INTENCA1I1 as a trigger source of CADC00. (ENCA1 ENCATINT1 Compare 1 match or Capture 1 interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
29	PIC2CADTEN00329	Selects the INTENCA1I0 as a trigger source of CADC00. (ENCA1 ENCATINT0 Compare 0 match or Capture 0 interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
28	PIC2CADTEN00328	Selects the INTENCA0IEC as a trigger source of CADC00. (ENCA0 ENCATIEC Clear interrupt signal by encoder) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
27	PIC2CADTEN00327	Selects the INTENCA0I1 as a trigger source of CADC00. (ENCA0 ENCATINT1 Compare 1 match or Capture 1 interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
26	PIC2CADTEN00326	Selects the INTENCA0I0 as a trigger source of CADC00. (ENCA0 ENCATINT0 Compare 0 match or Capture 0 interrupt) 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
25 to 2	—	Reserved When writing to these bits, write 0.

Table 33.34 PIC2CADTEN003 Register Contents (2/2)

Bit Position	Bit Name	Function
1	PIC2CADTEN0031	Output value of GTM timer output selected by PIC2CADTCFG0 bits 15 to 8 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.
0	PIC2CADTEN0030	Output value of GTM timer output selected by PIC2CADTCFG0 bit 7 to 0 0: Selection of the above signal as a CADC00 start trigger is disabled. 1: Selection of the above signal as a CADC00 start trigger is enabled.

33.3.2.23 PIC2CADTEN100 — CADC Stop Trigger Output Control Register 00

CADC stop trigger output control register 00 enables a trigger source from ATU timer G compare match to be selected as a trigger for stopping CADC00.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2CADTEN10015	PIC2CADTEN10014	PIC2CADTEN10013	PIC2CADTEN10012	PIC2CADTEN10011	PIC2CADTEN10010	PIC2CADTEN10009	PIC2CADTEN10008	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 33.35 PIC2CADTEN100 Register Contents

Bit Position	Bit Name	Function
15	PIC2CADTEN10015	OCRG7 compare match interrupt (ATU5 timer G subblock 7) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
14	PIC2CADTEN10014	OCRG6 compare match interrupt (ATU5 timer G subblock 6) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
13	PIC2CADTEN10013	OCRG5 compare match interrupt (ATU5 timer G subblock 5) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
12	PIC2CADTEN10012	OCRG4 compare match interrupt (ATU5 timer G subblock 4) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
11	PIC2CADTEN10011	OCRG3 compare match interrupt (ATU5 timer G subblock 3) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
10	PIC2CADTEN10010	OCRG2 compare match interrupt (ATU5 timer G subblock 2) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
9	PIC2CADTEN10009	OCRG1 compare match interrupt (ATU5 timer G subblock 1) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
8	PIC2CADTEN10008	OCRG0 compare match interrupt (ATU5 timer G subblock 0) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
7 to 0	—	Reserved When writing to these bits, write 0.

33.3.2.24 PIC2CADTEN101 — CADC Stop Trigger Output Control Register 01

CADC stop trigger output control register 01 enables a trigger source from ATU timer D compare match A to be selected as a trigger for stopping CADC00.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2CADTEN10115	PIC2CADTEN10114	PIC2CADTEN10113	PIC2CADTEN10112	PIC2CADTEN10111	PIC2CADTEN10110	PIC2CADTEN10109	PIC2CADTEN10108	PIC2CADTEN10107	PIC2CADTEN10106	PIC2CADTEN10105	PIC2CADTEN10104	PIC2CADTEN10103	PIC2CADTEN10102	PIC2CADTEN10101	PIC2CADTEN10100
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.36 PIC2CADTEN101 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing to these bits, write 0.
15	PIC2CADTEN10115	OCR1D63 compare match interrupt (ATU5 timer D subblock 6 channel 3 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
14	PIC2CADTEN10114	OCR1D62 compare match interrupt (ATU5 timer D subblock 6 channel 2 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
13	PIC2CADTEN10113	OCR1D61 compare match interrupt (ATU5 timer D subblock 6 channel 1 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
12	PIC2CADTEN10112	OCR1D60 compare match interrupt (ATU5 timer D subblock 6 channel 0 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
11	PIC2CADTEN10111	OCR1D43 compare match interrupt (ATU5 timer D subblock 4 channel 3 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
10	PIC2CADTEN10110	OCR1D42 compare match interrupt (ATU5 timer D subblock 4 channel 2 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
9	PIC2CADTEN10109	OCR1D41 compare match interrupt (ATU5 timer D subblock 4 channel 1 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
8	PIC2CADTEN10108	OCR1D40 compare match interrupt (ATU5 timer D subblock 4 channel 0 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.

Table 33.36 PIC2CADTEN101 Register Contents (2/2)

Bit Position	Bit Name	Function
7	PIC2CADTEN1017	OCR1D23 compare match interrupt (ATU5 timer D subblock 2 channel 3 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
6	PIC2CADTEN1016	OCR1D22 compare match interrupt (ATU5 timer D subblock 2 channel 2 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
5	PIC2CADTEN1015	OCR1D21 compare match interrupt (ATU5 timer D subblock 2 channel 1 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
4	PIC2CADTEN1014	OCR1D20 compare match interrupt (ATU5 timer D subblock 2 channel 0 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
3	PIC2CADTEN1013	OCR1D03 compare match interrupt (ATU5 timer D subblock 0 channel 3 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
2	PIC2CADTEN1012	OCR1D02 compare match interrupt (ATU5 timer D subblock 0 channel 2 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
1	PIC2CADTEN1011	OCR1D01 compare match interrupt (ATU5 timer D subblock 0 channel 1 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
0	PIC2CADTEN1010	OCR1D00 compare match interrupt (ATU5 timer D subblock 0 channel 0 compare match A interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.

33.3.2.25 PIC2CADTEN102 — CADC Stop Trigger Output Control Register 02

CADC stop trigger output control register 02 enables a trigger source from ATU timer D compare match B to be selected as a trigger for stopping CADC00.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2CADTEN10215	PIC2CADTEN10214	PIC2CADTEN10213	PIC2CADTEN10212	PIC2CADTEN10211	PIC2CADTEN10210	PIC2CADTEN10209	PIC2CADTEN10208	PIC2CADTEN10207	PIC2CADTEN10206	PIC2CADTEN10205	PIC2CADTEN10204	PIC2CADTEN10203	PIC2CADTEN10202	PIC2CADTEN10201	PIC2CADTEN10200
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.37 PIC2CADTEN102 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing to these bits, write 0.
15	PIC2CADTEN10215	OCR2D63 compare match interrupt (ATU5 timer D subblock 6 channel 3 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
14	PIC2CADTEN10214	OCR2D62 compare match interrupt (ATU5 timer D subblock 6 channel 2 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
13	PIC2CADTEN10213	OCR2D61 compare match interrupt (ATU5 timer D subblock 6 channel 1 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
12	PIC2CADTEN10212	OCR2D60 compare match interrupt (ATU5 timer D subblock 6 channel 0 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
11	PIC2CADTEN10211	OCR2D43 compare match interrupt (ATU5 timer D subblock 4 channel 3 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
10	PIC2CADTEN10210	OCR2D42 compare match interrupt (ATU5 timer D subblock 4 channel 2 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
9	PIC2CADTEN10209	OCR2D41 compare match interrupt (ATU5 timer D subblock 4 channel 1 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
8	PIC2CADTEN10208	OCR2D40 compare match interrupt (ATU5 timer D subblock 4 channel 0 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.

Table 33.37 PIC2CADTEN102 Register Contents (2/2)

Bit Position	Bit Name	Function
7	PIC2CADTEN1027	OCR2D23 compare match interrupt (ATU5 timer D subblock 2 channel 3 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
6	PIC2CADTEN1026	OCR2D22 compare match interrupt (ATU5 timer D subblock 2 channel 2 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
5	PIC2CADTEN1025	OCR2D21 compare match interrupt (ATU5 timer D subblock 2 channel 1 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
4	PIC2CADTEN1024	OCR2D20 compare match interrupt (ATU5 timer D subblock 2 channel 0 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
3	PIC2CADTEN1023	OCR2D03 compare match interrupt (ATU5 timer D subblock 0 channel 3 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
2	PIC2CADTEN1022	OCR2D02 compare match interrupt (ATU5 timer D subblock 0 channel 2 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
1	PIC2CADTEN1021	OCR2D01 compare match interrupt (ATU5 timer D subblock 0 channel 1 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
0	PIC2CADTEN1020	OCR2D00 compare match interrupt (ATU5 timer D subblock 0 channel 0 compare match B interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.

33.3.2.26 PIC2CADTEN103 — CADC Stop Trigger Output Control Register 03

CADC stop trigger output control register 03 enables trigger source from GTM and ENCA to be selected as a trigger for stopping CADC00. Whether enabled trigger source is used or not is defined by set timer output in CADC Start / Stop trigger output configuration register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2CADTEN10331	PIC2CADTEN10330	PIC2CADTEN10329	PIC2CADTEN10328	PIC2CADTEN10327	PIC2CADTEN10326	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PIC2CADTEN1031	PIC2CADTEN1030
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 33.38 PIC2CADTEN103 Register Contents (1/2)

Bit Position	Bit Name	Function
31	PIC2CADTEN10331	Selects the INTENCA1IEC as a trigger source of CADC00. (ENCA1 ENCATIEC Clear interrupt signal by encoder) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
30	PIC2CADTEN10330	Selects the INTENCA1I1 as a trigger source of CADC00. (ENCA1 ENCATINT1 Compare 1 match or Capture 1 interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
29	PIC2CADTEN10329	Selects the INTENCA1I0 as a trigger source of CADC00. (ENCA1 ENCATINT0 Compare 0 match or Capture 0 interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
28	PIC2CADTEN10328	Selects the ENCAT0INT1 as a trigger source of CADC00. (ENCA0 ENCATIEC Clear interrupt signal by encoder) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
27	PIC2CADTEN10327	Selects the ENCAT0INT1 as a trigger source of CADC00. (ENCA0 ENCATINT1 Compare 1 match or Capture 1 interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
26	PIC2CADTEN10326	Selects the ENCAT0INT0 as a trigger source of CADC00. (ENCA0 ENCATINT0 Compare 0 match or Capture 0 interrupt) 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
25 to 2	—	Reserved When writing to these bits, write 0.

Table 33.38 PIC2CADTEN103 Register Contents (2/2)

Bit Position	Bit Name	Function
1	PIC2CADTEN1031	Output value of GTM timer output selected by PIC2CADTCFG0 bits 15 to 8 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.
0	PIC2CADTEN1030	Output value of GTM timer output selected by PIC2CADTCFG0 bits 7 to 0 0: Selection of the above signal as a CADC00 end trigger is disabled. 1: Selection of the above signal as a CADC00 end trigger is enabled.

33.3.2.27 PIC2ADCATSEL0 — CADC Trigger Selection Control Register 0

CADC trigger selection control register 0 selects a trigger for starting CADC_j (j = 00).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CADCA TSEL00 1	CADCA TSEL00 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 33.39 PIC2ADCATSEL0 Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When writing to these bits, write 0.
1	CADCATSEL001	Selects the CADTRG00 pin as a trigger source for starting CADC00. 0: CADTRG00 is not selected. 1: CADTRG00 is selected.
0	CADCATSEL000	Selects the trigger selected by CADC start trigger output control register 0j (j = 0 to 3) as a trigger source for starting CADC00. 0: Does not select the trigger selected by register PIC2CADTEN00j. 1: Selects the trigger selected by register PIC2CADTEN00j.

33.3.2.28 PIC2ADCATSEL1 — CADC Trigger Selection Control Register 1

CADC trigger selection control register 1 selects a trigger for stopping CADC_j (j = 00).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CADCA TSEL10 1	CADCA TSEL10 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 33.40 PIC2ADCATSEL1 Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When writing to these bits, write 0.
1	CADCATSEL101	Selects the CADTRG00 pin as a trigger source for stopping CADC00. 0: CADTRG00 is not selected. 1: CADTRG00 is selected.
0	CADCATSEL100	Selects the trigger selected by CADC stop trigger output control register 0j (j = 0 to 2) as a trigger source for stopping CADC00. 0: Does not select the trigger selected by register PIC2CADTEN10j. 1: Selects the trigger selected by register PIC2CADTEN10j.

33.3.2.29 PIC2ADTCFGx — A/D Converter Trigger Output Configuration Register x (x = 0 to 4)

The PIC2ADTCFGx register assigns timer output of GTM ([A]TOM) to PIC2ADTEN800-834 (x = 0 to 4).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2ADTCFGx[31:24]								PIC2ADTCFGx[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2ADTCFGx[15:8]								PIC2ADTCFGx[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.41 PIC2ADTCFGx Register Contents

Bit Position	Bit Name	Function
31 to 24	PIC2ADTCFGx[31:24]	Selects the GTM [A]TOM output as a PIC2ADTEN800-834 Bit[3 + (x × 4)] register configuration. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.
23 to 16	PIC2ADTCFGx[23:16]	Selects the GTM [A]TOM output as a PIC2ADTEN800-834 Bit[2 + (x × 4)] register configuration. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.
15 to 8	PIC2ADTCFGx[15:8]	Selects the GTM [A]TOM output as a PIC2ADTEN800-834 Bit[1 + (x × 4)] register configuration. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.
7 to 0	PIC2ADTCFGx[7:0]	Selects the GTM [A]TOM output as a PIC2ADTEN800-834 Bit[0 + (x × 4)] register configuration. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.

Note 1. E2xFCC1: (n = 0 to 4, m = 0 to 15), E2M: (n = 0 to 3, m = 0 to 15)

Note 2. E2xFCC1: (n = 0 to 8, m = 0 to 7), E2M: (n = 0 to 7, m = 0 to 7)

33.3.2.30 PIC2DSADTCFGx — DSADC Start / Stop Trigger Output Configuration Register x (x = 0 to 2)

The PIC2DSADTCFGx register assigns timer output of GTM ([A]TOM) to PIC2DSADTENjk3 (j = 0 to 1, k = 0 to 5, x = 0 to 2).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2DSADTCFGx[31:24]								PIC2DSADTCFGx[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2DSADTCFGx[15:8]								PIC2DSADTCFGx[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.42 PIC2DSADTCFGx Register Contents

Bit Position	Bit Name	Function
31 to 24	PIC2DSADTCFGx [31:24]	Selects the GTM [A]TOM output as a PIC2DSADTENjk3 (j = 0 to 1, k = 0 to 5) Bit[3 + (x × 4)] register configuration. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.
23 to 16	PIC2DSADTCFGx [23:16]	Selects the GTM [A]TOM output as a PIC2DSADTENjk3 (j = 0 to 1, k = 0 to 5) Bit[2 + (x × 4)] register configuration. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.
15 to 8	PIC2DSADTCFGx [15:8]	Selects the GTM [A]TOM output as a PIC2DSADTENjk3 (j = 0 to 1, k = 0 to 5) Bit[1 + (x × 4)] register configuration. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.
7 to 0	PIC2DSADTCFGx [7:0]	Selects the GTM [A]TOM output as a PIC2DSADTENjk3 (j = 0 to 1, k = 0 to 5) Bit[0 + (x × 4)] register configuration. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.

Note 1. E2xFCC1: (n = 0 to 4, m = 0 to 15), E2M: (n = 0 to 3, m = 0 to 15)

Note 2. E2xFCC1: (n = 0 to 8, m = 0 to 7), E2M: (n = 0 to 7, m = 0 to 7)

33.3.2.31 PIC2CADTCFG0 — CADC Start / Stop Trigger Output Configuration Register 0

The PIC2CADTCFG0 register assigns timer output of GTM ([A]TOM) to PIC2CADTENjk3 (j = 0 to 1, k = 0).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2CADTCFG0[15:8]								PIC2CADTCFG0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.43 PIC2CADTCFG0 Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing to these bits, write 0.
15 to 8	PIC2CADTCFG0 [23:16]	Selects the GTM [A]TOM output as a PIC2CADTENjk3 (j = 0 to 1, k = 0) (Bit 1) register configuration. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.
7 to 0	PIC2CADTCFG0[7:0]	Selects the GTM [A]TOM output as a PIC2CADTENjk3 (j = 0 to 1, k = 0) (Bit 0) register configuration. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.

Note 1. E2xFCC1: (n = 0 to 4, m = 0 to 15), E2M: (n = 0 to 3, m = 0 to 15)

Note 2. E2xFCC1: (n = 0 to 8, m = 0 to 7), E2M: (n = 0 to 7, m = 0 to 7)

33.3.2.32 PIC2DSADTSENx — DSADC Read Gate Trigger Selection Control Register x (x = 0 to 1)

The PIC2DSADTSENx register selects a trigger for DSADC read gate. (x = 0 to 1).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2DSADTSENx[31:24]								PIC2DSADTSENx[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2DSADTSENx[15:8]								PIC2DSADTSENx[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.44 PIC2DSADTSENx Register Contents

Bit Position	Bit Name	Function
31 to 24	PIC2DSADTSENx [31:24]	Selects the GTM [A]TOM output as a read gate source for DSADC. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected. 193 + (n × 8) + m: GTM TIM[n]_CH[m] ^{*3} is selected.
23 to 16	PIC2DSADTSENx [23:16]	Selects the GTM [A]TOM output as a read gate source for DSADC. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected. 193 + (n × 8) + m: GTM TIM[n]_CH[m] ^{*3} is selected.
15 to 8	PIC2DSADTSENx [15:8]	Selects the GTM [A]TOM output as a read gate source for DSADC. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected. 193 + (n × 8) + m: GTM TIM[n]_CH[m] ^{*3} is selected.
7 to 0	PIC2DSADTSENx [7:0]	Selects the GTM [A]TOM output as a read gate source for DSADC. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected. 193 + (n × 8) + m: GTM TIM[n]_CH[m] ^{*3} is selected.

Note 1. E2xFCC1: (n = 0 to 4, m = 0 to 15), E2M: (n = 0 to 3, m = 0 to 15)

Note 2. E2xFCC1: (n = 0 to 8, m = 0 to 7), E2M: (n = 0 to 7, m = 0 to 7)

Note 3. E2xFCC1: (n = 0 to 6, m = 0 to 7), E2M: (n = 0 to 6, m = 0 to 7)

Table 33.45 List of Destination

PIC2DSADTSENx	Bit Position	Destination
PIC2DSADTSEN0	31 to 24	DSADC12
PIC2DSADTSEN0	23 to 16	DSADC20
PIC2DSADTSEN0	15 to 8	DSADC10
PIC2DSADTSEN0	7 to 0	DSADC00
PIC2DSADTSEN1	31 to 24	Reserved
PIC2DSADTSEN1	23 to 16	Reserved
PIC2DSADTSEN1	15 to 8	DSADC11
PIC2DSADTSEN1	7 to 0	DSADC13

33.3.2.33 PIC2CADTSEN0 — CADC Read Gate Trigger Selection Control Register 0

The PIC2CADTSEN0 register selects a trigger for CADC00 read gate.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PIC2CADTSEN0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.46 PIC2CADTSEN0 Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When writing to these bits, write 0.
7 to 0	PIC2CADTSEN0[7:0]	Selects the GTM [A]TOM output as a read gate source for CADC00. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected. 193 + (n × 8) + m: GTM TIM[n]_CH[m] ^{*3} is selected.

Note 1. E2xFCC1: (n = 0 to 4, m = 0 to 15), E2M: (n = 0 to 3, m = 0 to 15)

Note 2. E2xFCC1: (n = 0 to 8, m = 0 to 7), E2M: (n = 0 to 7, m = 0 to 7)

Note 3. E2xFCC1: (n = 0 to 6, m = 0 to 7), E2M: (n = 0 to 6, m = 0 to 7)

33.3.2.34 PIC2DFETSENx — DFE Timer Trigger Selection Control Register x (x = 0 to 11)

The PIC2DFETSENx register selects a source of a timer trigger of DFE (x = 0 to 11).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2DFETSENx[31:24]								PIC2DFETSENx[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2DFETSENx[15:8]								PIC2DFETSENx[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.47 PIC2DFETSENx Register Contents

Bit Position	Bit Name	Function
31 to 24	PIC2DFETSENx [31:24]	Selects the GTM [A]TOM output as a trigger source for DFE. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.
23 to 16	PIC2DFETSENx [23:16]	Selects the GTM [A]TOM output as a trigger source for DFE. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.
15 to 8	PIC2DFETSENx [15:8]	Selects the GTM [A]TOM output as a trigger source for DFE. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.
7 to 0	PIC2DFETSENx [7:0]	Selects the GTM [A]TOM output as a trigger source for DFE. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.

Note 1. E2xFCC1: (n = 0 to 4, m = 0 to 15), E2M: (n = 0 to 3, m = 0 to 15)

Note 2. E2xFCC1: (n = 0 to 8, m = 0 to 7), E2M: (n = 0 to 7, m = 0 to 7)

Table 33.48 List of Destination (1/2)

PIC2DFETSENx	Bit Position	Destination
PIC2DFETSEN0	31 to 24	Timer trigger Compare (Compare A3)
PIC2DFETSEN0	23 to 16	Timer trigger Compare (Compare A2)
PIC2DFETSEN0	15 to 8	Timer trigger Compare (Compare A1)
PIC2DFETSEN0	7 to 0	Timer trigger Compare (Compare A0)
PIC2DFETSEN1	31 to 24	Timer trigger Compare (Compare A7)
PIC2DFETSEN1	23 to 16	Timer trigger Compare (Compare A6)
PIC2DFETSEN1	15 to 8	Timer trigger Compare (Compare A5)
PIC2DFETSEN1	7 to 0	Timer trigger Compare (Compare A4)
PIC2DFETSEN2	31 to 24	Timer trigger Compare (Compare A11)
PIC2DFETSEN2	23 to 16	Timer trigger Compare (Compare A10)

Table 33.48 List of Destination (2/2)

PIC2DFETSENx	Bit Position	Destination
PIC2DFETSEN2	15 to 8	Timer trigger Compare (Compare A9)
PIC2DFETSEN2	7 to 0	Timer trigger Compare (Compare A8)
PIC2DFETSEN3	31 to 24	Timer trigger Compare (Compare A15)
PIC2DFETSEN3	23 to 16	Timer trigger Compare (Compare A14)
PIC2DFETSEN3	15 to 8	Timer trigger Compare (Compare A13)
PIC2DFETSEN3	7 to 0	Timer trigger Compare (Compare A12)
PIC2DFETSEN4	31 to 24	Timer trigger Compare (Compare A19)
PIC2DFETSEN4	23 to 16	Timer trigger Compare (Compare A18)
PIC2DFETSEN4	15 to 8	Timer trigger Compare (Compare A17)
PIC2DFETSEN4	7 to 0	Timer trigger Compare (Compare A16)
PIC2DFETSEN5	31 to 24	Timer trigger Compare (Compare B3)
PIC2DFETSEN5	23 to 16	Timer trigger Compare (Compare B2)
PIC2DFETSEN5	15 to 8	Timer trigger Compare (Compare B1)
PIC2DFETSEN5	7 to 0	Timer trigger Compare (Compare B0)
PIC2DFETSEN6	31 to 24	Timer trigger Compare (Compare B7)
PIC2DFETSEN6	23 to 16	Timer trigger Compare (Compare B6)
PIC2DFETSEN6	15 to 8	Timer trigger Compare (Compare B5)
PIC2DFETSEN6	7 to 0	Timer trigger Compare (Compare B4)
PIC2DFETSEN7	31 to 24	Timer trigger Compare (Compare B11)
PIC2DFETSEN7	23 to 16	Timer trigger Compare (Compare B10)
PIC2DFETSEN7	15 to 8	Timer trigger Compare (Compare B9)
PIC2DFETSEN7	7 to 0	Timer trigger Compare (Compare B8)
PIC2DFETSEN8	31 to 24	Timer trigger Compare (Compare B15)
PIC2DFETSEN8	23 to 16	Timer trigger Compare (Compare B14)
PIC2DFETSEN8	15 to 8	Timer trigger Compare (Compare B13)
PIC2DFETSEN8	7 to 0	Timer trigger Compare (Compare B12)
PIC2DFETSEN9	31 to 24	Timer trigger Compare (Compare B19)
PIC2DFETSEN9	23 to 16	Timer trigger Compare (Compare B18)
PIC2DFETSEN9	15 to 8	Timer trigger Compare (Compare B17)
PIC2DFETSEN9	7 to 0	Timer trigger Compare (Compare B16)
PIC2DFETSEN10	31 to 24	Reserved
PIC2DFETSEN10	23 to 16	Timer DFE Capture trigger 2
PIC2DFETSEN10	15 to 8	Timer DFE Capture trigger 1
PIC2DFETSEN10	7 to 0	Timer DFE Capture trigger 0
PIC2DFETSEN11	31 to 24	Reserved
PIC2DFETSEN11	23 to 16	Reserved
PIC2DFETSEN11	15 to 8	Reserved
PIC2DFETSEN11	7 to 0	Timer FIFO Capture trigger

33.3.2.35 PIC2DFETEDGSEL0 — DFE Timer Trigger Edge Selection Control Register 0

The PIC2DFETEDGSEL0 register selects an effective edge of the GTM timer output as trigger signal for DFE.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	PIC2DFE TEDGSE L019	PIC2DFE TEDGSE L018	PIC2DFE TEDGSE L017	PIC2DFE TEDGSE L016
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2DFE TEDGSE L015	PIC2DFE TEDGSE L014	PIC2DFE TEDGSE L013	PIC2DFE TEDGSE L012	PIC2DFE TEDGSE L011	PIC2DFE TEDGSE L010	PIC2DFE TEDGSE L009	PIC2DFE TEDGSE L008	PIC2DFE TEDGSE L007	PIC2DFE TEDGSE L006	PIC2DFE TEDGSE L005	PIC2DFE TEDGSE L004	PIC2DFE TEDGSE L003	PIC2DFE TEDGSE L002	PIC2DFE TEDGSE L001	PIC2DFE TEDGSE L000
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.49 PIC2DFETEDGSEL0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 20	—	Reserved When writing to these bits, write 0.
19	PIC2DFETEDGSEL0 19	Selects the effective edge of Timer trigger Compare A19 0: Rising edge is selected 1: Falling edge is selected
18	PIC2DFETEDGSEL0 18	Selects the effective edge of Timer trigger Compare A18 0: Rising edge is selected 1: Falling edge is selected
17	PIC2DFETEDGSEL0 17	Selects the effective edge of Timer trigger Compare A17 0: Rising edge is selected 1: Falling edge is selected
16	PIC2DFETEDGSEL0 16	Selects the effective edge of Timer trigger Compare A16 0: Rising edge is selected 1: Falling edge is selected
15	PIC2DFETEDGSEL0 15	Selects the effective edge of Timer trigger Compare A15 0: Rising edge is selected 1: Falling edge is selected
14	PIC2DFETEDGSEL0 14	Selects the effective edge of Timer trigger Compare A14 0: Rising edge is selected 1: Falling edge is selected
13	PIC2DFETEDGSEL0 13	Selects the effective edge of Timer trigger Compare A13 0: Rising edge is selected 1: Falling edge is selected
12	PIC2DFETEDGSEL0 12	Selects the effective edge of Timer trigger Compare A12 0: Rising edge is selected 1: Falling edge is selected
11	PIC2DFETEDGSEL0 11	Selects the effective edge of Timer trigger Compare A11 0: Rising edge is selected 1: Falling edge is selected

Table 33.49 PIC2DFETEDGSEL0 Register Contents (2/2)

Bit Position	Bit Name	Function
10	PIC2DFETEDGSEL0 10	Selects the effective edge of Timer trigger Compare A10 0: Rising edge is selected 1: Falling edge is selected
9	PIC2DFETEDGSEL0 09	Selects the effective edge of Timer trigger Compare A9 0: Rising edge is selected 1: Falling edge is selected
8	PIC2DFETEDGSEL0 08	Selects the effective edge of Timer trigger Compare A8 0: Rising edge is selected 1: Falling edge is selected
7	PIC2DFETEDGSEL0 07	Selects the effective edge of Timer trigger Compare A7 0: Rising edge is selected 1: Falling edge is selected
6	PIC2DFETEDGSEL0 06	Selects the effective edge of Timer trigger Compare A6 0: Rising edge is selected 1: Falling edge is selected
5	PIC2DFETEDGSEL0 05	Selects the effective edge of Timer trigger Compare A5 0: Rising edge is selected 1: Falling edge is selected
4	PIC2DFETEDGSEL0 04	Selects the effective edge of Timer trigger Compare A4 0: Rising edge is selected 1: Falling edge is selected
3	PIC2DFETEDGSEL0 03	Selects the effective edge of Timer trigger Compare A3 0: Rising edge is selected 1: Falling edge is selected
2	PIC2DFETEDGSEL0 02	Selects the effective edge of Timer trigger Compare A2 0: Rising edge is selected 1: Falling edge is selected
1	PIC2DFETEDGSEL0 01	Selects the effective edge of Timer trigger Compare A1 0: Rising edge is selected 1: Falling edge is selected
0	PIC2DFETEDGSEL0 00	Selects the effective edge of Timer trigger Compare A0 0: Rising edge is selected 1: Falling edge is selected

33.3.2.36 PIC2DFETEDGSEL1 — DFE Timer Trigger Edge Selection Control Register 1

The PIC2DFETEDGSEL1 register selects an effective edge of the GTM timer output as trigger signal for DFE.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	PIC2DFE TEDGSE L119	PIC2DFE TEDGSE L118	PIC2DFE TEDGSE L117	PIC2DFE TEDGSE L116
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2DFE TEDGSE L115	PIC2DFE TEDGSE L114	PIC2DFE TEDGSE L113	PIC2DFE TEDGSE L112	PIC2DFE TEDGSE L111	PIC2DFE TEDGSE L110	PIC2DFE TEDGSE L109	PIC2DFE TEDGSE L108	PIC2DFE TEDGSE L107	PIC2DFE TEDGSE L106	PIC2DFE TEDGSE L105	PIC2DFE TEDGSE L104	PIC2DFE TEDGSE L103	PIC2DFE TEDGSE L102	PIC2DFE TEDGSE L101	PIC2DFE TEDGSE L100
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.50 PIC2DFETEDGSEL1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 20	—	Reserved When writing to these bits, write 0.
19	PIC2DFETEDGSEL1 19	Selects the effective edge of Timer trigger Compare B19 0: Rising edge is selected 1: Falling edge is selected
18	PIC2DFETEDGSEL1 18	Selects the effective edge of Timer trigger Compare B18 0: Rising edge is selected 1: Falling edge is selected
17	PIC2DFETEDGSEL1 17	Selects the effective edge of Timer trigger Compare B17 0: Rising edge is selected 1: Falling edge is selected
16	PIC2DFETEDGSEL1 16	Selects the effective edge of Timer trigger Compare B16 0: Rising edge is selected 1: Falling edge is selected
15	PIC2DFETEDGSEL1 15	Selects the effective edge of Timer trigger Compare B15 0: Rising edge is selected 1: Falling edge is selected
14	PIC2DFETEDGSEL1 14	Selects the effective edge of Timer trigger Compare B14 0: Rising edge is selected 1: Falling edge is selected
13	PIC2DFETEDGSEL1 13	Selects the effective edge of Timer trigger Compare B13 0: Rising edge is selected 1: Falling edge is selected
12	PIC2DFETEDGSEL1 12	Selects the effective edge of Timer trigger Compare B12 0: Rising edge is selected 1: Falling edge is selected
11	PIC2DFETEDGSEL1 11	Selects the effective edge of Timer trigger Compare B11 0: Rising edge is selected 1: Falling edge is selected

Table 33.50 PIC2DFETEDGSEL1 Register Contents (2/2)

Bit Position	Bit Name	Function
10	PIC2DFETEDGSEL1 10	Selects the effective edge of Timer trigger Compare B10 0: Rising edge is selected 1: Falling edge is selected
9	PIC2DFETEDGSEL1 09	Selects the effective edge of Timer trigger Compare B9 0: Rising edge is selected 1: Falling edge is selected
8	PIC2DFETEDGSEL1 08	Selects the effective edge of Timer trigger Compare B8 0: Rising edge is selected 1: Falling edge is selected
7	PIC2DFETEDGSEL1 07	Selects the effective edge of Timer trigger Compare B7 0: Rising edge is selected 1: Falling edge is selected
6	PIC2DFETEDGSEL1 06	Selects the effective edge of Timer trigger Compare B6 0: Rising edge is selected 1: Falling edge is selected
5	PIC2DFETEDGSEL1 05	Selects the effective edge of Timer trigger Compare B5 0: Rising edge is selected 1: Falling edge is selected
4	PIC2DFETEDGSEL1 04	Selects the effective edge of Timer trigger Compare B4 0: Rising edge is selected 1: Falling edge is selected
3	PIC2DFETEDGSEL1 03	Selects the effective edge of Timer trigger Compare B3 0: Rising edge is selected 1: Falling edge is selected
2	PIC2DFETEDGSEL1 02	Selects the effective edge of Timer trigger Compare B2 0: Rising edge is selected 1: Falling edge is selected
1	PIC2DFETEDGSEL1 01	Selects the effective edge of Timer trigger Compare B1 0: Rising edge is selected 1: Falling edge is selected
0	PIC2DFETEDGSEL1 00	Selects the effective edge of Timer trigger Compare B0 0: Rising edge is selected 1: Falling edge is selected

33.3.2.37 PIC2DFETEDGSEL2 — DFE Timer Trigger Edge Selection Control Register 2

The PIC2DFETEDGSEL2 register selects an effective edge of the GTM timer output as trigger signal for DFE.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	PIC2DFE TEDGSE L203	PIC2DFE TEDGSE L202	PIC2DFE TEDGSE L201	PIC2DFE TEDGSE L200
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 33.51 PIC2DFETEDGSEL2 Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When writing to these bits, write 0.
3	PIC2DFETEDGSEL2 03	Selects the effective edge of Timer FIFO Capture trigger 0: Rising edge is selected 1: Falling edge is selected
2	PIC2DFETEDGSEL2 02	Selects the effective edge of Timer DFE Compare trigger 2 0: Rising edge is selected 1: Falling edge is selected
1	PIC2DFETEDGSEL2 01	Selects the effective edge of Timer DFE Compare trigger 1 0: Rising edge is selected 1: Falling edge is selected
0	PIC2DFETEDGSEL2 00	Selects the effective edge of Timer DFE Compare trigger 0 0: Rising edge is selected 1: Falling edge is selected

33.3.2.38 PIC2PSI5SENx — PSI5-S Sync Output Control Register x (x = 0 to 3)

The PIC2PSI5SENx register selects a source of a sync trigger of PSI5-S (x = 0 to 3).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2PSI5SENx[31:24]								PIC2PSI5SENx[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2PSI5SENx[15:8]								PIC2PSI5SENx[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.52 PIC2PSI5SENx Register Contents

Bit Position	Bit Name	Function
31 to 24	PIC2PSI5SENx [31:24]	Selects the GTM [A]TOM output as a signal source for PSI5-S. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.
23 to 16	PIC2PSI5SENx [23:16]	Selects the GTM [A]TOM output as a signal source for PSI5-S. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.
15 to 8	PIC2PSI5SENx [15:8]	Selects the GTM [A]TOM output as a signal source for PSI5-S. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.
7 to 0	PIC2PSI5SENx [7:0]	Selects the GTM [A]TOM output as a signal source for PSI5-S. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected.

Note 1. E2xFCC1: (n = 0 to 4, m = 0 to 15), E2M: (n = 0 to 3, m = 0 to 15)

Note 2. E2xFCC1: (n = 0 to 8, m = 0 to 7), E2M: (n = 0 to 7, m = 0 to 7)

Table 33.53 List of Destination (1/2)

PIC2PSI5SENx	Bit Position	Destination
PIC2PSI5SEN0	31 to 24	PSI5-S Synchronous trigger Ch3
PIC2PSI5SEN0	23 to 16	PSI5-S Synchronous trigger Ch2
PIC2PSI5SEN0	15 to 8	PSI5-S Synchronous trigger Ch1
PIC2PSI5SEN0	7 to 0	Reserved
PIC2PSI5SEN1	31 to 24	PSI5-S Synchronous trigger Ch7
PIC2PSI5SEN1	23 to 16	PSI5-S Synchronous trigger Ch6
PIC2PSI5SEN1	15 to 8	PSI5-S Synchronous trigger Ch5
PIC2PSI5SEN1	7 to 0	PSI5-S Synchronous trigger Ch4

Table 33.53 List of Destination (2/2)

PIC2PSI5SENx	Bit Position	Destination
PIC2PSI5SEN2	31 to 24	PSI5-S Timestamp counter B clear
PIC2PSI5SEN2	23 to 16	PSI5-S Timestamp counter A clear
PIC2PSI5SEN2	15 to 8	PSI5-S Timestamp counter B clock
PIC2PSI5SEN2	7 to 0	PSI5-S Timestamp counter A clock
PIC2PSI5SEN3	31 to 24	Reserved
PIC2PSI5SEN3	23 to 16	Reserved
PIC2PSI5SEN3	15 to 8	PSI5-S Timestamp counter B enable
PIC2PSI5SEN3	7 to 0	PSI5-S Timestamp counter A enable

33.3.2.39 PIC2GTMINENx — GTM Timer Input Module (TIM) Source Select Register x (x = 0, 2, 4, 6, 8, 10, 12, 14)

The PIC2GTMINENx register selects a source of timer input of GTM (TIM) (x = 0, 2, 4, 6, 8, 10, 12, 14).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2GTMINENx[31:24]								PIC2GTMINENx[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2GTMINENx[15:8]								PIC2GTMINENx[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.54 PIC2GTMINENx Register Contents

Bit Position	Bit Name	Function
31 to 24	PIC2GTMINENx [31:24]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[(x / 2)]_CH3) 0 to 127: Please refer to "Table 33.55".
23 to 16	PIC2GTMINENx [23:16]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[(x / 2)]_CH2) 0 to 127: Please refer to "Table 33.55".
15 to 8	PIC2GTMINENx [15:8]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[(x / 2)]_CH1) 0 to 127: Please refer to "Table 33.55".
7 to 0	PIC2GTMINENx [7:0]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[(x / 2)]_CH0) 0 to 127: Please refer to "Table 33.55".

Table 33.55 PIC2GTMINENx List of Function (1/3)

Function
0 : GPIO selected. (via Port Functions)
1 : ADI00 ADCH0 scan group 0 (SG0) end interrupt
2 : ADI01 ADCH0 scan group 1 (SG1) end interrupt
3 : ADI02 ADCH0 scan group 2 (SG2) end interrupt
4 : ADI03 ADCH0 scan group 3 (SG3) end interrupt
5 : ADI04 ADCH0 scan group 4 (SG4) end interrupt
6 : ADI10 ADCH1 scan group 0 (SG0) end interrupt
7 : ADI11 ADCH1 scan group 1 (SG1) end interrupt
8 : ADI12 ADCH1 scan group 2 (SG2) end interrupt
9 : ADI13 ADCH1 scan group 3 (SG3) end interrupt
10 : ADI14 ADCH1 scan group 4 (SG4) end interrupt
11 : ADI20 ADCH2 scan group 0 (SG0) end interrupt
12 : ADI21 ADCH2 scan group 1 (SG1) end interrupt
13 : ADI22 ADCH2 scan group 2 (SG2) end interrupt
14 : ADI23 ADCH2 scan group 3 (SG3) end interrupt
15 : ADI24 ADCH2 scan group 4 (SG4) end interrupt
16 : ADI30 ADCH3 scan group 0 (SG0) end interrupt
17 : ADI31 ADCH3 scan group 1 (SG1) end interrupt

Table 33.55 PIC2GTMINENx List of Function (2/3)

Function	
18	: ADI32 ADCH3 scan group 2 (SG2) end interrupt
19	: ADI33 ADCH3 scan group 3 (SG3) end interrupt
20	: ADI34 ADCH3 scan group 4 (SG4) end interrupt
21	: DSADC00 AD conversion end interrupt
22	: DSADC10 AD conversion end interrupt
23	: DSADC20 AD conversion end interrupt
24	: DSADC12 AD conversion end interrupt
25	: DSADC13 AD conversion end interrupt
26	: DSADC11 AD conversion end interrupt
27	: Reserved
28	: Reserved
29	: Reserved
30	: Reserved
31	: CADC00 AD conversion end interrupt
32	: Reserved
33	: DFE0 P/H updating trigger 0
34	: DFE0 P/H updating trigger 1
35	: DFE0 P/H updating trigger 2
36	: DFE0 P/H updating trigger 3
37	: DFE1 P/H updating trigger 0
38	: DFE1 P/H updating trigger 1
39	: DFE1 P/H updating trigger 2
40	: DFE1 P/H updating trigger 3
41	: DFE0 CH0 output data interrupt
42	: DFE0 CH1 output data interrupt
43	: DFE0 CH2 output data interrupt
44	: DFE0 CH3 output data interrupt
45	: DFE0 CH4 output data interrupt
46	: DFE0 CH5 output data interrupt
47	: DFE0 CH6 output data interrupt
48	: DFE0 CH7 output data interrupt
49	: DFE0 CH8 output data interrupt
50	: DFE0 CH9 output data interrupt
51	: DFE0 CH10 output data interrupt
52	: DFE0 CH11 output data interrupt
53	: DFE1 CH0 output data interrupt
54	: DFE1 CH1 output data interrupt
55	: DFE1 CH2 output data interrupt
56	: DFE1 CH3 output data interrupt
57	: Reserved
58	: Reserved
59	: Reserved
60	: Reserved
61	: Reserved
62	: Reserved

Table 33.55 PIC2GTMINENx List of Function (3/3)

Function	
63	: Reserved
64	: Reserved
65	: ENCA0 Encoder input (phase A)
66	: ENCA0 Encoder input (phase B)
67	: ENCA0 Encoder input (phase Z)
68	: ENCA0 Compare 0 match or Capture 0 interrupt signal
69	: ENCA0 Compare 1 match or Capture 1 interrupt signal
70	: ENCA0 Clear interrupt signal by encoder input (phase Z)
71	: ENCA0 Clock enable output
72	: ENCA0 Encoder count status output
73	: ENCA1 Encoder input (phase A)
74	: ENCA1 Encoder input (phase B)
75	: ENCA1 Encoder input (phase Z)
76	: ENCA1 Compare 0 match or Capture 0 interrupt signal
77	: ENCA1 Compare 1 match or Capture 1 interrupt signal
78	: ENCA1 Clear interrupt signal by encoder input (phase Z)
79	: ENCA1 Clock enable output
80	: ENCA1 Encoder count status output
81 to 127	: Reserved

33.3.2.40 PIC2GTMINENx — GTM Timer Input Module (TIM) Source Select Register x (x = 1, 3, 5, 7, 9, 11, 13, 15)

The PIC2GTMINENx register selects a source of timer input of GTM (TIM) (x = 1, 3, 5, 7, 9, 11, 13, 15).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2GTMINENx[31:24]								PIC2GTMINENx[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2GTMINENx[15:8]								PIC2GTMINENx[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.56 PIC2GTMINENx Register Contents

Bit Position	Bit Name	Function
31 to 24	PIC2GTMINENx [31:24]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[((x-1) / 2)]_CH7) 0 to 127: Please refer to "Table 33.57".
23 to 16	PIC2GTMINENx [23:16]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[((x-1) / 2)]_CH6) 0 to 127: Please refer to "Table 33.57".
15 to 8	PIC2GTMINENx [15:8]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[((x-1) / 2)]_CH5) 0 to 127: Please refer to "Table 33.57".
7 to 0	PIC2GTMINENx [7:0]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[((x-1) / 2)]_CH4) 0 to 127: Please refer to "Table 33.57".

Table 33.57 PIC2GTMINENx List of Function (1/3)

Function
0 : GPIO selected. (via Port Functions)
1 : ADI00 ADCH0 scan group 0 (SG0) end interrupt
2 : ADI01 ADCH0 scan group 1 (SG1) end interrupt
3 : ADI02 ADCH0 scan group 2 (SG2) end interrupt
4 : ADI03 ADCH0 scan group 3 (SG3) end interrupt
5 : ADI04 ADCH0 scan group 4 (SG4) end interrupt
6 : ADI10 ADCH1 scan group 0 (SG0) end interrupt
7 : ADI11 ADCH1 scan group 1 (SG1) end interrupt
8 : ADI12 ADCH1 scan group 2 (SG2) end interrupt
9 : ADI13 ADCH1 scan group 3 (SG3) end interrupt
10 : ADI14 ADCH1 scan group 4 (SG4) end interrupt
11 : ADI20 ADCH2 scan group 0 (SG0) end interrupt
12 : ADI21 ADCH2 scan group 1 (SG1) end interrupt
13 : ADI22 ADCH2 scan group 2 (SG2) end interrupt
14 : ADI23 ADCH2 scan group 3 (SG3) end interrupt
15 : ADI24 ADCH2 scan group 4 (SG4) end interrupt
16 : ADI30 ADCH3 scan group 0 (SG0) end interrupt
17 : ADI31 ADCH3 scan group 1 (SG1) end interrupt

Table 33.57 PIC2GTMINENx List of Function (2/3)

Function	
18	: ADI32 ADCH3 scan group 2 (SG2) end interrupt
19	: ADI33 ADCH3 scan group 3 (SG3) end interrupt
20	: ADI34 ADCH3 scan group 4 (SG4) end interrupt
21	: DSADC00 AD conversion end interrupt
22	: DSADC10 AD conversion end interrupt
23	: DSADC20 AD conversion end interrupt
24	: DSADC12 AD conversion end interrupt
25	: DSADC13 AD conversion end interrupt
26	: DSADC11 AD conversion end interrupt
27	: Reserved
28	: Reserved
29	: Reserved
30	: Reserved
31	: CADC00 AD conversion end interrupt
32	: Reserved
33	: DFE0 P/H updating trigger 0
34	: DFE0 P/H updating trigger 1
35	: DFE0 P/H updating trigger 2
36	: DFE0 P/H updating trigger 3
37	: DFE1 P/H updating trigger 0
38	: DFE1 P/H updating trigger 1
39	: DFE1 P/H updating trigger 2
40	: DFE1 P/H updating trigger 3
41	: DFE0 CH0 output data interrupt
42	: DFE0 CH1 output data interrupt
43	: DFE0 CH2 output data interrupt
44	: DFE0 CH3 output data interrupt
45	: DFE0 CH4 output data interrupt
46	: DFE0 CH5 output data interrupt
47	: DFE0 CH6 output data interrupt
48	: DFE0 CH7 output data interrupt
49	: DFE0 CH8 output data interrupt
50	: DFE0 CH9 output data interrupt
51	: DFE0 CH10 output data interrupt
52	: DFE0 CH11 output data interrupt
53	: DFE1 CH0 output data interrupt
54	: DFE1 CH1 output data interrupt
55	: DFE1 CH2 output data interrupt
56	: DFE1 CH3 output data interrupt
57	: Reserved
58	: Reserved
59	: Reserved
60	: Reserved
61	: Reserved
62	: Reserved

Table 33.57 PIC2GTMINENx List of Function (3/3)

Function	
63	: Reserved
64	: Reserved
65	: ENCA0 Encoder input (phase A)
66	: ENCA0 Encoder input (phase B)
67	: ENCA0 Encoder input (phase Z)
68	: ENCA0 Compare 0 match or Capture 0 interrupt signal
69	: ENCA0 Compare 1 match or Capture 1 interrupt signal
70	: ENCA0 Clear interrupt signal by encoder input (phase Z)
71	: ENCA0 Clock enable output
72	: ENCA0 Encoder count status output
73	: ENCA1 Encoder input (phase A)
74	: ENCA1 Encoder input (phase B)
75	: ENCA1 Encoder input (phase Z)
76	: ENCA1 Compare 0 match or Capture 0 interrupt signal
77	: ENCA1 Compare 1 match or Capture 1 interrupt signal
78	: ENCA1 Clear interrupt signal by encoder input (phase Z)
79	: ENCA1 Clock enable output
80	: ENCA1 Encoder count status output
81 to 127	: Reserved

33.3.2.41 PIC2ENCATCFG0 — Encoder Timer Trigger Output Configuration Register 0

The PIC2ENCATCFG0 register assigns timer output of GTM ([A]TOM) to PIC2ENCAnTSELj (n = 0 to 1; j = 0 to 1).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2ENCATCFG0[15:8]								PIC2ENCATCFG0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.58 PIC2ENCATCFG0 Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing to these bits, write 0.
15 to 8	PIC2ENCATCFGx [15:8]	Selects the GTM [A]TOM output as a PIC2ENCAnTSELj Bit 1 register configuration. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected. 177 + (n × 8) + m: GTM_TIM[n]_IRQ[m] ^{*3} is selected.
7 to 0	PIC2ENCATCFGx [7:0]	Selects the GTM [A]TOM output as a PIC2ENCAnTSELj Bit 0 register configuration. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected. 177 + (n × 8) + m: GTM_TIM[n]_IRQ[m] ^{*3} is selected.

Note 1. E2xFCC1: (n = 0 to 4, m = 0 to 15), E2M: (n = 0 to 3, m = 0 to 15)

Note 2. E2xFCC1: (n = 0 to 8, m = 0 to 7), E2M: (n = 0 to 7, m = 0 to 7)

Note 3. E2xFCC1: (n = 0 to 6, m = 0 to 7), E2M: (n = 0 to 6, m = 0 to 7)

33.3.2.42 PIC2ENCATCFG1 — Encoder Timer Trigger Output Configuration Register 1

The PIC2ENCATCFG1 register assigns timer output of GTM ([A]TOM) to PIC2ENCAnTSELj (n = 0 to 1; j = 0 to 1).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2ENCATCFG1[15:8]								PIC2ENCATCFG1[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.59 PIC2ENCATCFG1 Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When writing to these bits, write 0.
15 to 8	PIC2ENCATCFG1 [15:8]	Selects the GTM [A]TOM output as a PIC2ENCAnTSELj Bit 3 register configuration. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected. 177 + (n × 8) + m: GTM_TIM[n]_IRQ[m] ^{*3} is selected.
7 to 0	PIC2ENCATCFG1 [7:0]	Selects the GTM [A]TOM output as a PIC2ENCAnTSELj Bit 2 register configuration. 0: GTM signal is not selected. 1 + (n × 16) + m: GTM TOM[n]_CH[m] ^{*1} is selected. 97 + (n × 8) + m: GTM ATOM[n]_CH[m] ^{*2} is selected. 177 + (n × 8) + m: GTM_TIM[n]_IRQ[m] ^{*3} is selected.

Note 1. E2xFCC1: (n = 0 to 4, m = 0 to 15), E2M: (n = 0 to 3, m = 0 to 15)

Note 2. E2xFCC1: (n = 0 to 8, m = 0 to 7), E2M: (n = 0 to 7, m = 0 to 7)

Note 3. E2xFCC1: (n = 0 to 6, m = 0 to 7), E2M: (n = 0 to 6, m = 0 to 7)

33.3.2.43 PIC2ENCAISEN0 — Encoder Input Selection Control Register 0

The PIC2ENCAISEN0 register selects ENCA_n input signals. (n = 0 to 1)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIC2ENCAISEN0[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.60 PIC2ENCAISEN0 Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When writing to these bits, write 0.
1 to 0	PIC2ENCAISEN0[1:0]	Selects the input pins (ENCA _n E0, ENCA _n E1, ENCA _n EC) of ENCA _n timer. (n = 0 to 1) 00: Series connection 01: Parallel connection 10: Cascade connection 11: Reserved

33.3.3 Function

33.3.3.1 ADCH Trigger Selection Function

Each of 5 ADCH_n*¹ scan groups can be triggered by a hardware trigger.

Trigger signals from ATU or GTM selected by ADCH_nTSEL_m are ORed with trigger signal from external pin ADTRG before they are mapped to ADCH scan group hardware triggers.

Note 1. E2xFCC1: (n = 0 to 3), E2M: (n = 0 to 3)

NOTE

One external ADTRG pin is mapped to all ADCH scan groups.

- ATU Trigger Selection

Hardware triggers for each scan group in each ADCH_n*¹ can be generated from 16 sources of ATU timer C compare match, 16 sources of ATU timer D compare match A, 16 sources of ATU timer D compare match B, 8 sources of ATU timer G compare match, and 1 source of ATU DMA switch function. Total of 20 ADCH scan group triggers can be mapped to ATU outputs.

Note 1. E2xFCC1: (n = 0 to 3), E2M: (n = 0 to 3)

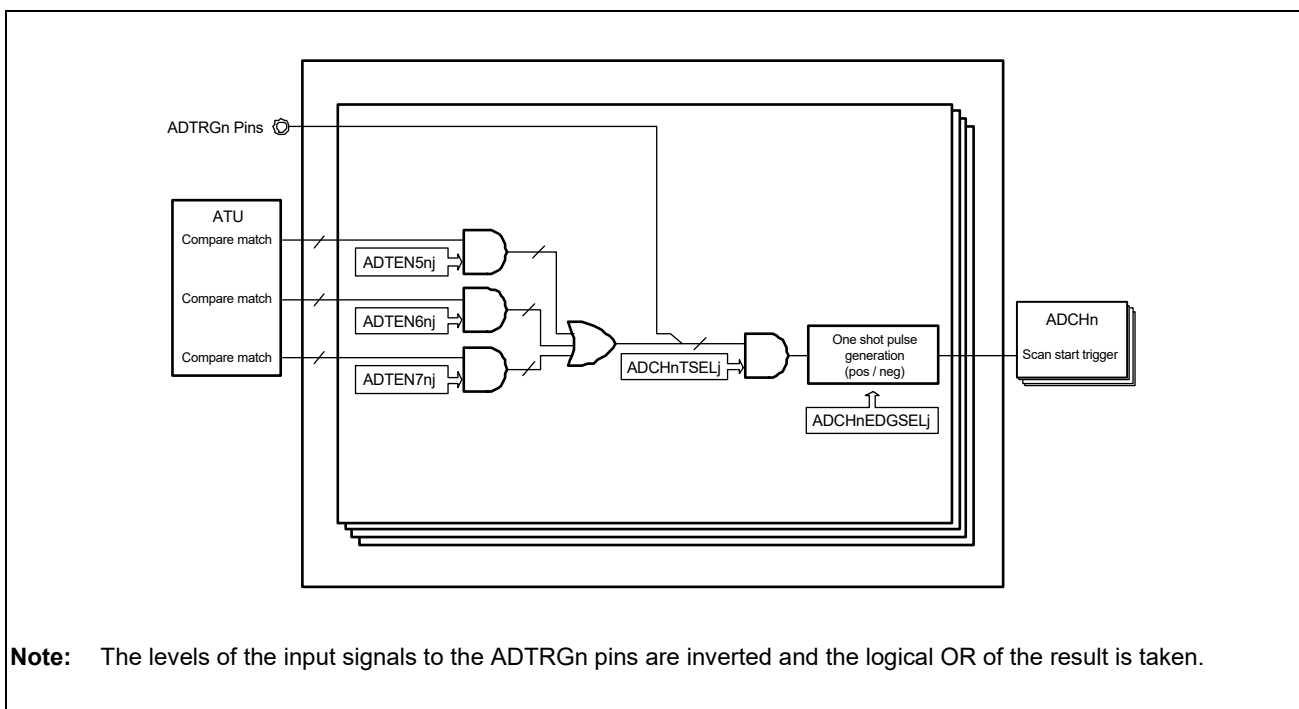


Figure 33.4 Block Diagram of Trigger Selection for ADCH_n Scan Group j in ATU Configuration

- GTM Trigger Selection

Hardware triggers for each scan groups in each ADCHn*¹ can be generated from GTM TOM[i]_CH[n]*² and ATOM[j]_CH[m]*³.

PIC2 detects only the rising edge from the above GTM signals and generates a pulse at the time. Each pulse is used as the trigger source for each ADCHn instead of the above GTM signals.

Therefore, to use falling edges of the above GTM signals to trigger the ADCHn scan group, select the rising edge detection by the PIC2ADCHnEDGSEL register, and generate a pulse at the falling edge with reference to one of the examples shown below.

- Invert the polarity of the GTM signal in the GTM, and select the inverted signal as the trigger source in the PIC2.
- Use the DTM in the GTM to generate a pulse synchronized with the falling edge of the GTM signal.

When using both rising and falling edges to trigger the ADCHn scan group, choose one of the following options:

- Generate the trigger pulses at the rising edge and the falling edge individually and merge them by using the PIC2.
- Generate the pulses at both edges by the DTM in the GTM.

Note 1. E2xFCC1: (n = 0 to 3), E2M: (n = 0 to 3)

Note 2. E2xFCC1: (i = 0 to 4, n = 0 to 15), E2M: (i = 0 to 3, n = 0 to 15)

Note 3. E2xFCC1: (j = 0 to 8, m = 0 to 7), E2M: (j = 0 to 7, m = 0 to 7)

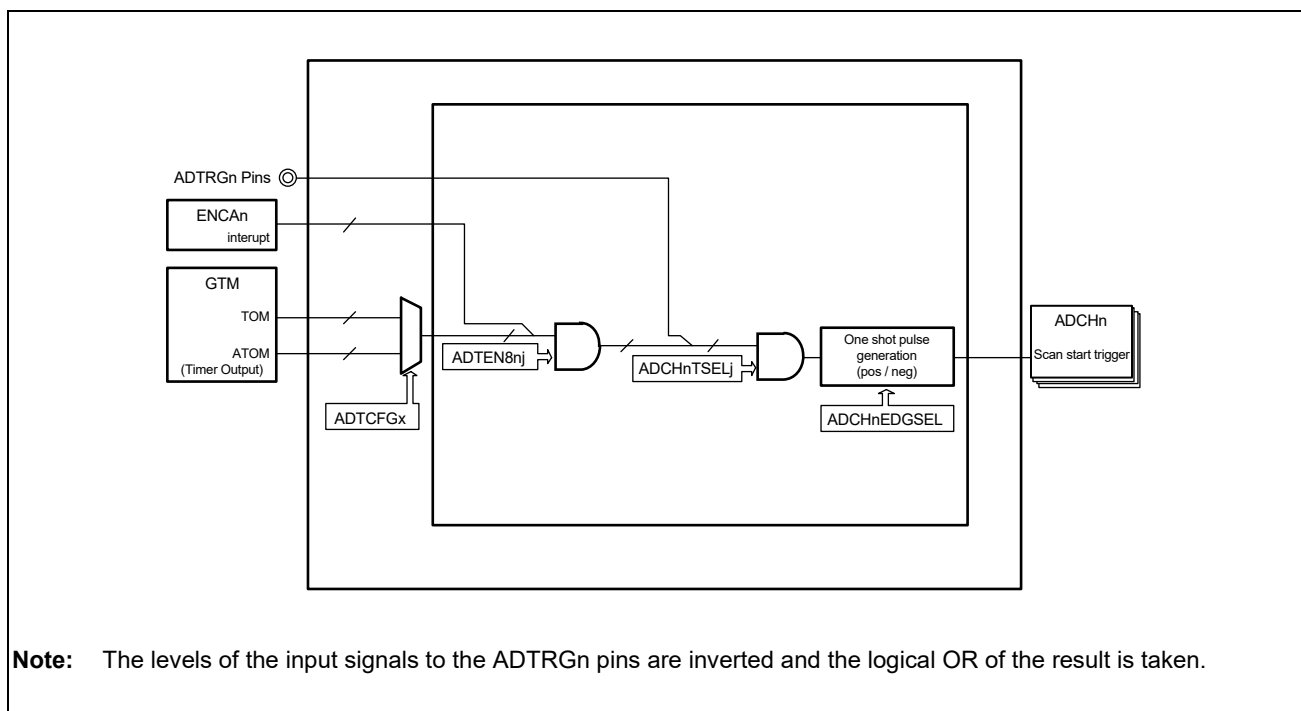


Figure 33.5 Block Diagram of Trigger Selection for ADCHn Scan Group j in GTM Configuration

33.3.3.2 DSADC Trigger Selection Function

DSADC has hardware triggers for start and stop. ATU or GTM outputs and external pin DSADTRGn are ORed before they are mapped to DSADC hardware triggers.

DSADC has hardware triggers for the read gate function. GTM outputs can be mapped to DSADC hardware read gate triggers. In this case GTM outputs are multiplexed.

- ATU trigger selection

Hardware triggers for each DSADC channel can be generated from 16 sources of ATU timer D compare match A, 16 sources of ATU timer D compare match B, and 8 sources of ATU timer G compare match.

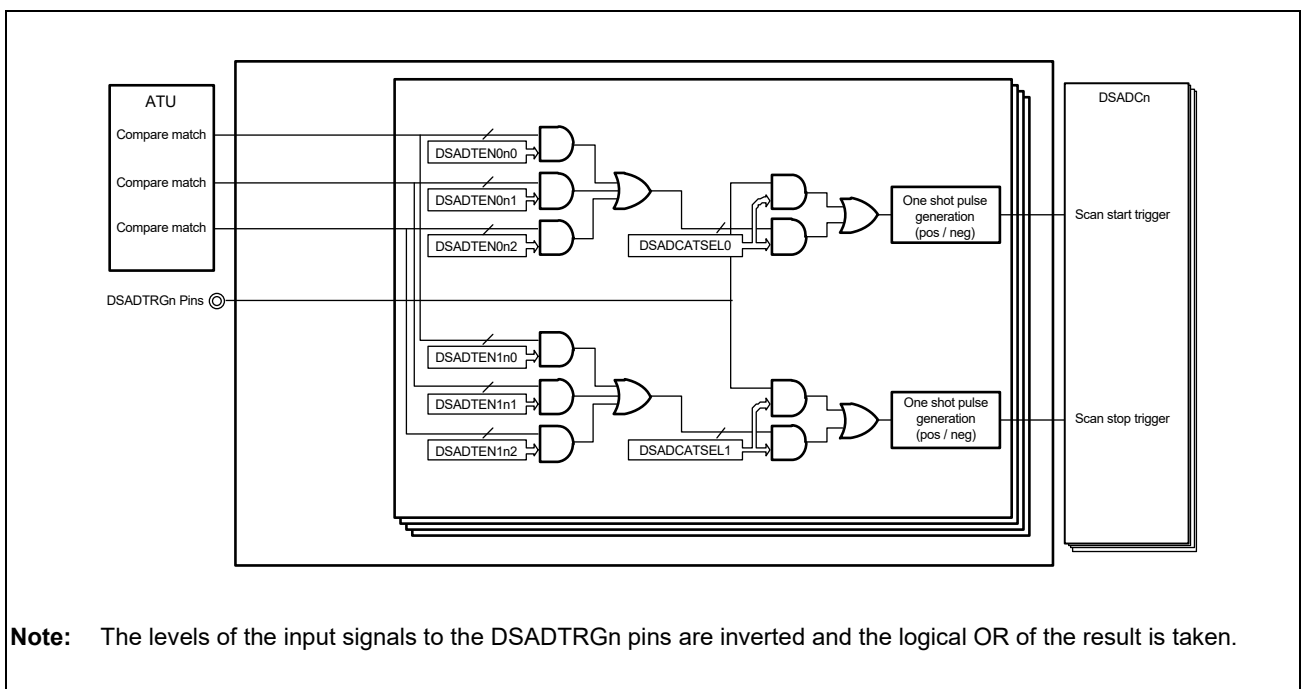


Figure 33.6 Block Diagram of Trigger Selection Function for DSADCn in ATU Configuration

• GTM trigger selection

Hardware triggers for each DSADC channel can be generated from GTM TOM[i]_CH[n]*¹ and ATOM[j]_CH[m]*².

Note 1. E2xFCC1: (i = 0 to 4, n = 0 to 15), E2M: (i = 0 to 3, n = 0 to 15)

Note 2. E2xFCC1: (j = 0 to 8, m = 0 to 7), E2M: (j = 0 to 7, m = 0 to 7)

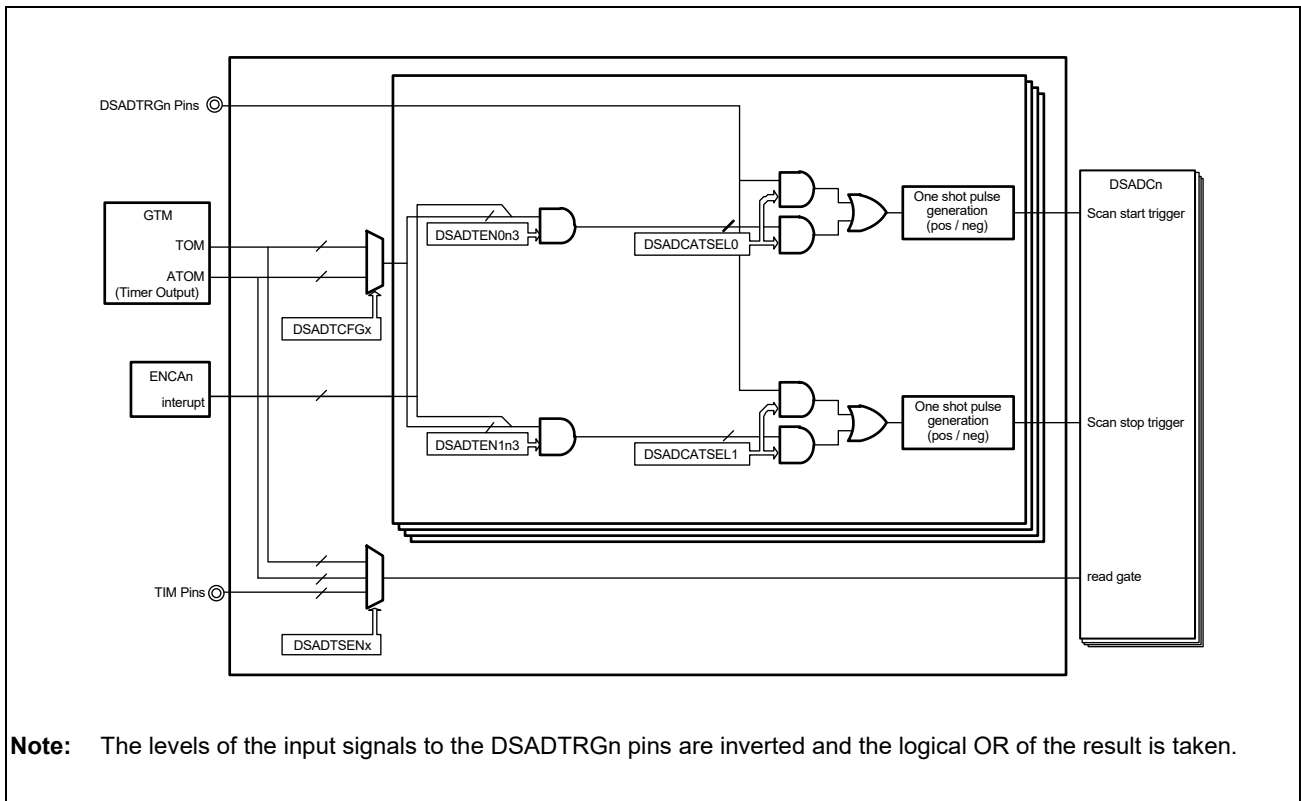


Figure 33.7 Block Diagram of Trigger Selection Function for DSADCn in GTM Configuration

33.3.3.3 CADC Trigger Selection Function

CADC has hardware triggers for start and stop. ATU or GTM signals and external pin CADTRG00 are ORed before result is input to CADC hardware triggers.

CADC has hardware triggers for the read gate function. GTM outputs can be mapped to CADC hardware read gate triggers. In this case GTM outputs are multiplexed.

- ATU trigger selection

Hardware triggers for each CADC channel can be generated from 16 sources of ATU timer D compare match A, 16 sources of ATU timer D compare match B, and 8 sources of ATU timer G compare match.

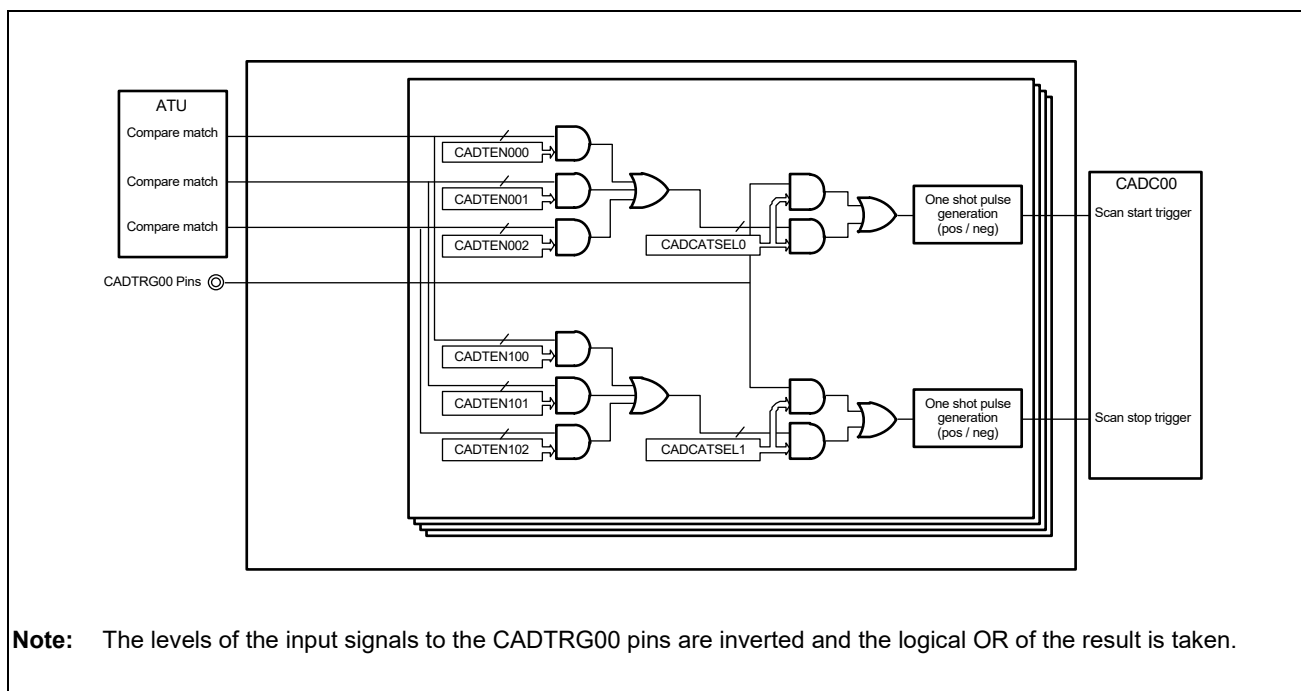


Figure 33.8 Block Diagram of Trigger Selection Function for CADC00 in ATU Configuration

• GTM Trigger selection

Hardware triggers for each CADC channel can be generated from GTM TOM[i]_CH[n]*¹ and ATOM[j]_CH[m]*²

Note 1. E2xFCC1: (i = 0 to 4, n = 0 to 15), E2M: (i = 0 to 3, n = 0 to 15)

Note 2. E2xFCC1: (j = 0 to 8, m = 0 to 7), E2M: (j = 0 to 7, m = 0 to 7)

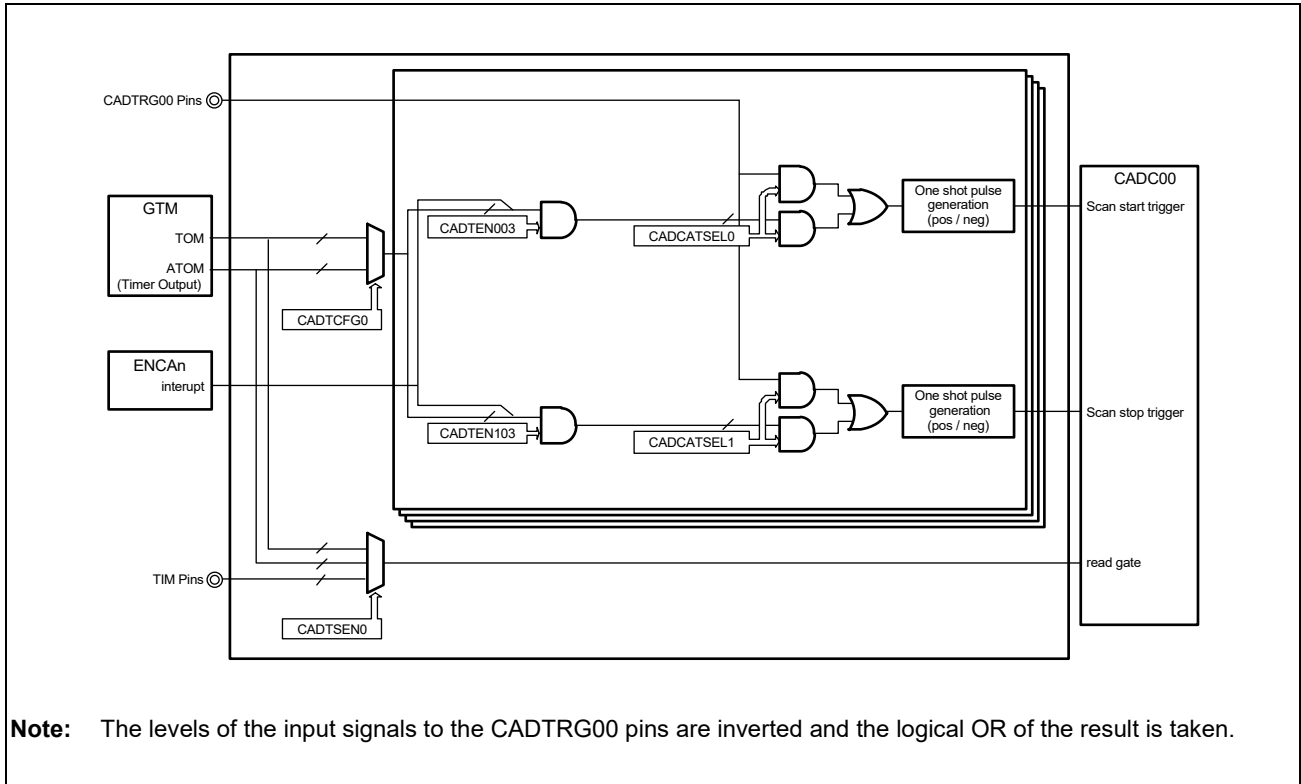


Figure 33.9 Block Diagram of Trigger Selection Function for CADC00 in GTM Configuration

33.3.3.4 ENCA Trigger Selection Function

ENCA has the trigger which captures the encoder counter value to a capture register. It's possible to map GTM output to capture trigger of ENCA. In this case GTM outputs are multiplexed.

- GTM Trigger selection

ENCA trigger can be generated from the GTM TOM[i]_CH[n]*¹, ATOM[j]_CH[m]*² and TIM[k]_IRQ[p]*³.

PIC2 detects only the rising edge from the above GTM signals and generates a pulse at the time. Each pulse is used as the trigger source for each ENCA_n instead of the above TOM and ATOM signals.

Therefore, to use falling edges of the above TOM or ATOM signals to capture the encoder counter value, select the rising edge detection by the ENCA_nIOC0 register, and generate a pulse at the falling edge with reference to one of the examples shown below.

- Invert the polarity of the TOM or ATOM signal in the GTM, and select the inverted signal as the trigger source in the PIC2.
- Use the DTM in the GTM to generate a pulse synchronized with the falling edge of the GTM signal.
- Use a TIM in the GTM, route the TOM or ATOM signal to the TIM auxiliary input (TIM_AUX_IN), and generate an interrupt pulse (TIM[k]_IRQ[p]) by detecting the falling edge of the TOM or ATOM signal with the TIM.

When using both rising and falling edges to capture the encoder counter value, choose one of the following options:

- Generate the trigger pulses at the rising edge and the falling edge individually and merge them by using the PIC2.
- Generate the pulses at both edges by the DTM from the TOM or ATOM signals.
- Generate TIM interrupts at the rising and falling edges of TOM or ATOM signals.

Note 1. E2xFCC1: (i = 0 to 4, n = 0 to 15), E2M: (i = 0 to 3, n = 0 to 15)

Note 2. E2xFCC1: (j = 0 to 8, m = 0 to 7), E2M: (j = 0 to 7, m = 0 to 7)

Note 3. E2xFCC1: (k = 0 to 7, p = 0 to 7), E2M: (k = 0 to 7, p = 0 to 7)

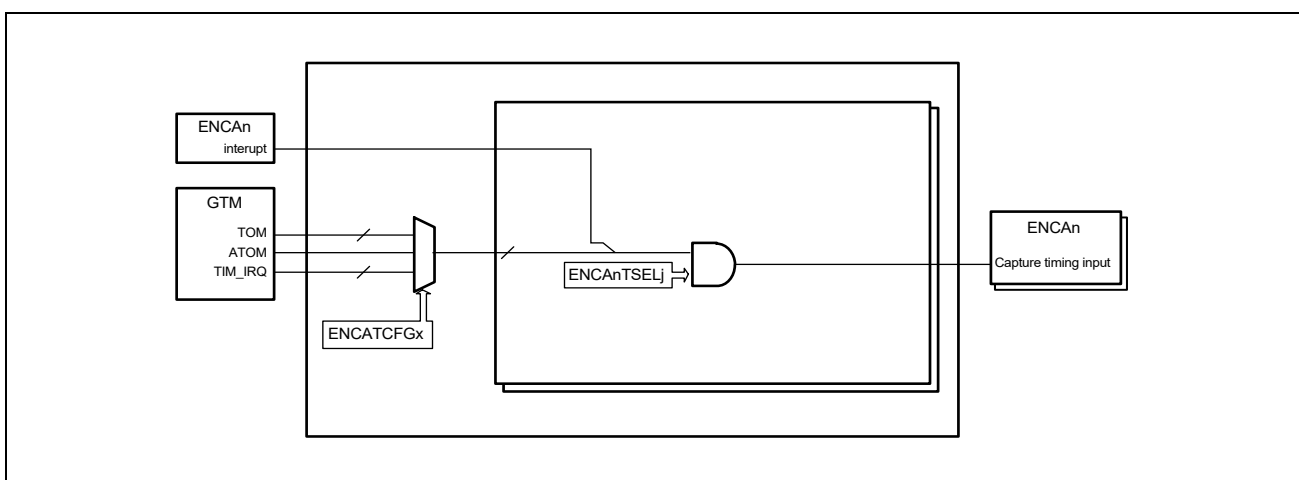


Figure 33.10 Block Diagram of Trigger Selection Function of ENCA

33.3.3.5 DFE Trigger Selection Function

DFE has a comparing timer trigger, a filter result capture trigger and an FIFO capture trigger. GTM outputs can be mapped to each DFE trigger. GTM outputs in this case are multiplexed.

- GTM Trigger selection

DFE triggers can be generated for each DFE channel from the GTM TOM[i]_CH[n]*¹ and ATOM[j]_CH[m]*²

Note 1. E2xFCC1: (i = 0 to 4, n = 0 to 15), E2M: (i = 0 to 3, n = 0 to 15)

Note 2. E2xFCC1: (j = 0 to 8, m = 0 to 7), E2M: (j = 0 to 7, m = 0 to 7)

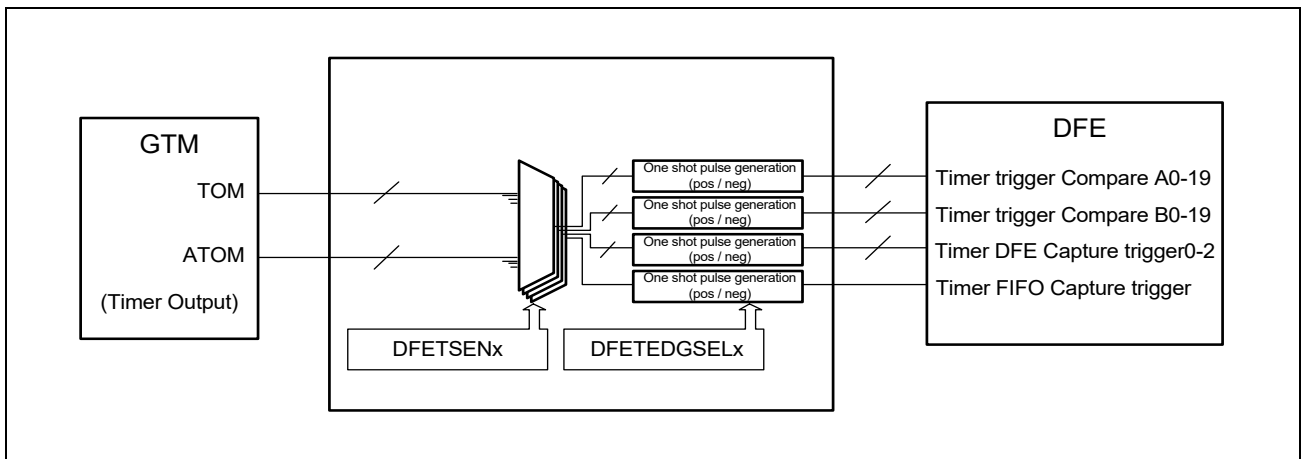


Figure 33.11 Block Diagram of Trigger Selection Function of DFE

33.3.3.6 PSI5-S Timestamp and the Sync Pulse Signal Selection Function

PSI5-S has Time stamp and Sync Pulse Generation. GTM outputs can be mapped to Time stamp (clock, enable, clear) and Sync Pulse of PSI5-S. GTM outputs in this case are multiplexed.

- GTM Trigger selection

PSI5-S time stamp trigger and sync pulse can be generated from GTM TOM[i]_CH[n]*¹ and ATOM[j]_CH[m]*²

Note 1. E2xFCC1: (i = 0 to 4, n = 0 to 15), E2M: (i = 0 to 3, n = 0 to 15)

Note 2. E2xFCC1: (j = 0 to 8, m = 0 to 7), E2M: (j = 0 to 7, m = 0 to 7)

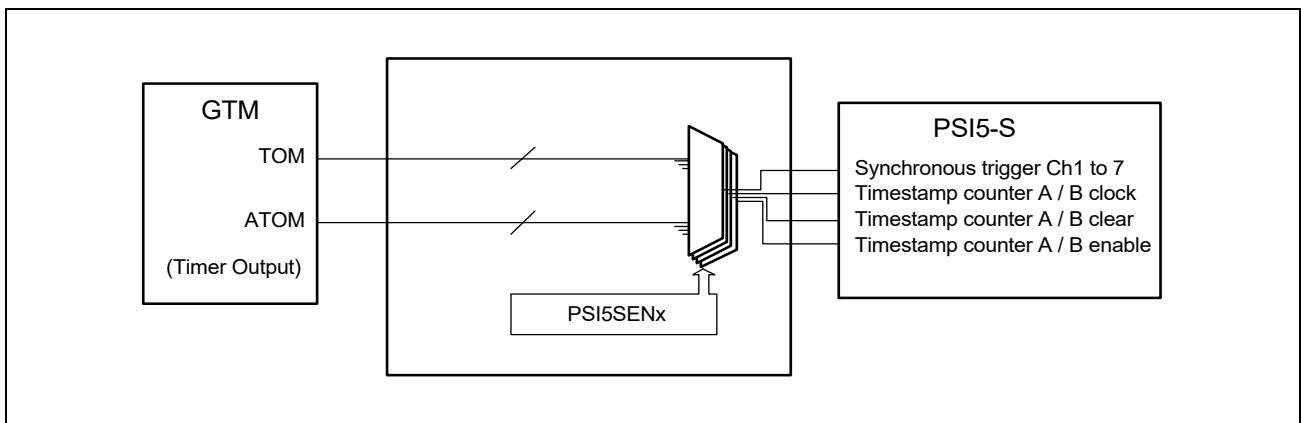


Figure 33.12 Block Diagram of Trigger Selection Function of PSI5-S

33.3.3.7 GTM Timer Input (TIM) Selection Function

Source of GTM Timer Input signals for TIM[i] (i: 0 to 7) can be selected from the following:

- PORT
It's possible to select the GTM timer input chosen in GPIO. (Defaults)
- ADCH
It's possible to select AD conversion end interrupt as timer input of GTM.
- DSADC
It's possible to select AD conversion update interrupt as timer input of GTM.
- CADC
It's possible to select AD conversion update interrupt as timer input of GTM.
- ENCA
It's possible to select count clock output and direction of rotation notifications from encoder inputs (phase A, phase B, phase Z) and compare capture interrupt and phase Z edge detection as timer input of GTM.
- DFE
It's possible to select peak hold renewal notice of a filter result as timer input of GTM.

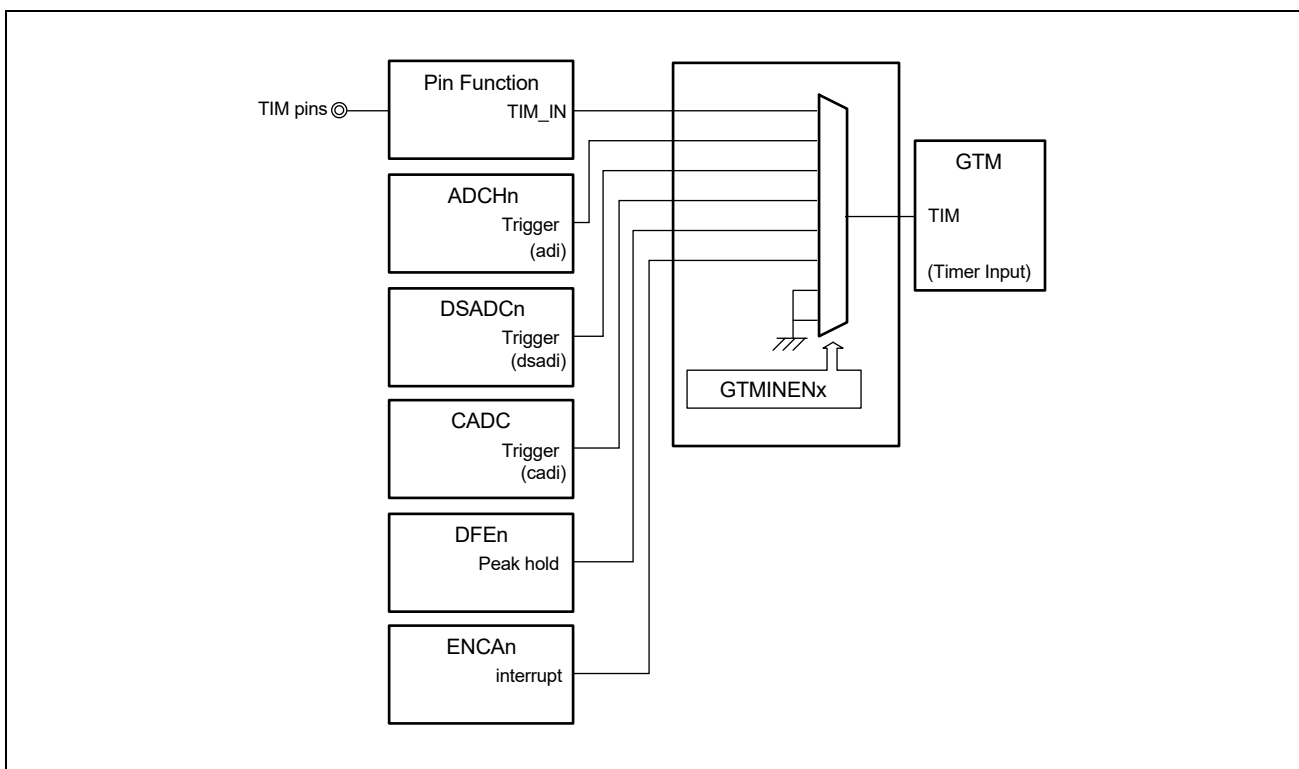


Figure 33.13 Block Diagram of Trigger Selection Function of GTM

33.3.3.8 ENCA Encoder Input Selection

It's possible to select encoder inputs of an encoder timer (phase A, phase B, phase Z) to ENCA_n.

Encoder inputs can be selected for a series, parallel or cascade connection.

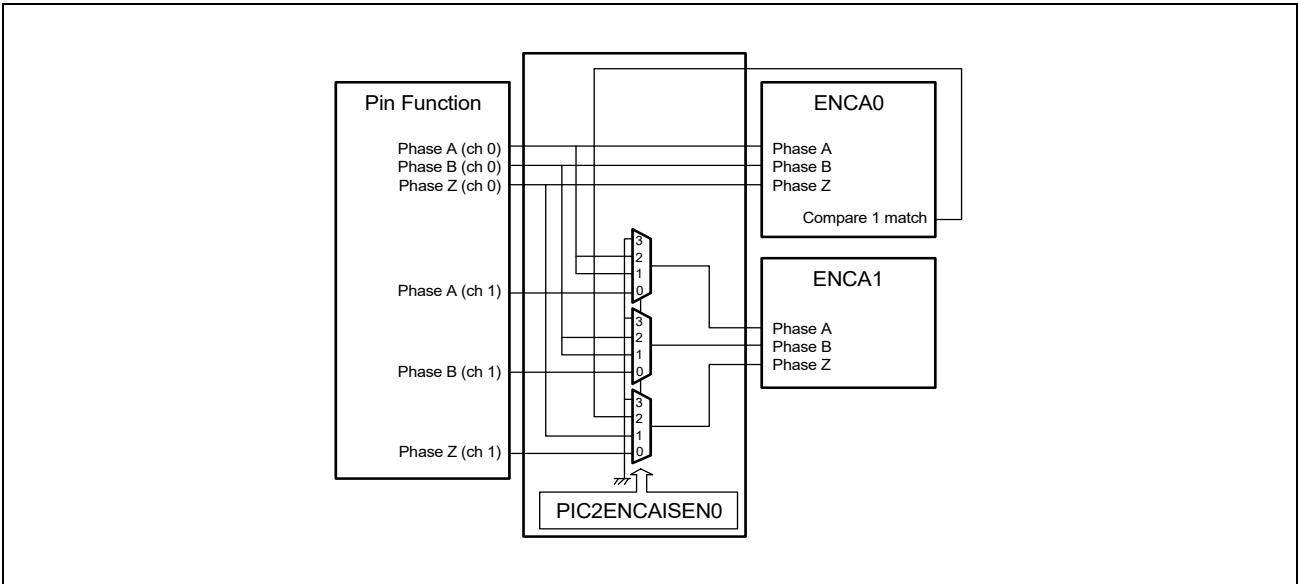


Figure 33.14 Block Diagram of ENCA Encoder Input Selection Function

Section 34 Analog to Digital Converter (ADCH)

This section contains a description of the A/D Converter (ADCH).

The first part of this section describes all of this products specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the ADCH.

34.1 Features

34.1.1 Number of Units and Channels

This microcontroller has the following number of ADCH units.

Table 34.1 Number of Units (For RH850/E2x-FCC1, E2M)

Product Name	RH850/E2x-FCC1, E2M	
	292 Pins	373 Pins
Number of ADCH Units	2	4
Name of ADCH Units	ADCHn (n = 0, 2)	ADCHn (n = 0 to 3)
Number of IFC Units	1	2
Name of IFC Units	IFCu (u = 0)	IFCu (u = 0, 1)
Number of ASF Units	1	1
Name of ASF Units	ASF _r (r = 0)	
Number of AVSEG Units	1	1
Name of AVSEG Units	AVSEG	
Number of ABFG Units	1	1
Name of ABFG Units	ABFG	
Number of AIR Units	1	1
Name of AIR Units	AIR	

An ADCH_n unit has the same number of physical channels as the number of A/D input pins and the same number of virtual channels as the number of addresses where the results of A/D conversion will be stored. The numbers of channels on individual products are listed below.

Table 34.2 Unit Configurations and Physical Channels (For RH850/E2x-FCC1, E2M)

Unit Name (Number of Channels)		RH850/E2x-FCC1, E2M	
		292 Pins	373 Pins
ADCH0	Dedicated inputs	28	20
ADCH1	Dedicated inputs	—	12
ADCH0, 1	Shared inputs	—	8
ADCH2	Dedicated inputs	24	20
ADCH3	Dedicated inputs	—	12
ADCH2, 3	Shared inputs	—	8

Table 34.3 Unit Configurations and Virtual Channels (For RH850/E2x-FCC1, E2M)

Unit Name (Number of Channels)	RH850/E2x-FCC1, E2M	
	292 Pins	373 Pins
ADCH0	40	40
ADCH1	40	40
ADCH2	40	40
ADCH3	40	40

Floating-point data register 0j and 2j of an IFC0 unit has the same number of virtual channels with ADCH0 and ADCH2.

Floating-point data register 1j and 3j of an IFC1 unit has the same number of virtual channels with ADCH1 and ADCH3.

An ASF_r unit has the same number of channels.

Table 34.4 Unit Configurations and Channels of ASF (For RH850/E2x-FCC1, E2M)

Unit Name (Number of Channels)	RH850/E2x-FCC1, E2M	
	292 Pins	373 Pins
ASF0	16	16

An ABFG unit has the same number of boundary flags.

Table 34.5 Unit Configurations and Boundary Flags of ABFG (For RH850/E2x-FCC1, E2M)

Unit Name	RH850/E2x-FCC1, E2M	
	292 Pins	373 Pins
ABFG	40	40

Table 34.6 Indices

Index	Description
n	Throughout this section, the individual ADCH units are identified by the index "n" (n = 0 to 3); for example, ADCH _n ADHALTR indicates the A/D force halt register.
p	Throughout this section, the individual physical channel groups (channel group in the unit) of ADCH _n are identified by the index "p"; for example, AN _{npq} .
q	Throughout this section, the individual physical sub channels (sub channel in the unit) of ADCH _n are identified by the index "q"; for example, AN _{npq} .
j	Throughout this section, the individual virtual channels of ADCH _n are identified by the index "j"; for example, ADCH _n VCR _j indicates the virtual channel register
x	Throughout this section, the individual scan groups (SG) of ADCH _n are identified by the index "x" (x = 0 to 4); for example, ADCH _n SGSTCR _x indicates the scan group x start control register.
y	Throughout this section, the individual A/D timers of ADCH _n are identified by the index "y" (y = 3 to 4); for example, ADCH _n ADTSTCR _y indicates the A/D timer y start control register.
u	Throughout this section, the individual IFC units are identified by the index "u" (u = 0 to 1)
r	Throughout this section, the individual ASF units are identified by the index "r" (r = 0); for example, ASF _r DR _i indicates the Accumulation data register i.
i	Throughout this section, the individual ASF units are identified by the index "i"; for example, ASF _r DR _i indicates the Accumulation data register i.
w	Throughout this section, the individual boundary flags of ABFG are identified by the index "w"; for example, ABFG _w BCR _w indicates the Input boundary flag control register w.

Table 34.7 Indices of Products (For RH850/E2x-FCC1, E2M)

Indices of Each Product	
RH850/E2x-FCC1, E2M 292 Pins	RH850/E2x-FCC1, E2M 373 Pins
p = 0 to 6 (ADCH0) —	p = 0 to 6 (ADCH0) p = 0 to 4 (ADCH1)
p = 0 to 5 (ADCH2) —	p = 0 to 6 (ADCH2) p = 0 to 4 (ADCH3)
q = 0 to 3 (ADCH0) —	q = 0 to 3 (ADCH0) q = 0 to 3 (ADCH1)
q = 0 to 3 (ADCH2) —	q = 0 to 3 (ADCH2) q = 0 to 3 (ADCH3)
j = 0 to 39 (ADCH0) —	j = 0 to 39 (ADCH0) j = 0 to 39 (ADCH1)
j = 0 to 39 (ADCH2) —	j = 0 to 39 (ADCH2) j = 0 to 39 (ADCH3)
x = 0 to 4	x = 0 to 4
y = 3 to 4	y = 3 to 4
i = 0 to 15 (ASF0)	i = 0 to 15 (ASF0)
w = 0 to 39 (ABFG)	w = 0 to 39 (ABFG)

34.1.2 Register Base Address

The ADCH_n base addresses are listed in the following table.

The ADCH_n register addresses are given as offsets from the base address.

Table 34.8 Register Base Address

Base Address Name	Base Address	Bus Group
<ADCH0_base>	FFF2 0000 _H	Peripheral Group 7
<ADCH2_base>	FFF2 1000 _H	Peripheral Group 7
<ASF0_base>	FFF2 2000 _H	Peripheral Group 7
<IFC0_base>	FFF2 3000 _H	Peripheral Group 7
<ADCH1_base>	FFF2 8000 _H	Peripheral Group 8
<ADCH3_base>	FFF2 9000 _H	Peripheral Group 8
<IFC1_base>	FFF2 A000 _H	Peripheral Group 8
<AVSEG_base>	FFF4 4000 _H	Peripheral Group 7
<ABFG_base>	FFF4 5000 _H	Peripheral Group 7
<AIR_base>	FFF4 6000 _H	Peripheral Group 7

34.1.3 Clock Supply

The ADCH_n clock supplies are shown in the following table.

Table 34.9 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
ADCH _n	ADCLK	CLK_LSB
	Register access clock	CLK_LSB
IFCu	Register access clock	CLK_LSB
ASFr	Register access clock	CLK_LSB
AVSEG	Register access clock	CLK_LSB
ABFG	Register access clock	CLK_LSB
AIR	Register access clock	CLK_LSB

34.1.4 Interrupt Requests

The ADCHn interrupt requests are listed in the following table.

Table 34.10 Interrupt Requests (1)

Interrupt symbol name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Other Trigger Signals
ADCH0						
*1	ADE0 (merged INT_UL0)	A/D error interrupt Upper/lower limit error interrupt	*1	—	—	—
*1	ADI00	Scan group 0 (SG0) end interrupt	*1	*1	*1	GTM*2/ATU
*1	ADI01	Scan group 1 (SG1) end interrupt	*1	*1	*1	GTM*2/ATU
*1	ADI02	Scan group 2 (SG2) end interrupt	*1	*1	*1	GTM*2/ATU
*1	ADI03	Scan group 3 (SG3) end interrupt	*1	*1	*1	GTM*2/ATU
*1	ADI04	Scan group 4 (SG4) end interrupt	*1	*1	*1	GTM*2/ATU
*1	ADMPX10	MPX interrupt	*1	*1	*1	—
ADCH2						
*1	ADE2 (merged INT_UL2)	A/D error interrupt Upper/lower limit error interrupt	*1	—	—	—
*1	ADI20	Scan group 0 (SG0) end interrupt	*1	*1	*1	GTM*2
*1	ADI21	Scan group 1 (SG1) end interrupt	*1	*1	*1	GTM*2
*1	ADI22	Scan group 2 (SG2) end interrupt	*1	*1	*1	GTM*2
*1	ADI23	Scan group 3 (SG3) end interrupt	*1	*1	*1	GTM*2
*1	ADI24	Scan group 4 (SG4) end interrupt	*1	*1	*1	GTM*2
*1	ADMPX12	MPX interrupt	*1	*1	*1	—

Note 1. ADE0, ADE2, ADI00 to ADI04, ADI20 to ADI24, ADMPX10 and ADMPX12 are chosen by the AIR (ADC Interrupt Router), and these are forwarded to the INTC and DMAC.

For the Interrupt Number and DMA Trigger Number, refer to **Table 34.15** to **Table 34.28**.

Note 2. ADI00 to ADI04 and ADI20 to ADI24 are chosen by the PIC (Peripheral Interconnection), and these are forwarded to the GTM. For details, refer to **Section 33, Peripheral Interconnection (PIC)**.

Table 34.11 Interrupt Requests (2)

Interrupt symbol name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Other Trigger Signals
ADCH1						
*1	ADE1 (merged INT_UL1)	A/D error interrupt Upper/lower limit error interrupt	*1	—	—	—
*1	ADI10	Scan group 0 (SG0) end interrupt	*1	*1	*1	GTM*2
*1	ADI11	Scan group 1 (SG1) end interrupt	*1	*1	*1	GTM*2
*1	ADI12	Scan group 2 (SG2) end interrupt	*1	*1	*1	GTM*2
*1	ADI13	Scan group 3 (SG3) end interrupt	*1	*1	*1	GTM*2
*1	ADI14	Scan group 4 (SG4) end interrupt	*1	*1	*1	GTM*2
*1	ADMPXI1	MPX interrupt	*1	*1	*1	—
ADCH3						
*1	ADE3 (merged INT_UL3)	A/D error interrupt Upper/lower limit error interrupt	*1	—	—	—
*1	ADI30	Scan group 0 (SG0) end interrupt	*1	*1	*1	GTM*2
*1	ADI31	Scan group 1 (SG1) end interrupt	*1	*1	*1	GTM*2
*1	ADI32	Scan group 2 (SG2) end interrupt	*1	*1	*1	GTM*2
*1	ADI33	Scan group 3 (SG3) end interrupt	*1	*1	*1	GTM*2
*1	ADI34	Scan group 4 (SG4) end interrupt	*1	*1	*1	GTM*2
*1	ADMPXI3	MPX interrupt	*1	*1	*1	—

Note 1. ADE1, ADE3, ADI10 to ADI14, ADI30 to ADI34, ADMPXI1 and ADMPXI3 are chosen by the AIR (ADC Interrupt Router), and these are forwarded to the INTC and DMAC.

For the Interrupt Number and DMA Trigger Number, refer to **Table 34.15** to **Table 34.28**.

Note 2. ADI10 to ADI14 and ADI30 to ADI34 are chosen by the PIC (Peripheral Interconnection), and these are forwarded to the GTM. For details, refer to **Section 33, Peripheral Interconnection (PIC)**.

Table 34.12 Interrupt Requests (3)

Interrupt symbol name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Other Trigger Signals
ASF0						
*1	ASI0	Ch0 integration end interrupt	*1	*1	*1	—
*1	ASI1	Ch1 integration end interrupt	*1	*1	*1	—
*1	ASI2	Ch2 integration end interrupt	*1	*1	*1	—
*1	ASI3	Ch3 integration end interrupt	*1	*1	*1	—
*1	ASI4	Ch4 integration end interrupt	*1	*1	*1	—
*1	ASI5	Ch5 integration end interrupt	*1	*1	*1	—
*1	ASI6	Ch6 integration end interrupt	*1	*1	*1	—
*1	ASI7	Ch7 integration end interrupt	*1	*1	*1	—
*1	ASI8	Ch8 integration end interrupt	*1	*1	*1	—
*1	ASI9	Ch9 integration end interrupt	*1	*1	*1	—
*1	ASI10	Ch10 integration end interrupt	*1	*1	*1	—
*1	ASI11	Ch11 integration end interrupt	*1	*1	*1	—
*1	ASI12	Ch12 integration end interrupt	*1	*1	*1	—
*1	ASI13	Ch13 integration end interrupt	*1	*1	*1	—
*1	ASI14	Ch14 integration end interrupt	*1	*1	*1	—
*1	ASI15	Ch15 integration end interrupt	*1	*1	*1	—

Note 1. ASI0 to ASI15 are chosen by the AIR (ADC Interrupt Router), and these are forwarded to the INTC and DMAC.
For the Interrupt Number and DMA Trigger Number, refer to **Table 34.15** to **Table 34.28**.

Table 34.13 Interrupt Requests (4)

Interrupt symbol name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Other Trigger Signals
ABFG						
*1	BFP0	Boundary flag pulse 0 interrupt	*1	*1	*1	—
*1	BFP1	Boundary flag pulse 1 interrupt	*1	*1	*1	—
*1	BFP2	Boundary flag pulse 2 interrupt	*1	*1	*1	—
*1	BFP3	Boundary flag pulse 3 interrupt	*1	*1	*1	—
*1	BFP4	Boundary flag pulse 4 interrupt	*1	*1	*1	—
*1	BFP5	Boundary flag pulse 5 interrupt	*1	*1	*1	—
*1	BFP6	Boundary flag pulse 6 interrupt	*1	*1	*1	—
*1	BFP7	Boundary flag pulse 7 interrupt	*1	*1	*1	—
*1	BFP8	Boundary flag pulse 8 interrupt	*1	*1	*1	—
*1	BFP9	Boundary flag pulse 9 interrupt	*1	*1	*1	—
*1	BFP10	Boundary flag pulse 10 interrupt	*1	*1	*1	—
*1	BFP11	Boundary flag pulse 11 interrupt	*1	*1	*1	—
*1	BFP12	Boundary flag pulse 12 interrupt	*1	*1	*1	—
*1	BFP13	Boundary flag pulse 13 interrupt	*1	*1	*1	—
*1	BFP14	Boundary flag pulse 14 interrupt	*1	*1	*1	—
*1	BFP15	Boundary flag pulse 15 interrupt	*1	*1	*1	—
*1	BFP16	Boundary flag pulse 16 interrupt	*1	*1	*1	—
*1	BFP17	Boundary flag pulse 17 interrupt	*1	*1	*1	—
*1	BFP18	Boundary flag pulse 18 interrupt	*1	*1	*1	—
*1	BFP19	Boundary flag pulse 19 interrupt	*1	*1	*1	—
*1	BFP20	Boundary flag pulse 20 interrupt	*1	*1	*1	—
*1	BFP21	Boundary flag pulse 21 interrupt	*1	*1	*1	—
*1	BFP22	Boundary flag pulse 22 interrupt	*1	*1	*1	—
*1	BFP23	Boundary flag pulse 23 interrupt	*1	*1	*1	—
*1	BFP24	Boundary flag pulse 24 interrupt	*1	*1	*1	—
*1	BFP25	Boundary flag pulse 25 interrupt	*1	*1	*1	—

Note 1. The BFPw are chosen by the AIR (ADC Interrupt Router), and these are forwarded to the INTC and DMAC.
For the Interrupt Number and DMA Trigger Number, refer to **Table 34.15** to **Table 34.28**.

Table 34.14 Interrupt Requests (5)

Interrupt symbol name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Other Trigger Signals
*1	BFP26	Boundary flag pulse 26 interrupt	*1	*1	*1	—
*1	BFP27	Boundary flag pulse 27 interrupt	*1	*1	*1	—
*1	BFP28	Boundary flag pulse 28 interrupt	*1	*1	*1	—
*1	BFP29	Boundary flag pulse 29 interrupt	*1	*1	*1	—
*1	BFP30	Boundary flag pulse 30 interrupt	*1	*1	*1	—
*1	BFP31	Boundary flag pulse 31 interrupt	*1	*1	*1	—
*1	BFP32	Boundary flag pulse 32 interrupt	*1	*1	*1	—
*1	BFP33	Boundary flag pulse 33 interrupt	*1	*1	*1	—
*1	BFP34	Boundary flag pulse 34 interrupt	*1	*1	*1	—
*1	BFP35	Boundary flag pulse 35 interrupt	*1	*1	*1	—
*1	BFP36	Boundary flag pulse 36 interrupt	*1	*1	*1	—
*1	BFP37	Boundary flag pulse 37 interrupt	*1	*1	*1	—
*1	BFP38	Boundary flag pulse 38 interrupt	*1	*1	*1	—
*1	BFP39	Boundary flag pulse 39 interrupt	*1	*1	*1	—

Note 1. The BFPw are chosen by the AIR (ADC Interrupt Router), and those are forwarded to the INTC and DMAC.
For the Interrupt Number and DMA Trigger Number, refer to **Table 34.15** to **Table 34.28**.

Table 34.15 Interrupt Requests (6)

Interrupt symbol name	Unit Interrupt Signal	Outline			Interrupt Number	sDMA Trigger Number	DTS Trigger Number
		Resource 0	Resource 1	Initial resource number			
AIR (Interrupt request)							
INTAIRINTREQ0	INTAIRINTREQ0	ADMPXI0	—	0	245	—	—
INTAIRINTREQ1	INTAIRINTREQ1	ADMPXI2	—	0	246	—	—
INTAIRINTREQ2	INTAIRINTREQ2	ADMPXI1	BFP18	0	247	—	—
INTAIRINTREQ3	INTAIRINTREQ3	ADMPXI3	BFP19	0	248	—	—
INTAIRINTREQ4	INTAIRINTREQ4	ADI00	—	0	249	—	—
INTAIRINTREQ5	INTAIRINTREQ5	ADI01	—	0	250	—	—
INTAIRINTREQ6	INTAIRINTREQ6	ADI02	—	0	251	—	—
INTAIRINTREQ7	INTAIRINTREQ7	ADI03	—	0	252	—	—
INTAIRINTREQ8	INTAIRINTREQ8	ADI04	—	0	253	—	—
INTAIRINTREQ9	INTAIRINTREQ9	ADI20	—	0	254	—	—
INTAIRINTREQ10	INTAIRINTREQ10	ADI21	—	0	255	—	—
INTAIRINTREQ11	INTAIRINTREQ11	ADI22	—	0	256	—	—
INTAIRINTREQ12	INTAIRINTREQ12	ADI23	—	0	257	—	—
INTAIRINTREQ13	INTAIRINTREQ13	ADI24	—	0	258	—	—
INTAIRINTREQ14	INTAIRINTREQ14	ADI10	BFP20	0	259	—	—
INTAIRINTREQ15	INTAIRINTREQ15	ADI11	BFP21	0	260	—	—

Table 34.16 Interrupt Requests (7)

Interrupt symbol name	Unit Interrupt Signal	Outline			Interrupt Number	sDMA Trigger Number	DTS Trigger Number
		Resource 0	Resource 1	Initial resource number			
INTAIRINTREQ16	INTAIRINTREQ16	ADI12	BFP22	0	261	—	—
INTAIRINTREQ17	INTAIRINTREQ17	ADI13	BFP23	0	262	—	—
INTAIRINTREQ18	INTAIRINTREQ18	ADI14	BFP24	0	263	—	—
INTAIRINTREQ19	INTAIRINTREQ19	ADI30	BFP25	0	264	—	—
INTAIRINTREQ20	INTAIRINTREQ20	ADI31	BFP26	0	265	—	—
INTAIRINTREQ21	INTAIRINTREQ21	ADI32	BFP27	0	266	—	—
INTAIRINTREQ22	INTAIRINTREQ22	ADI33	BFP28	0	267	—	—

Table 34.17 Interrupt Requests (8)

Interrupt symbol name	Unit Interrupt Signal	Outline			Interrupt Number	sDMA Trigger Number	DTS Trigger Number
		Resource 0	Resource 1	Initial resource number			
INTAIRINTREQ23	INTAIRINTREQ23	ADI34	BFP29	0	268	—	—
INTAIRINTREQ24	INTAIRINTREQ24	ADE0 (merged INT_UL0)		0	269	—	—
INTAIRINTREQ25	INTAIRINTREQ25	ADE2 (merged INT_UL2)		0	270	—	—
INTAIRINTREQ26	INTAIRINTREQ26	ADE1 (merged INT_UL1)	BFP13	0	271	—	—
INTAIRINTREQ27	INTAIRINTREQ27	ADE3 (merged INT_UL3)	BFP14	0	272	—	—
INTAIRINTREQ28	INTAIRINTREQ28	DSADE00* ³	—	0	273	—	—
INTAIRINTREQ29	INTAIRINTREQ29	DSADE10* ³	—	0	274	—	—
INTAIRINTREQ30	INTAIRINTREQ30	DSADE20* ³	—	0	275	—	—

Note 3. For details of these signals, refer to **Section 35, Delta-Sigma Analog to Digital Converter (DSADC)**.

Table 34.18 Interrupt Requests (9)

Interrupt symbol name	Unit Interrupt Signal	Outline			Interrupt Number	sDMA Trigger Number	DTS Trigger Number
		Resource 0	Resource 1	Initial resource number			
INTAIRINTREQ31	INTAIRINTREQ31	DSADE12* ³	BFP15	0	276	—	—
INTAIRINTREQ32	INTAIRINTREQ32	DSADE13* ³	BFP16	0	277	—	—
INTAIRINTREQ33	INTAIRINTREQ33	DSADE11* ³	BFP17	0	278	—	—
INTAIRINTREQ34	INTAIRINTREQ34	—	BFP9	0	279	—	—
INTAIRINTREQ35	INTAIRINTREQ35	—	BFP10	0	280	—	—
INTAIRINTREQ36	INTAIRINTREQ36	—	BFP11	0	281	—	—
INTAIRINTREQ37	INTAIRINTREQ37	—	BFP12	0	282	—	—
INTAIRINTREQ38	INTAIRINTREQ38	CADE00* ⁴	—	0	283	—	—
INTAIRINTREQ39	INTAIRINTREQ39	ASI0	BFP30	0	284	—	—

Note 3. For details of these signals, refer to **Section 35, Delta-Sigma Analog to Digital Converter (DSADC)**.

Note 4. For details of these signals, refer to **Section 36, Cyclic Analog to Digital Converter (CADC)**.

Table 34.19 Interrupt Requests (10)

Interrupt symbol name	Unit Interrupt Signal	Outline			Interrupt Number	sDMA Trigger Number	DTS Trigger Number
		Resource 0	Resource 1	Initial resource number			
INTAIRINTREQ40	INTAIRINTREQ40	ASI1	BFP31	0	285	—	—
INTAIRINTREQ41	INTAIRINTREQ41	ASI2	BFP32	0	286	—	—
INTAIRINTREQ42	INTAIRINTREQ42	ASI3	BFP33	0	287	—	—
INTAIRINTREQ43	INTAIRINTREQ43	ASI4	BFP34	0	288	—	—
INTAIRINTREQ44	INTAIRINTREQ44	ASI5	BFP35	0	289	—	—

Table 34.20 Interrupt Requests (11)

Interrupt symbol name	Unit Interrupt Signal	Outline			Interrupt Number	sDMA Trigger Number	DTS Trigger Number
		Resource 0	Resource 1	Initial resource number			
INTAIRINTREQ45	INTAIRINTREQ45	ASI6	BFP36	0	290	—	—
INTAIRINTREQ46	INTAIRINTREQ46	ASI7	BFP37	0	291	—	—
INTAIRINTREQ47	INTAIRINTREQ47	ASI8	BFP38	0	292	—	—
INTAIRINTREQ48	INTAIRINTREQ48	ASI9	BFP39	0	293	—	—
INTAIRINTREQ49	INTAIRINTREQ49	ASI10	—	0	294	—	—
INTAIRINTREQ50	INTAIRINTREQ50	ASI11	—	0	295	—	—
INTAIRINTREQ51	INTAIRINTREQ51	ASI12	—	0	296	—	—
INTAIRINTREQ52	INTAIRINTREQ52	ASI13	—	0	297	—	—

Table 34.21 Interrupt Requests (12)

Interrupt symbol name	Unit Interrupt Signal	Outline			Interrupt Number	sDMA Trigger Number	DTS Trigger Number
		Resource 0	Resource 1	Initial resource number			
INTAIRINTREQ53	INTAIRINTREQ53	ASI14	—	0	298	—	—
INTAIRINTREQ54	INTAIRINTREQ54	ASI15	—	0	299	—	—
INTAIRINTREQ55	INTAIRINTREQ55	—	BFP0	1	300	—	—
INTAIRINTREQ56	INTAIRINTREQ56	—	BFP1	1	301	—	—
INTAIRINTREQ57	INTAIRINTREQ57	—	BFP2	1	302	—	—
INTAIRINTREQ58	INTAIRINTREQ58	—	BFP3	1	303	—	—
INTAIRINTREQ59	INTAIRINTREQ59	—	BFP4	1	304	—	—
INTAIRINTREQ60	INTAIRINTREQ60	—	BFP5	1	305	—	—
INTAIRINTREQ61	INTAIRINTREQ61	—	BFP6	1	306	—	—
INTAIRINTREQ62	INTAIRINTREQ62	—	BFP7	1	307	—	—
INTAIRINTREQ63	INTAIRINTREQ63	—	BFP8	1	308	—	—

Table 34.22 Interrupt Requests (13)

Interrupt symbol name	Unit Interrupt Signal	Outline			Interrupt Number	sDMA Trigger Number	DTS Trigger Number
		Resource 0	Resource 1	Initial resource number			
AIR (DMA request)							
INTAIRDMAREQ0	INTAIRDMAREQ0	ADMPXI0	—	0	—	group0-0	group0-0
INTAIRDMAREQ1	INTAIRDMAREQ1	ADMPXI2	—	0	—	group0-1	group0-1
INTAIRDMAREQ2	INTAIRDMAREQ2	ADMPXI1	BFP18	0	—	group0-2	group0-2
INTAIRDMAREQ3	INTAIRDMAREQ3	ADMPXI3	BFP19	0	—	group0-3	group0-3
INTAIRDMAREQ4	INTAIRDMAREQ4	ADI00	—	0	—	group0-4	group0-4
INTAIRDMAREQ5	INTAIRDMAREQ5	ADI01	—	0	—	group0-5	group0-5
INTAIRDMAREQ6	INTAIRDMAREQ6	ADI02	—	0	—	group0-6	group0-6
INTAIRDMAREQ7	INTAIRDMAREQ7	ADI03	—	0	—	group0-7	group0-7
INTAIRDMAREQ8	INTAIRDMAREQ8	ADI04	—	0	—	group0-8	group0-8
INTAIRDMAREQ9	INTAIRDMAREQ9	ADI20	—	0	—	group0-9	group0-9
INTAIRDMAREQ10	INTAIRDMAREQ10	ADI21	—	0	—	group0-10	group0-10
INTAIRDMAREQ11	INTAIRDMAREQ11	ADI22	—	0	—	group0-11	group0-11
INTAIRDMAREQ12	INTAIRDMAREQ12	ADI23	—	0	—	group0-12	group0-12
INTAIRDMAREQ13	INTAIRDMAREQ13	ADI24	—	0	—	group0-13	group0-13
INTAIRDMAREQ14	INTAIRDMAREQ14	ADI10	BFP20	0	—	group0-14	group0-14
INTAIRDMAREQ15	INTAIRDMAREQ15	ADI11	BFP21	0	—	group0-15	group0-15

Table 34.23 Interrupt Requests (14)

Interrupt symbol name	Unit Interrupt Signal	Outline			Interrupt Number	sDMA Trigger Number	DTS Trigger Number
		Resource 0	Resource 1	Initial resource number			
INTAIRDMAREQ16	INTAIRDMAREQ16	ADI12	BFP22	0	—	group0-16	group0-16
INTAIRDMAREQ17	INTAIRDMAREQ17	ADI13	BFP23	0	—	group0-17	group0-17
INTAIRDMAREQ18	INTAIRDMAREQ18	ADI14	BFP24	0	—	group0-18	group0-18
INTAIRDMAREQ19	INTAIRDMAREQ19	ADI30	BFP25	0	—	group0-19	group0-19
INTAIRDMAREQ20	INTAIRDMAREQ20	ADI31	BFP26	0	—	group0-20	group0-20
INTAIRDMAREQ21	INTAIRDMAREQ21	ADI32	BFP27	0	—	group0-21	group0-21

Table 34.24 Interrupt Requests (15)

Interrupt symbol name	Unit Interrupt Signal	Outline			Interrupt Number	sDMA Trigger Number	DTS Trigger Number
		Resource 0	Resource 1	Initial resource number			
INTAIRDMAREQ22	INTAIRDMAREQ22	ADI33	BFP28	0	—	group0-22	group0-22
INTAIRDMAREQ23	INTAIRDMAREQ23	ADI34	BFP29	0	—	group0-23	group0-23
INTAIRDMAREQ24	INTAIRDMAREQ24	—	BFP9	1	—	group0-24	group0-24
INTAIRDMAREQ25	INTAIRDMAREQ25	—	BFP10	1	—	group0-25	group0-25
INTAIRDMAREQ26	INTAIRDMAREQ26	—	BFP11	1	—	group0-26	group0-26
INTAIRDMAREQ27	INTAIRDMAREQ27	—	BFP12	1	—	group0-27	group0-27
INTAIRDMAREQ28	INTAIRDMAREQ28	DSADI00*3	—	0	—	group0-28	group0-28
INTAIRDMAREQ29	INTAIRDMAREQ29	DSADI10*3	—	0	—	group0-29	group0-29
INTAIRDMAREQ30	INTAIRDMAREQ30	DSADI20*3	—	0	—	group0-30	group0-30

Note 3. For details of signals, refer to **Section 35, Delta-Sigma Analog to Digital Converter (DSADC)**.

Table 34.25 Interrupt Requests (16)

Interrupt symbol name	Unit Interrupt Signal	Outline			Interrupt Number	sDMA Trigger Number	DTS Trigger Number
		Resource 0	Resource 1	Initial resource number			
INTAIRDMAREQ31	INTAIRDMAREQ31	DSADI12*3	BFP17	0	—	group0-31	group0-31
INTAIRDMAREQ32	INTAIRDMAREQ32	DSADI13*3	—	0	—	group0-32	group0-32
INTAIRDMAREQ33	INTAIRDMAREQ33	DSADI11*3	—	0	—	group0-33	group0-33
INTAIRDMAREQ34	INTAIRDMAREQ34	—	BFP13	0	—	group0-34	group0-34
INTAIRDMAREQ35	INTAIRDMAREQ35	—	BFP14	0	—	group0-35	group0-35
INTAIRDMAREQ36	INTAIRDMAREQ36	—	BFP15	0	—	group0-36	group0-36
INTAIRDMAREQ37	INTAIRDMAREQ37	—	BFP16	0	—	group0-37	group0-37
INTAIRDMAREQ38	INTAIRDMAREQ38	CADI00*4	—	0	—	group0-38	group0-38
INTAIRDMAREQ39	INTAIRDMAREQ39	ASI0	BFP30	0	—	group0-39	group0-39

Note 3. For details of signals, refer to **Section 35, Delta-Sigma Analog to Digital Converter (DSADC)**.

Note 4. For details of signals, refer to **Section 36, Cyclic Analog to Digital Converter (CADC)**.

Table 34.26 Interrupt Requests (17)

Interrupt symbol name	Unit Interrupt Signal	Outline			Interrupt Number	sDMA Trigger Number	DTS Trigger Number
		Resource 0	Resource 1	Initial resource number			
INTAIRDMAREQ40	INTAIRDMAREQ40	ASI1	BFP31	0	—	group0-40	group0-40
INTAIRDMAREQ41	INTAIRDMAREQ41	ASI2	BFP32	0	—	group0-41	group0-41
INTAIRDMAREQ42	INTAIRDMAREQ42	ASI3	BFP33	0	—	group0-42	group0-42
INTAIRDMAREQ43	INTAIRDMAREQ43	ASI4	BFP34	0	—	group0-43	group0-43
INTAIRDMAREQ44	INTAIRDMAREQ44	ASI5	BFP35	0	—	group0-44	group0-44

Table 34.27 Interrupt Requests (18)

Interrupt symbol name	Unit Interrupt Signal	Outline			Interrupt Number	sDMA Trigger Number	DTS Trigger Number
		Resource 0	Resource 1	Initial resource number			
INTAIRDMAREQ45	INTAIRDMAREQ45	ASI6	BFP36	0	—	group0-45	group0-45
INTAIRDMAREQ46	INTAIRDMAREQ46	ASI7	BFP37	0	—	group0-46	group0-46
INTAIRDMAREQ47	INTAIRDMAREQ47	ASI8	BFP38	0	—	group0-47	group0-47
INTAIRDMAREQ48	INTAIRDMAREQ48	ASI9	BFP39	0	—	group0-48	group0-48
INTAIRDMAREQ49	INTAIRDMAREQ49	ASI10	—	0	—	group0-49	group0-49
INTAIRDMAREQ50	INTAIRDMAREQ50	ASI11	—	0	—	group0-50	group0-50
INTAIRDMAREQ51	INTAIRDMAREQ51	ASI12	—	0	—	group0-51	group0-51
INTAIRDMAREQ52	INTAIRDMAREQ52	ASI13	—	0	—	group0-52	group0-52

Table 34.28 Interrupt Requests (19)

Interrupt symbol name	Unit Interrupt Signal	Outline			Interrupt Number	sDMA Trigger Number	DTS Trigger Number
		Resource 0	Resource 1	Initial resource number			
INTAIRDMAREQ53	INTAIRDMAREQ53	ASI14	—	0	—	group0-53	group0-53
INTAIRDMAREQ54	INTAIRDMAREQ54	ASI15	—	0	—	group0-54	group0-54
INTAIRDMAREQ55	INTAIRDMAREQ55	—	BFP0	1	—	group0-55	group0-55
INTAIRDMAREQ56	INTAIRDMAREQ56	—	BFP1	1	—	group0-56	group0-56
INTAIRDMAREQ57	INTAIRDMAREQ57	—	BFP2	1	—	group0-57	group0-57
INTAIRDMAREQ58	INTAIRDMAREQ58	—	BFP3	1	—	group0-58	group0-58
INTAIRDMAREQ59	INTAIRDMAREQ59	—	BFP4	1	—	group0-59	group0-59
INTAIRDMAREQ60	INTAIRDMAREQ60	—	BFP5	1	—	group0-60	group0-60
INTAIRDMAREQ61	INTAIRDMAREQ61	—	BFP6	1	—	group0-61	group0-61
INTAIRDMAREQ62	INTAIRDMAREQ62	—	BFP7	1	—	group0-62	group0-62
INTAIRDMAREQ63	INTAIRDMAREQ63	—	BFP8	1	—	group0-63	group0-63

34.1.5 Reset Source

The ADCHn reset sources are listed in the following table. The ADCHn is initialized by the following reset sources.

Table 34.29 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
ADCHn	All registers	✓	✓	✓	✓	—	—
IFCu	All registers	✓	✓	✓	✓	—	—
ASFr	All registers	✓	✓	✓	✓	—	—
AVSEG	All registers	✓	✓	✓	✓	—	—
ABFG	All registers	✓	✓	✓	✓	—	—
AIR	All registers	✓	✓	✓	✓	—	—

34.1.6 External Input/Output Signals

The External Input/Output signals of the ADCHn are listed below.

Table 34.30 ADCH0 External Input / Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
ADCH0		
A0VCC	Power supply pin for the analog part	A0VCC
A0VSS	Ground pin for the analog part	A0VSS
A0VREFH	Reference voltage pin for the analog part	A0VREFH
AN000	Analog input pin	AN000
AN001	Analog input pin	AN001
AN002	Analog input pin	AN002
AN003	Analog input pin	AN003
AN010	Analog input pin	AN010
AN011	Analog input pin	AN011
AN012	Analog input pin	AN012
AN013	Analog input pin	AN013
AN020	Analog input pin	AN020
AN021	Analog input pin	AN021
AN022	Analog input pin	AN022
AN023	Analog input pin	AN023
AN030	Analog input pin	AN030
AN031	Analog input pin/ External analog multiplexer input pin	AN031
AN032	Analog input pin	AN032
AN033	Analog input pin	AN033
AN040	Analog input pin	AN040
AN041	Analog input pin	AN041
AN042	Analog input pin	AN042
AN043	Analog input pin	AN043
AN050	Analog input pin	AN050
AN051	Analog input pin	AN051
AN052	Analog input pin	AN052
AN053	Analog input pin	AN053
AN060	Analog input pin	AN060
AN061	Analog input pin	AN061
AN062	Analog input pin	AN062
AN063	Analog input pin	AN063
ADTRG0	External trigger pin*1	ADTRG0
ADEND0	A/D conversion timing monitor pin	ADEND0
ADMPX00	ADMPX value	ADMPX00
ADMPX01	ADMPX value	ADMPX01
ADMPX02	ADMPX value	ADMPX02
ADMPX03	ADMPX value	ADMPX03
ADMPX04	ADMPX value	ADMPX04

Note 1. When the external trigger pin is used, the noise filter for the port needs to be set. For details, refer to **Section 2, Pin Function**.

Table 34.31 ADCH2 External Input / Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
ADCH2		
A2VCC	Power supply pin for the analog part	A2VCC
A2VSS	Ground pin for the analog part	A2VSS
A2VREFH	Reference voltage pin for the analog part	A2VREFH
AN200	Analog input pin	AN200
AN201	Analog input pin	AN201
AN202	Analog input pin	AN202
AN203	Analog input pin	AN203
AN210	Analog input pin	AN210
AN211	Analog input pin	AN211
AN212	Analog input pin	AN212
AN213	Analog input pin	AN213
AN220	Analog input pin	AN220
AN221	Analog input pin	AN221
AN222	Analog input pin	AN222
AN223	Analog input pin	AN223
AN230	Analog input pin	AN230
AN231	Analog input pin	AN231
AN232	Analog input pin	AN232
AN233	Analog input pin	AN233
AN240	Analog input pin	AN240
AN241	Analog input pin	AN241
AN242	Analog input pin	AN242
AN243	Analog input pin / External analog multiplexer input pin	AN243
AN250	Analog input pin	AN250
AN251	Analog input pin	AN251
AN252	Analog input pin	AN252
AN253	Analog input pin	AN253
AN260	Analog input pin	AN260*2
AN261	Analog input pin	AN261*2
AN262	Analog input pin	AN262*2
AN263	Analog input pin	AN263*2
ADTRG2	External trigger pin*1	ADTRG2
ADEND2	A/D conversion timing monitor pin	ADEND2
ADMPX20	ADMPX value	ADMPX20
ADMPX21	ADMPX value	ADMPX21
ADMPX22	ADMPX value	ADMPX22
ADMPX23	ADMPX value	ADMPX23
ADMPX24	ADMPX value	ADMPX24

Note 1. When the external trigger pin is used, the noise filter for the port needs to be set. For details, refer to **Section 2, Pin Function**.

Note 2. These signals are not allocated to terminals in the 292-pin package.

Table 34.32 ADCH1 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
ADCH1		
A1VCC	Power supply pin for the analog part	A1VCC
A1VSS	Ground pin for the analog part	A1VSS
A1VREFH	Reference voltage pin for the analog part	A1VREFH
AN100	Analog input pin / External analog multiplexer input pin	AN100
AN101	Analog input pin	AN101
AN102	Analog input pin	AN102
AN103	Analog input pin	AN103
AN110	Analog input pin	AN110
AN111	Analog input pin	AN111
AN112	Analog input pin	AN112
AN113	Analog input pin	AN113
AN120	Analog input pin	AN120
AN121	Analog input pin	AN121
AN122	Analog input pin	AN122
AN123	Analog input pin	AN123
AN130	Analog input pin	AN130
AN131	Analog input pin	AN131
AN132	Analog input pin	AN132
AN133	Analog input pin	AN133
AN140	Analog input pin	AN140
AN141	Analog input pin	AN141
AN142	Analog input pin	AN142
AN143	Analog input pin	AN143
ADTRG1	External trigger pin* ¹	ADTRG1
ADEND1	A/D conversion timing monitor pin	ADEND1
ADMPX10	ADMPX value	ADMPX10
ADMPX11	ADMPX value	ADMPX11
ADMPX12	ADMPX value	ADMPX12
ADMPX13	ADMPX value	ADMPX13
ADMPX14	ADMPX value	ADMPX14

Note 1. When the external trigger pin is used, the noise filter for the port needs to be set. For details, refer to **Section 2, Pin Function**.

Table 34.33 ADCH3 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
ADCH3		
A3VCC	Power supply pin for the analog part	A3VCC
A3VSS	Ground pin for the analog part	A3VSS
A3VREFH	Reference voltage pin for the analog part	A3VREFH
AN300	Analog input pin / External analog multiplexer input pin	AN300
AN301	Analog input pin	AN301
AN302	Analog input pin	AN302
AN303	Analog input pin	AN303
AN310	Analog input pin	AN310
AN311	Analog input pin	AN311
AN312	Analog input pin	AN312
AN313	Analog input pin	AN313
AN320	Analog input pin	AN320
AN321	Analog input pin	AN321
AN322	Analog input pin	AN322
AN323	Analog input pin	AN323
AN330	Analog input pin	AN330
AN331	Analog input pin	AN331
AN332	Analog input pin	AN332
AN333	Analog input pin	AN333
AN340	Analog input pin	AN340
AN341	Analog input pin	AN341
AN342	Analog input pin	AN342
AN343	Analog input pin	AN343
ADTRG3	External trigger pin*1	ADTRG3
ADEND3	A/D conversion timing monitor pin	ADEND3
ADMPX30	ADMPX value	ADMPX30
ADMPX31	ADMPX value	ADMPX31
ADMPX32	ADMPX value	ADMPX32
ADMPX33	ADMPX value	ADMPX33
ADMPX34	ADMPX value	ADMPX34

Note 1. When the external trigger pin is used, the noise filter for the port needs to be set. For details, refer to **Section 2, Pin Function**.

For ADCHn, it's possible to perform an A/D conversion of an analog input with the assistance of RRAMP.

For ADCH0, it's possible to perform an A/D conversion of VCC, E0VCC (called EVCC hereafter) and VDD as secondary power supply voltage monitor.

For analog inputs and VMON (Secondary power supply voltage monitor), the correspondence list of physical channel, physical channel groups, physical sub channel and RRAMP is shown in the following table.

Table 34.34 ADCH0 Physical Channel, Physical Channel Group, Physical Sub Channel and RRAMP

Analog Input or VMON	Physical Channel	Physical Channel Group	Physical Sub Channel	RRAMP Correspondence			
				RRAMP00	RRAMP01	RRAMP02	—
AN000	0	0	0	✓	—	—	—
AN001	1		1	—	✓	—	—
AN002	2		2	—	—	✓	—
AN003	3		3	✓	—	—	—
AN010	4	1	0	—	✓	—	—
AN011	5		1	—	—	✓	—
AN012	6		2	✓	—	—	—
AN013	7		3	—	✓	—	—
AN020	8	2	0	—	—	✓	—
AN021	9		1	✓	—	—	—
AN022	10		2	—	✓	—	—
AN023	11		3	—	—	✓	—
AN030	12	3	0	✓	—	—	—
AN031	13		1	—	✓	—	—
AN032	14		2	—	—	✓	—
AN033	15		3	✓	—	—	—
AN040	16	4	0	—	✓	—	—
AN041	17		1	—	—	✓	—
AN042	18		2	✓	—	—	—
AN043	19		3	—	✓	—	—
AN050	20	5	0	—	—	✓	—
AN051	21		1	✓	—	—	—
AN052	22		2	—	✓	—	—
AN053	23		3	—	—	✓	—
AN060	24	6	0	✓	—	—	—
AN061	25		1	—	✓	—	—
AN062	26		2	—	—	✓	—
AN063	27		3	✓	—	—	—
VMON_VCC*1	28	7	0	—	—	—	—
VMON_EVCC*2	29		1	—	—	—	—
VMON_VDD*3	30		2	—	—	—	—

Note 1. This is the secondary supply voltage monitor for VCC.

Note 2. This is the secondary supply voltage monitor for EVCC.

Note 3. This is the secondary supply voltage monitor for VDD.

Table 34.35 ADCH2 Physical Channel, Physical Channel Group, Physical Sub Channel and RRAMP

Analog Input	Physical Channel	Physical Channel Group	Physical Sub Channel	RRAMP Correspondence			
				RRAMP20	RRAMP21	RRAMP22	—
AN200	0	0	0	✓	—	—	—
AN201	1		1	—	✓	—	—
AN202	2		2	—	—	✓	—
AN203	3		3	✓	—	—	—
AN210	4	1	0	—	✓	—	—
AN211	5		1	—	—	✓	—
AN212	6		2	✓	—	—	—
AN213	7		3	—	✓	—	—
AN220	8	2	0	—	—	✓	—
AN221	9		1	✓	—	—	—
AN222	10		2	—	✓	—	—
AN223	11		3	—	—	✓	—
AN230	12	3	0	✓	—	—	—
AN231	13		1	—	✓	—	—
AN232	14		2	—	—	✓	—
AN233	15		3	✓	—	—	—
AN240	16	4	0	—	✓	—	—
AN241	17		1	—	—	✓	—
AN242	18		2	✓	—	—	—
AN243	19		3	—	✓	—	—
AN250	20	5	0	—	—	✓	—
AN251	21		1	✓	—	—	—
AN252	22		2	—	✓	—	—
AN253	23		3	—	—	✓	—
AN260 *1	24	6	0	✓	—	—	—
AN261 *1	25		1	—	✓	—	—
AN262 *1	26		2	—	—	✓	—
AN263 *1	27		3	✓	—	—	—

Note 1. These signals are not allocated to terminals in the 292-pin package.

Table 34.36 ADCH1 Physical Channel, Physical Channel Group, Physical Sub Channel and RRAMP

Analog Input	Physical Channel	Physical Channel Group	Physical Sub Channel	RRAMP Correspondence			
				RRAMP10	RRAMP11	—	—
AN100	0	0	0	✓	—	—	—
AN101	1		1	—	✓	—	—
AN102	2		2	✓	—	—	—
AN103	3		3	—	✓	—	—
AN110	4	1	0	✓	—	—	—
AN111	5		1	—	✓	—	—
AN112	6		2	✓	—	—	—
AN113	7		3	—	✓	—	—
AN120	8	2	0	✓	—	—	—
AN121	9		1	—	✓	—	—
AN122	10		2	✓	—	—	—
AN123	11		3	—	✓	—	—
AN130	12	3	0	✓	—	—	—
AN131	13		1	—	✓	—	—
AN132	14		2	✓	—	—	—
AN133	15		3	—	✓	—	—
AN140	16	4	0	✓	—	—	—
AN141	17		1	—	✓	—	—
AN142	18		2	✓	—	—	—
AN143	19		3	—	✓	—	—

Table 34.37 ADCH3 Physical Channel, Physical Channel Group, Physical Sub Channel and RRAMP

Analog Input	Physical Channel	Physical Channel Group	Physical Sub Channel	RRAMP Correspondence			
				RRAMP30	RRAMP31	—	—
AN300	0	0	0	✓	—	—	—
AN301	1		1	—	✓	—	—
AN302	2		2	✓	—	—	—
AN303	3		3	—	✓	—	—
AN310	4	1	0	✓	—	—	—
AN311	5		1	—	✓	—	—
AN312	6		2	✓	—	—	—
AN313	7		3	—	✓	—	—
AN320	8	2	0	✓	—	—	—
AN321	9		1	—	✓	—	—
AN322	10		2	✓	—	—	—
AN323	11		3	—	✓	—	—
AN330	12	3	0	✓	—	—	—
AN331	13		1	—	✓	—	—
AN332	14		2	✓	—	—	—
AN333	15		3	—	✓	—	—
AN340	16	4	0	✓	—	—	—
AN341	17		1	—	✓	—	—
AN342	18		2	✓	—	—	—
AN343	19		3	—	✓	—	—

34.2 Overview

34.2.1 Functional Overview

The ADCH has the following features.

- Configuration of analog input pins
See **Section 34.1.1, Number of Units and Channels** about the number of modules, the number of total analog input pins, the number of dedicated analog input pins and the number of shared analog input pins.
- Sample-and-hold function
Each module has an internal sample-and-hold circuit that enables each module to perform an A/D conversion independently.
- Advanced A/D converter
Resolution: 12 / 10 bits
A/D conversion method: Successive approximation
Conversion speed: 1.0 μ s
- Supporting five scan groups
Each ADCH has five scan groups. Scan settings can be made independently for each scan group.
- Two scan modes
Each ADCH has two scan modes.
Multicycle scan mode executes the specified number of scans.
Continuous scan mode executes scans repeatedly without limit.
- Virtual channels
Each ADCH has several virtual channels. See **Section 34.1.1, Number of Units and Channels** about the number of virtual channels. Analog channels for which A/D conversion is to be made and other accompanying information are set for each virtual channel. By sequentially performing the processing for the virtual channels indicated by the start virtual channel pointer and the end virtual channel pointer in each scan group, scans (which can perform A/D conversion for any analog channels in any order) can be executed. Each virtual channel can be mapped to any physical channel.
- Extended physical channels
Each ADCH can extend physical channels by using an external analog multiplexer. Furthermore, an external analog multiplexer can be used with either I/O port (pins) or the CSIH (peripheral). The I/O port and/or CSIH is under the control of the ADCH for multiplexer channel selection. The CSIH interface requires DMA when used. (One channel can be set for each ADCH.)
- Interval function
The ADCH can start scan groups in any cycle by using the A/D timer equipped in the scan groups 3 and 4. This enables scans with intervals inserted.
- A/D-converted value adding function
The ADCH performs A/D conversion sequentially twice or four times for a channel, and stores the addition result in the data register. The setting of use or not use of the addition function can be set each virtual channel. But the setting of addition count can be set as common setting in all virtual channels.
The effect of the moving average filter can be gained by using this function.
In addition, using the optional ADC Summation Function (ASF) makes an extended summation function available.
For details, see **Section 34.8, ADC Summation Function (ASF)**.

- Data registers

Data registers corresponding to virtual channels are provided.

- Start trigger for each scan group

Hardware triggers and software triggers can start the processing of each scan group. Only scan groups 3 and 4 can start processing by an A/D timer trigger.

- Asynchronous/synchronous suspend and resume function

Processing for a scan group can interrupt an ongoing processing for another scan group.

The priority is as follows:

Low	High	
$SG0 < SG1 < SG2 < SG3 < SG4$		(SG: Scan group)

The following suspend methods can be chosen:

Synchronous suspend: Processing is suspended after the ongoing virtual channel processing has stopped.

Asynchronous suspend: Processing is suspended immediately.

Synchronous/asynchronous mixture type suspend:

SGx except for SG0 is “synchronous suspend” mode.

SG0 is in “asynchronous suspend” mode.

- Entry to the digital filter engine, and the ADC summation function, and the Generic Timer Module

A/D-converted values can be directly entered into the Digital Filter Engine (DFE), or the ADC Summation Function (ASF), or the Generic Timer Module (GTM). Whether to enable or disable entry and a tag used to define the target channel to be entered can be set for each virtual channel. Entry to the DFE (GTM) or entry to the ASF can be set for each scan group.

For details of the DFE, refer to **Section 37, Digital Filter Engine (DFE)**.

For details of the ASF, refer to **Section 34.8, ADC Summation Function (ASF)**.

For details of the GTM, refer to **Section 31, Generic Timer Module (GTM)**.

- Scan end interrupt and DMA transfer

Each scan group can generate an interrupt request to the INTC and activate the DMAC each time a processing for the virtual channel indicated by the end virtual channel pointer ends or a virtual channel ends.

- Interrupt and DMA transfer for an external analog multiplexer

The ADCH can generate an interrupt request to the INTC and activate the DMAC when the specified virtual channel is started. This enables transfer of the MPX value to an external analog multiplexer in cooperation with an I/O port or CSIH. There is a register to provide for a programmable ADC conversion delay (N us (N = 0, 1, 2, ..., 10)) for DMA transfer time and external multiplexer settling time.

- Selectable analog conversion voltage

The AnVREFH pin can be used to set the voltage range for analog conversion.

- A/D conversion monitor output

The processing timing of a desired virtual channel can be output to the A/D conversion monitor output pin (ADENDn).

- A wide range of safety functions

The ADCH has various safety functions, including self-diagnosis, pin-level self-diagnosis, wiring-break detection, self-diagnosis of the wiring-break detection function, self-diagnosis of the secondary power supply voltage monitor function, normality check for analog selection; and for the data registers, upper-limit/lower-limit check, parity check, overwrite check, and read and clear functions.

- Upper/lower-limit-excursion-notice-function to ADC VMON secondary error generator (AVSEG) or ADC boundary flag generator (ABFG) in each virtual channel

The ADCHn can output to the ADC-VMON-secondary-error-generator (AVSEG) (The output to AVSEG is ADCH0 only) or the ADC-boundary-flag-generator (ABFG) a signal to notify that an A/D conversion result has increased above the upper limit value of the designated table in each virtual channel or that an A/D conversion result has decreased below the lower limit value of the designated table in each virtual channel.

Once the Table register for the upper/lower limit value is set, it can be rewritten at any time.

For details of the AVSEG, **Section 34.9, ADC VMON Secondary Error Generator (AVSEG)**.

For details of the ABFG, **Section 34.10, ADC Boundary Flag Generator (ABFG)**.

- Secondary power supply voltage monitor (VMON)

The ADCH0 can convert the voltage of VCC, EVCC, and the VDD power supply in AD.

Secondary HDET, Secondary LDET of each power supply (VCC, EVCC and VDD) can be notified to ECM by using the upper/lower-limit-excess-notice-function of ADCH0 and AVSEG.

For details of AVSEG, **Section 34.9, ADC VMON Secondary Error Generator (AVSEG)**.

For the whole description of power supply voltage monitor, refer to **Section 12, Power Supply Voltage Monitor**.

34.2.2 Block Diagram

The block diagram of ADCH0 is shown in **Figure 34.1**. The block diagram of ADCH2 is shown in **Figure 34.2**. The block diagram of ADCH1 is shown in **Figure 34.3**. The block diagram of ADCH3 is shown in **Figure 34.4**.

(1) Configuration of ADCH0

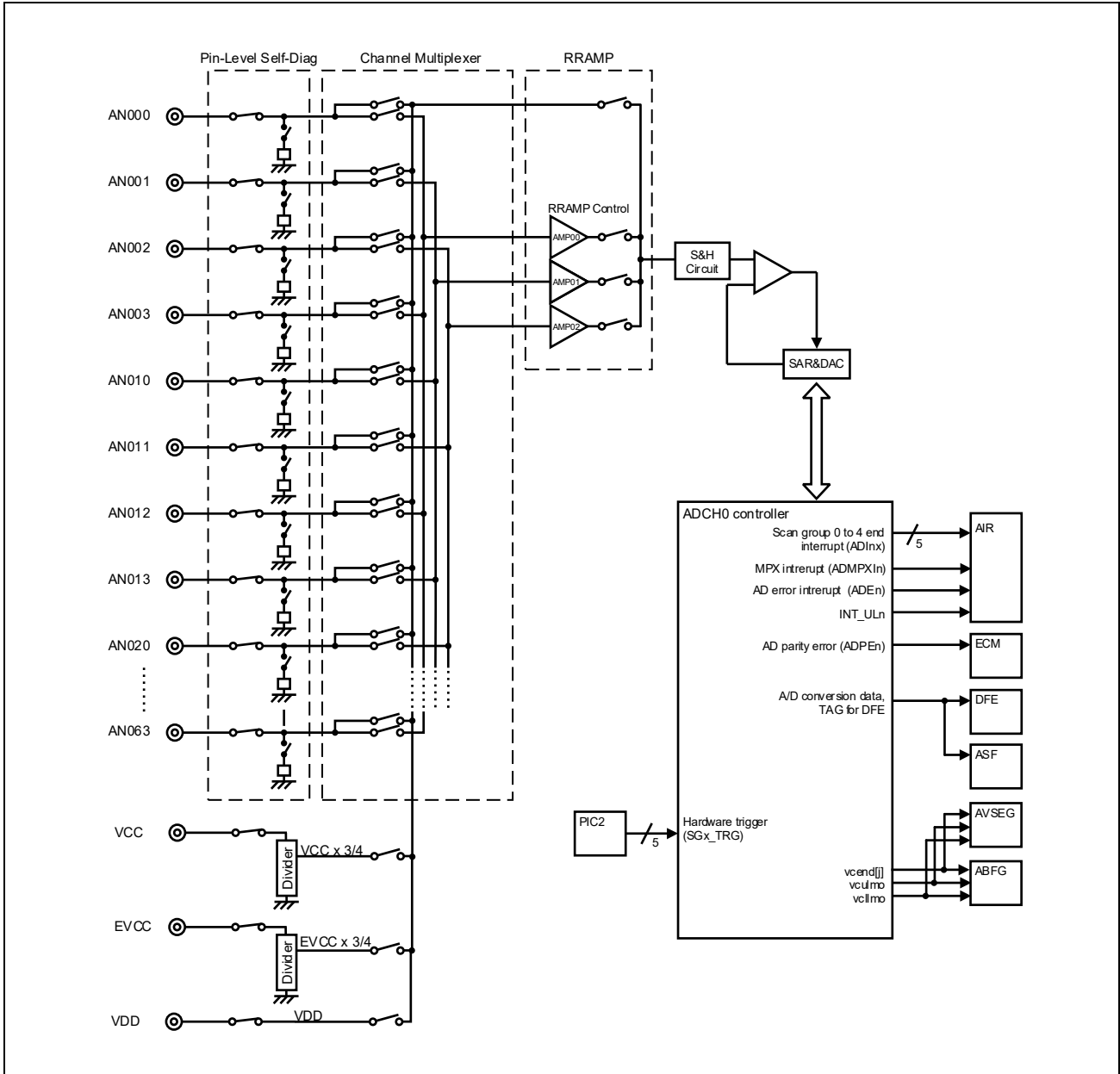


Figure 34.1 ADCH0 Block Diagram

(2) Configuration of ADCH2

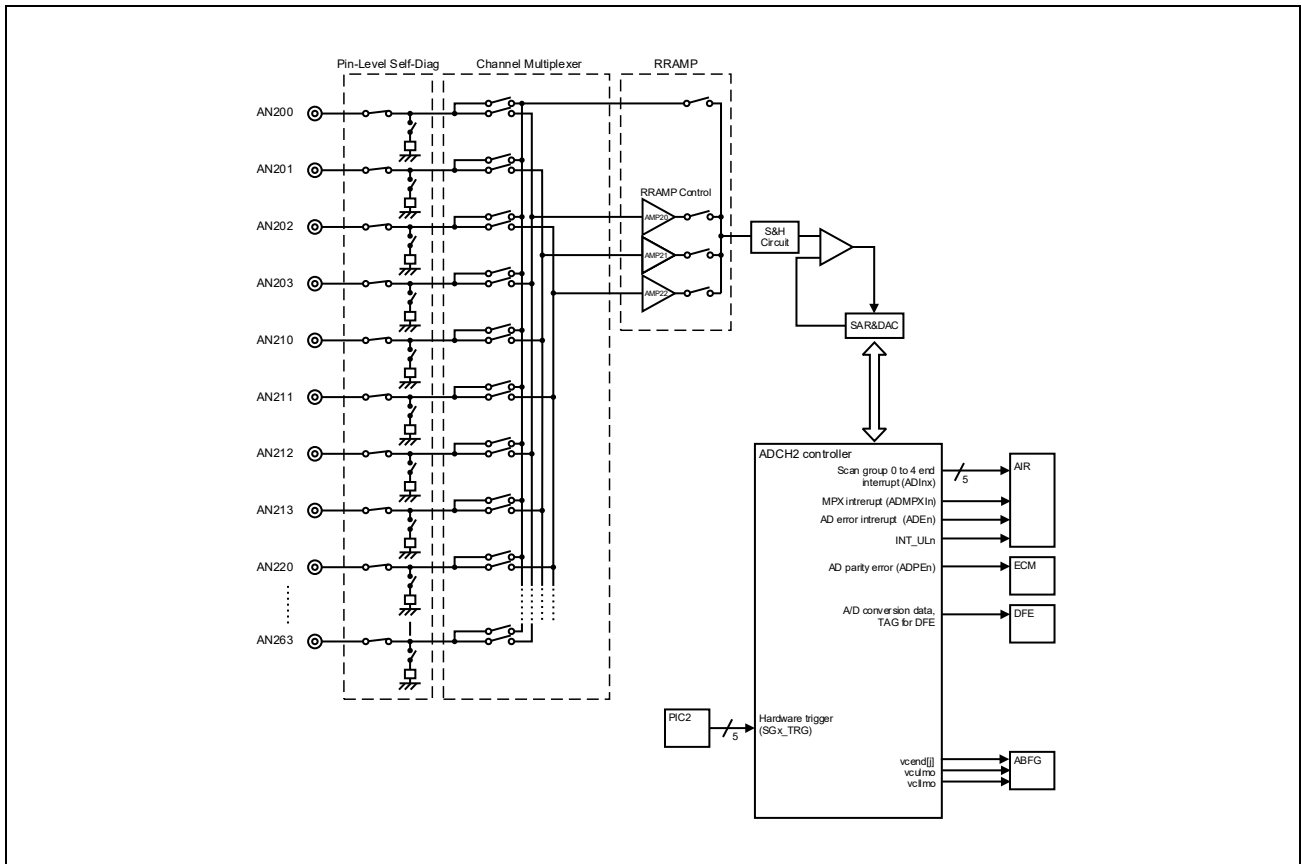


Figure 34.2 ADCH2 Block Diagram

(3) Configuration of ADCH1

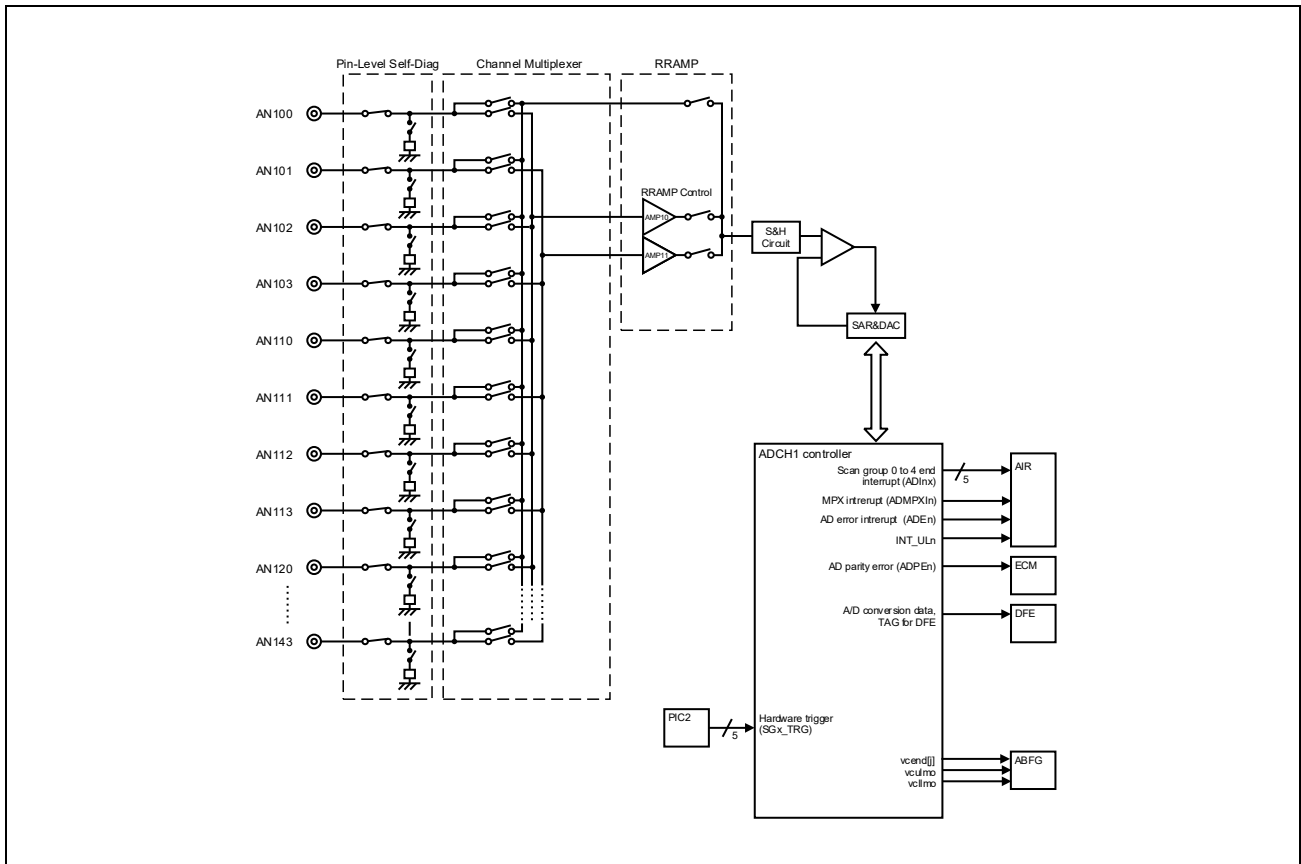


Figure 34.3 ADCH1 Block Diagram

(4) Configuration of ADCH3

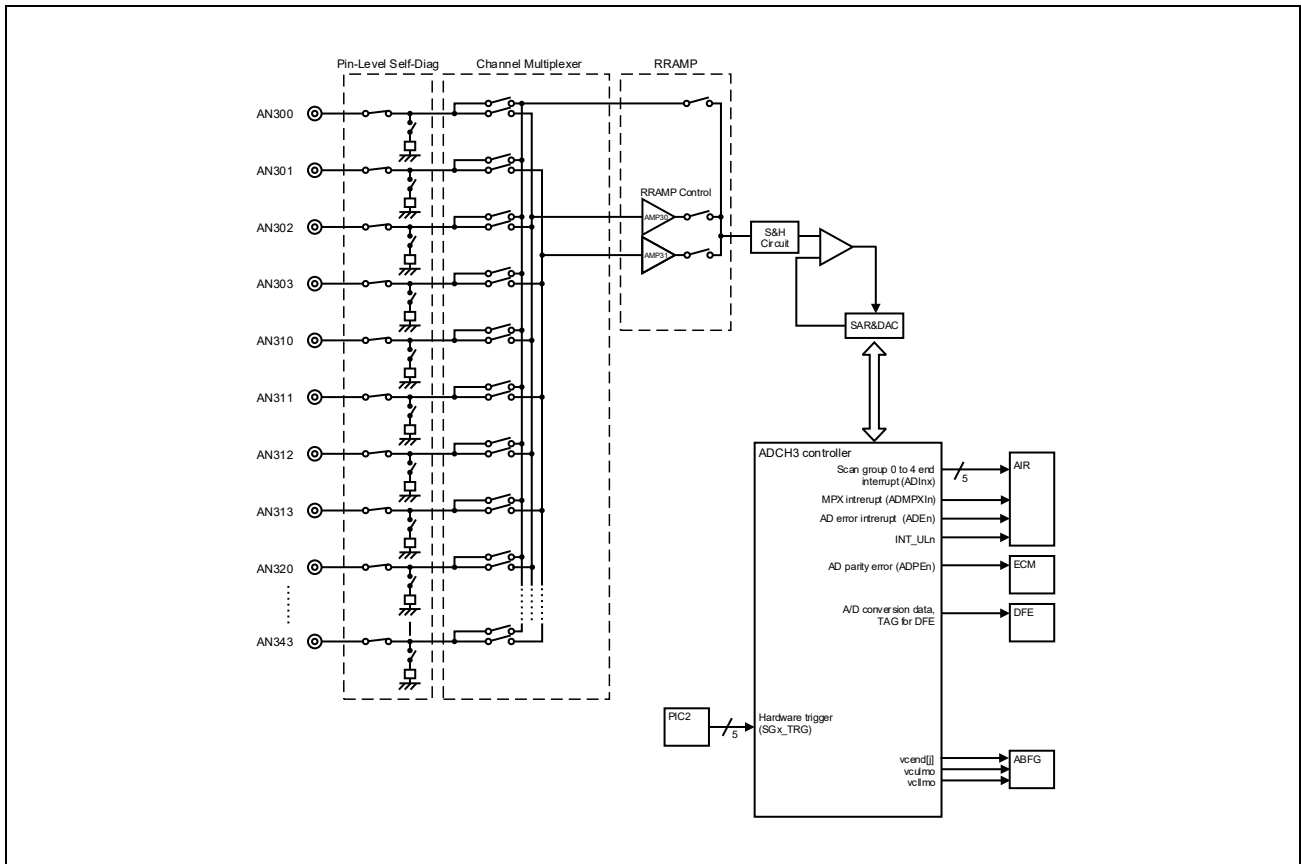


Figure 34.4 ADCH3 Block Diagram

34.3 Registers

34.3.1 List of Registers

The ADCH registers are listed in the following table.

For details about <ADCHn_base>, see **Section 34.1.2, Register Base Address**.

Table 34.38 List of Registers (1/3)

Module Name	Register Name	Symbol	Address	Access	Access Protection
ADCH Common Registers					
ADCH0	A/D Synchronization Start Control Register	ADCH0ADSYNSTCR	<ADCH0_base> + 300 _H	8	—
ADCH0	A/D Timer Synchronization Start Control Register	ADCH0ADTSYNSTCR	<ADCH0_base> + 304 _H	8	—
ADCH0	Voltage Monitor Voltage Divider Control Register 1	ADCH0VMONVDCR1	<ADCH0_base> + 30C _H	8	—
ADCH0	Voltage Monitor Voltage Divider Control Register 2	ADCH0VMONVDCR2	<ADCH0_base> + 310 _H	8	—
ADCH Specific Registers (Virtual Channel)					
ADCHn	Virtual Channel Register j	ADCHnVCRj	<ADCHn_base> + j × 4 _H	8, 16, 32	—
ADCHn	Data Register j	ADCHnDRj	<ADCHn_base> + 100 _H + j × 2 _H	16, 32 (j is even) 16 (j is odd)	—
ADCHn	Data Supplementary Information Register j	ADCHnDIRj	<ADCHn_base> + 200 _H + j × 4 _H	16, 32	—
ADCH Specific Registers (Control)					
ADCHn	A/D Halt Register	ADCHnADHALTR	<ADCHn_base> + 380 _H	8	—
ADCHn	A/D Control Register 1	ADCHnADCR1	<ADCHn_base> + 384 _H	8	—
ADCHn	MPX Current Control Register	ADCHnMPXCURCR	<ADCHn_base> + 388 _H	8	—
ADCHn	MPX Interrupt Enable Register	ADCHnMPXINTER	<ADCHn_base> + 38A _H	8	—
ADCHn	MPX Current Register	ADCHnMPXCURR	<ADCHn_base> + 38C _H	16, 32	—
ADCHn	MPX Optional Wait Register	ADCHnMPXOWR	<ADCHn_base> + 390 _H	8	—
ADCHn	MPX Command Information Register	ADCHnMPXCMDR	<ADCHn_base> + 394 _H	8	—
ADCHn	A/D Control Register 2	ADCHnADCR2	<ADCHn_base> + 398 _H	8	—
ADCHn	DFE / ASF Entry Scan Group Enable Register	ADCHnDFASENTSGER	<ADCHn_base> + 39C _H	8, 16	—
ADCHn	A/D Conversion Monitor Virtual Channel Pointer	ADCHnADENDP	<ADCHn_base> + 3A0 _H	8	—

Table 34.38 List of Registers (2/3)

Module Name	Register Name	Symbol	Address	Access	Access Protection
ADCH Specific Registers (Safety-related)					
ADCHn	Safety Control Register	ADCHnSFTCR	<ADCHn_base> + 3C0 _H	8	—
ADCHn	Pin-Level Self-Diagnosis Control Register	ADCHnTDCCR	<ADCHn_base> + 3C4 _H	8	—
ADCHn	Wiring-Break Detection Control Register	ADCHnODCR	<ADCHn_base> + 3C8 _H	8, 16, 32	—
ADCHn	Error Clear Register	ADCHnECR	<ADCHn_base> + 3D8 _H	8	—
ADCHn	Overwrite Error Register	ADCHnOWER	<ADCHn_base> + 3E0 _H	8	—
ADCHn	Parity Error Register	ADCHnPER	<ADCHn_base> + 3E4 _H	8	—
ADCHn	Virtual Channel Upper / Lower Limit Table Register 0	ADCHnVCULLMTBR0	<ADCHn_base> + 700 _H	32	—
ADCHn	Virtual Channel Upper / Lower Limit Table Register 1	ADCHnVCULLMTBR1	<ADCHn_base> + 704 _H	32	—
ADCHn	Virtual Channel Upper / Lower Limit Table Register 2	ADCHnVCULLMTBR2	<ADCHn_base> + 708 _H	32	—
ADCHn	Virtual Channel Upper / Lower Limit Table Register 3	ADCHnVCULLMTBR3	<ADCHn_base> + 70C _H	32	—
ADCHn	Virtual Channel Upper / Lower Limit Table Register 4	ADCHnVCULLMTBR4	<ADCHn_base> + 710 _H	32	—
ADCHn	Virtual Channel Upper / Lower Limit Table Register 5	ADCHnVCULLMTBR5	<ADCHn_base> + 714 _H	32	—
ADCHn	Virtual Channel Upper / Lower Limit Table Register 6	ADCHnVCULLMTBR6	<ADCHn_base> + 718 _H	32	—
ADCHn	Virtual Channel Upper / Lower Limit Error Interrupt Enable Register 1	ADCHnVCLMINTER1	<ADCHn_base> + 730 _H	8, 16, 32	—
ADCHn	Virtual Channel Upper / Lower Limit Error Interrupt Enable Register 2	ADCHnVCLMINTER2	<ADCHn_base> + 734 _H	8, 16, 32	—
ADCHn	Virtual Channel Upper / Lower Limit Excess Status Register 1	ADCHnVCLMSR1	<ADCHn_base> + 738 _H	32	—
ADCHn	Virtual Channel Upper / Lower Limit Excess Status Register 2	ADCHnVCLMSR2	<ADCHn_base> + 73C _H	32	—
ADCHn	Virtual Channel Upper / Lower Limit Excess Status Clear Register 1	ADCHnVCLMSCR1	<ADCHn_base> + 740 _H	8, 16, 32	—
ADCHn	Virtual Channel Upper / Lower Limit Excess Status Clear Register 2	ADCHnVCLMSCR2	<ADCHn_base> + 744 _H	8, 16, 32	—

Table 34.38 List of Registers (3/3)

Module Name	Register Name	Symbol	Address	Access	Access Protection
Scan Group Specific Registers					
ADCHn	Scan Group x Start Control Register	ADCHnSGSTCRx	<ADCHn_base> + x × 80 _H + 480 _H	8	
ADCHn	Scan Group x Stop Control Register	ADCHnSGSTPCRx	<ADCHn_base> + x × 80 _H + 484 _H	8	
ADCHn	A/D Timer y Start Control Register	ADCHnADTSTCRy	<ADCHn_base> + y × 80 _H + 488 _H	8	
ADCHn	A/D Timer y End Control Register	ADCHnADTENDCRy	<ADCHn_base> + y × 80 _H + 48C _H	8	
ADCHn	Scan Group x Control Register	ADCHnSGCRx	<ADCHn_base> + x × 80 _H + 490 _H	8	
ADCHn	Scan Group x Start Virtual Channel Pointer	ADCHnSGVCSPx	<ADCHn_base> + x × 80 _H + 494 _H	8	
ADCHn	Scan Group x End Virtual Channel Pointer	ADCHnSGVCEPx	<ADCHn_base> + x × 80 _H + 498 _H	8	
ADCHn	Scan Group x Multicycle Register	ADCHnSGMICYRx	<ADCHn_base> + x × 80 _H + 49C _H	8	
ADCHn	Scan Group x Virtual Channel Pointer Register	ADCHnSGVCPRx	<ADCHn_base> + x × 80 _H + 4A0 _H	16	
ADCHn	Scan Group x Status Register	ADCHnSGSRx	<ADCHn_base> + x × 80 _H + 4A4 _H	8	
ADCHn	A/D Timer y Initial Phase Register	ADCHnADTIPRy	<ADCHn_base> + y × 80 _H + 4A8 _H	32	
ADCHn	A/D Timer y Cycle Register	ADCHnADTPRRy	<ADCHn_base> + y × 80 _H + 4AC _H	32	
ADCHn	Scan Group x Virtual Channel Optional Waiting Times Register	ADCHnSGVCOWRx	<ADCHn_base> + x × 80 _H + 4B4 _H	16	

34.3.2 ADCH Common Registers

This section describes the common registers for each ADCH.

34.3.2.1 ADCH0ADSYNSTCR — A/D Synchronization Start Control Register

ADCH0ADSYNSTCR is an 8-bit write-only register that controls the simultaneous start of A/D conversions for scan groups of each ADCH. The register bits are always read as 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADSTART
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 34.39 ADCH0ADSYNSTCR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ADSTART	This bit starts an A/D conversion for scan groups of each ADCH. The Start condition for SGx of ADCHn: 1 is written to ADSTART when SGACTION for SGx of ADCHn is 0 and ADSTARTEN is 1. The A/D conversion is started simultaneously for the scan groups (of each ADCH) for which ADSTARTEN has been set to 1. Writing 0 to this bit is ignored.

34.3.2.2 ADCH0ADTSYNSTCR — A/D Timer Synchronization Start Control Register

ADCH0ADTSYNSTCR is an 8-bit write-only register that controls the simultaneous start of counting by the A/D timers of each ADCH. The register bits are always read as 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADTSTART
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 34.40 ADCH0ADTSYNSTCR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ADTSTART	This bit starts the counting by the A/D timers of each ADCH. The start condition for SGx of ADCHn: 1 is written to ADTSTART when ADTACT for A/D timer y of ADCHn is 0 and ADTSTARTE is 1. The A/D timers (of each ADCH) for which ADTST has been set to 1 start counting simultaneously. Writing 0 to this bit is ignored.

34.3.2.3 ADCH0VMONVDCR1 — Voltage Monitor Voltage Divider Control Register 1

ADCH0VMONVDCR1 is an 8-bit readable / writable register that controls the voltage monitor voltage divider of the power supply. This is only present in ADCH0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	VDE1
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 34.41 ADCH0VMONVDCR1 Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	VDE1	Voltage divider enable (resistance voltage divider control) The resistance voltage divider is controlled by ADCH0VMONVDCR2.VDE2 and VDE1. For details, refer to Table 34.43 .

CAUTION

To prevent a malfunction, there are the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.

Do not perform an A/D conversion with ADCH0VMONVDCR1.VDE1, ADCH0VMONVDCR2.VDE2 = 01_B or 10_B.

34.3.2.4 ADCH0VMONVDCR2 — Voltage Monitor Voltage Divider Control Register 2

ADCH0VMONVDCR2 is an 8-bit readable / writable register that controls the voltage monitor voltage divider of the power supply. This is only present in ADCH0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	VDE2
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 34.42 ADCH0VMONVDCR2 Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	VDE2	Voltage divider enable (resistance voltage divider control) The resistance voltage divider is controlled by ADCH0VMONVDCR1.VDE1 and VDE2. For details, refer to Table 34.43 .

CAUTION

To prevent a malfunction, there is the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.

Do not perform an A/D conversion with ADCH0VMONVDCR1.VDE1, ADCH0VMONVDCR2.VDE2 = 01_B or 10_B setting.

Table 34.43 Resistance Voltage Divider and Pull-down Control Setting Table

VDE1	VDE2	Resistance voltage divide	Pull-down
0	0	Off	On
0	1	Off	Off
1	0	Off	Off
1	1	On	Off

34.3.3 ADCH Specific Registers (Virtual Channel, Control, Safety-related)

34.3.3.1 ADCHnVCRj — Virtual Channel Register j

ADCHnVCRj is a 32-bit readable/writable register used for each virtual channel. ADCHnVCRj is initialized to 0000 0000_H at reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VCULME	VCLLME	—	—	—	VCULLMTBS[2:0]			—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNVCLS[2:0]			DFENT	DFTAG[3:0]				ADIE	—	GCTRL[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.44 ADCHnVCRj Register Contents (1/3)

Bit Position	Bit Name	Function
31	VCULME	Virtual channel upper limit excess notice enable 0: An A/D conversion result greater than the upper limit of the virtual channel is not notified. 1: An A/D conversion result greater than the upper limit of the virtual channel is notified. Notification is made by vchend[j] (A/D conversion end notice of ADCHnVCRj) and vculmo (virtual channel upper limit excess notice). For details, first see Figure 34.27 and see Section 34.4.9, Virtual Channel Upper/Lower Limit Excess Notice .
30	VCLLME	Virtual channel lower limit excess notice enable 0: An A/D conversion result is less than the lower limit of the virtual channel is not notified. 1: An A/D conversion result is less than the lower limit of the virtual channel is notified. Notification is made by vchend[j] (A/D conversion end notice of ADCHnVCRj) and vcllmo (virtual channel lower limit excess notice). For details, first see Figure 34.27 and see Section 34.4.9, Virtual Channel Upper/Lower Limit Excess Notice .
29 to 27	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 34.44 ADCHnVCRj Register Contents (2/3)

Bit Position	Bit Name	Function
26 to 24	VCULLMTBS[2:0]	Virtual channel upper/lower limit table register selection Selects Virtual channel upper/lower limit table register that is the comparative target. 0 _H : ADCHnVCULLMTBR0 is chosen [default] 1 _H : ADCHnVCULLMTBR1 is chosen 2 _H : ADCHnVCULLMTBR2 is chosen 3 _H : ADCHnVCULLMTBR3 is chosen 4 _H : ADCHnVCULLMTBR4 is chosen 5 _H : ADCHnVCULLMTBR5 is chosen 6 _H : ADCHnVCULLMTBR6 is chosen 7 _H : Setting prohibited
23 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 13	CNVCLS[2:0]	Conversion Class 0 _H : Normal A/D conversion 3 _H : A/D converter self-diagnosis 4 _H : Normal A/D conversion in addition mode 5 _H : Normal A/D conversion with the MPX 6 _H : Normal A/D conversion with the MPX in addition mode Other than above: Setting prohibited
12	DFENT	DFE Entry 0: Entry disabled 1: Entry enabled This bit specifies whether to enable or disable entry to the DFE (Digital Filter Engine), or ASF (ADC summation function). Only the scan groups enabled by DFENTSGxE and ASENTSGxE in the ADCHnDFASENTSGER are entered.
11 to 8	DFTAG[3:0]	DFE-TAG When entry to the DFE is requested, entry is made to the DFE channel for which the same tag with DFTAG[3:0] is set. If this applies to multiple channels, entry is made to multiple channels. When entry to the ASF is requested, entry is made to the ASF channel corresponding to the values set in DFTAG[3:0]. When entry to the ASF is requested, set the DFTAG [3:0] of each ADCHnVCRj to a different value. When the DFTAG [3:0] of several ADCHnVCRj is set to the same value, accumulation data of ASF can't be guaranteed.
7	ADIE	Virtual Channel End Interrupt Enable 0: ADInx is not output when the A/D conversion for virtual channel n ends in SGx. 1: ADInx is output when the A/D conversion for virtual channel n ends in SGx. ADIE in ADCHnSGCRx is independent of ADIE in ADCHnVCRj. For details, see Section 34.4.6.1, Scan End Interrupt Request .
6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 34.44 ADCHnVCRj Register Contents (3/3)

Bit Position	Bit Name	Function
5 to 0	GCTRL[5:0]	<p>General Control</p> <ul style="list-style-type: none"> ●For normal A/D conversions (CNVCLS[2:0] = 0_H) <ul style="list-style-type: none"> –GCTRL[4:2]: Physical Channel Group These bits specify a physical channel group. –GCTRL[1:0]: Physical Sub channel These bits specify a physical sub channel. Always set the other GCTRL bits to 0. ●For self-diagnosis (CNVCLS[2:0] = 3_H) <ul style="list-style-type: none"> – GCTRL[4:0]: Self-diagnostic Voltage Level 10_H: AnVREFH × 1 0C_H: AnVREFH × 3/4 08_H: AnVREFH × 1/2 04_H: AnVREFH × 1/4 00_H: AnVREFH × 0 Other than the above: Setting prohibited Always set the other GCTRL bits to 0. ●For normal A/D conversions in addition mode (CNVCLS[2:0] = 4_H) <ul style="list-style-type: none"> –GCTRL[4:2]: Physical Channel Group These bits specify a physical channel group. –GCTRL[1:0]: Physical sub channel These bits specify a physical sub channel. Always set the other GCTRL bits to 0. The count specified by ADDNT is reflected as the number of addition times. ●For normal A/D conversions with the MPX (CNVCLS[2:0] = 5_H) <ul style="list-style-type: none"> –GCTRL[4:0]: MPX Channel Setting These bits set the MPX value to be transferred to an external analog multiplexer. The GCTRL[4:0] value is transferred to ADCHnMPXCURR at the start of a virtual channel, and an interrupt request (ADMPXIn) or a DMA request is issued. Activating an interrupt or the DMAC, and transferring ADCHnMPXCURR to PSRn of the I/O port, or to CSIHnTX0H of CSIH enables the transfer of the MPX value to an external analog multiplexer. For details, see Section 34.4.4.4, Normal A/D Conversion with the MPX and Normal A/D Conversion with the MPX in Addition Mode. Always set the other GCTRL bits to 0. ●For normal A/D conversions with the MPX in addition mode (CNVCLS[2:0] = 6_H) <ul style="list-style-type: none"> –GCTRL[4:0]: MPX Channel Setting These bits set the MPX value to be transferred to an external analog multiplexer. The GCTRL[4:0] value is transferred to ADCHnMPXCURR at the start of a virtual channel, and an interrupt request (ADMPXIn) or a DMA request is issued. Activating an interrupt or the DMAC, and transferring ADCHnMPXCURR to PSRn of the I/O port, or to CSIHnTX0H of CSIH enables the transfer of the MPX value to an external analog multiplexer. For details, see Section 34.4.4.4, Normal A/D Conversion with the MPX and Normal A/D Conversion with the MPX in Addition Mode. The count specified by ADDNT is reflected as the number of addition times. Always set the other GCTRL bits to 0.

CAUTION

To prevent a malfunction, there are the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**. But when scan group is terminated, clearing ADIE to 0 isn't included in the limitations.

34.3.3.2 ADCHnDRj — Data Register j

ADCHnDRj is a 16-bit read-only register that stores an A/D converted value.

ADCHnDRj is provided for each virtual channel, and its format depends on the DFMT[1:0] setting for the ADCHnADCR2 and the ADDNT setting when CNVCLS[2:0] = 4_H or 6_H. The format of data transferred to the Digital Filter Engine (DFE), ADC Summation Function (ASF), and Integer/Floating-point Conversion module (IFC) also depends on the above settings.

When unsigned fixed-point format is selected, the format of the data transferred to the DFE, ASF and IFC is signed fixed-point format.

ADCHnDRj is initialized to 0000_H at reset.

A method of rounding from 12-bit to 10-bit resolution is described below.

Table 34.45 A rounding method from 12-bit to 10-bit resolution

Addition	12 bit A/D conversion value			10 bit A/D conversion value
	[11:2]	[1]	[0]	[9:0]
None (One time conversion)	3FF _H	Round-down	Round-down	12 bit A/D conversion value[11:2]
	3FE _H to 000 _H	This bit is added to [11:2]		12 bit A/D conversion value[11:2] + 12 bit A/D conversion value[1]
Addition	12 bit A/D conversion value			10 bit A/D conversion value
	[12:2]	[1]	[0]	[10:0]
Twice	7FF _H	Round-down	Round-down	12 bit A/D conversion value[12:2]
	7FE _H to 000 _H	This bit is added to [12:2]		12 bit A/D conversion value[12:2] + 12 bit A/D conversion value[1]
Addition	12 bit A/D conversion value			10 bit A/D conversion value
	[13:2]	[1]	[0]	[11:0]
4 times	FFF _H	Round-down	Round-down	12 bit A/D conversion value[13:2]
	FFE _H to 000 _H	This bit is added to [13:2]		12 bit A/D conversion value[13:2] + 12 bit A/D conversion value[1]

The format of ADCHnDRj or ADCHnDIRj by the combination of the data format (ADCHnADCR2.DFMT [1:0]), the addition mode (ADCHnVCRj.CNVCLS [2:0]) and the number of addition times (ADCHnADCR2.ADDNT) is described below.

When the data format (ADCHnADCR2.DFMT [1:0]), the addition mode (ADCHnVCRj.CNVCLS [2:0]) and the number of addition times (ADCHnADCR2.ADDNT) are rewritten before reading ADCHnDRj or ADCHnDIRj, ADCHnDRj or ADCHnDIRj are read in the rewritten format.

(1) The format when reading from the P-Bus

For signed 12 bit fixed-point format (DFMT[1:0] = 00_B)

Addition	CNVCLS	ADDNT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
None (One time conversion)	≠ 4 _H , 6 _H	—	S													0	0	0
Twice	= 4 _H , 6 _H	0	S														0	0
4 times		1	S															0

Position of decimal point ↑

For signed 12 bit integer point format (DFMT[1:0] = 01_B)

Addition	CNVCLS	ADDNT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
None (One time conversion)	≠ 4 _H , 6 _H	—	S	S	S	S												
Twice	= 4 _H , 6 _H	0	S	S	S													
4 times		1	S	S														

Position of decimal point ↗

For unsigned 12 bit fixed-point format (DFMT[1:0] = 10_B)

Addition	CNVCLS	ADDNT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
None (One time conversion)	≠ 4 _H , 6 _H	—														0	0	0
Twice	= 4 _H , 6 _H	0															0	0
4 times		1																0

Position of decimal point ↑

For unsigned 10 bit fixed-point format (DFMT[1:0] = 11_B)

Addition	CNVCLS	ADDNT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
None (One time conversion)	≠ 4 _H , 6 _H	—												0	0	0	0	0
Twice	= 4 _H , 6 _H	0													0	0	0	0
4 times		1														0	0	0

Position of decimal point ↑

S : Sign bit (always 0)

0 : Zero extension

(2) The format of the data transferred to the DFE, GTM, ASF, IFC

For unsigned 12 bit fixed-point format (DFMT[1:0] = 00_B/10_B/11_B)

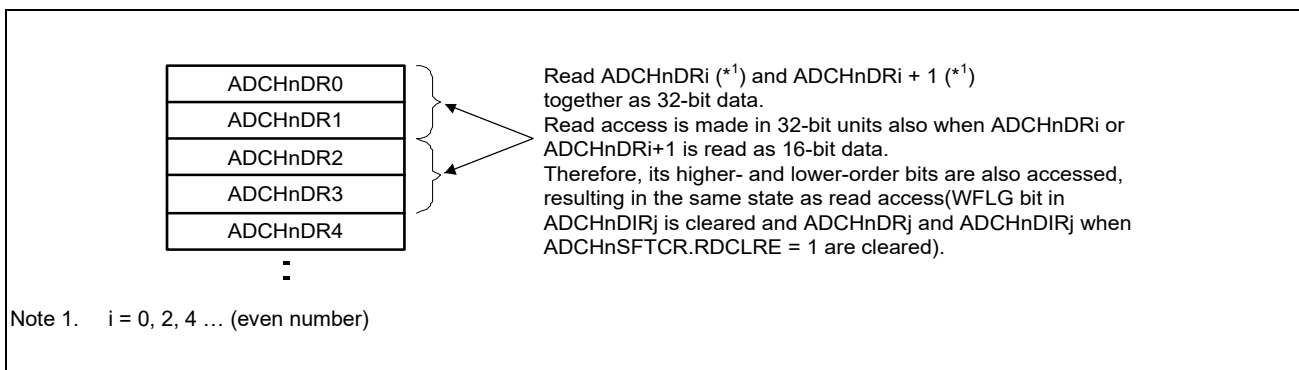
Addition	CNVCLS	ADDNT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
None (One time conversion)	≠ 4 _H , 6 _H	—	S													0	0	0
Twice	= 4 _H , 6 _H	0	S														0	0
4 times		1	S															0

↑
Position of decimal point

- S : Sign bit (always 0)
- 0 : Zero extension
- : When DFMT is 11_B, it's fixed to zero.

ADCHnDRj is cleared to 0000_H when the ACDHnDRj or the ADCHnDIRj is read or ADCHnDRj is read via the IFC while RDCLRE is set to 1.

ADCHnDRi and ADCHnDRi+1 (i = 0, 2, 4, ... (even number)) must always be read together as 32-bit data.



When reading the ADCHnDRj by 16 bits, set the “Read and Clear Enable” to invalid (ADCHnSFTCR.RDCLRE=0) and set “Overwrite check Function” to invalid and then read ADCHnDRj*1.

- Note 1.**
- Don't perform the expected value comparison of the WFLG bit of ADCHnDIRj.
 - Don't perform the expected value comparison of an overwrite error register.
 - Set the “Overwrite Error Interrupt Enable (OWEIE)” of the “Safety Control Register (ADCHnSFTCR)” to 0 (disable).
- Note 2.** When using DFE, ASF and IFC, setting to the signed 12 bit integer point format is prohibited.
- Note 3.** When after one of them of (i) to (v) is done and before A/D conversion value is updated, then ADCHnDRi and ADCHnDRi+1 where the A/D conversion value was stored is read, there is a possibility that parity error which isn't intended occur, so be careful and use.
- (i) The set value of ADCHnADCR2.DFMT[1:0] is changed from one of 0_H, 1_H and 2_H to 3_H.
 - (ii) The set value of ADCHnADCR2.DFMT[1:0] is changed from 3_H to one of 0_H, 1_H and 2_H.
 - (iii) The set value of ADCHnADCR2.ADDNT is changed.
 - (iv) The set value of ADCHnVCRj.CNVCLS[2:0] of corresponding scan group is changed from one of 0_H, 3_H and 5_H to one of 4_H and 6_H.

(v) The set value of ADCHnVCRj.CNVCLS[2:0] of corresponding scan group is changed from one of 4_H and 6_H to one of 0_H, 3_H and 5_H.

Note 4. For ADCHnDRj reading, when ADCHnDRi or ADCHnDRi+1 is read by 16 bits, ADCHnDRi and ADCHnDRi+1 are made the target of reading.

For example, one of them of (i) to (v) is done after scan group completion, then the scan group is performed again in setting of ADCHnSFTCR.RDCLRE=0.

Then if ADCHnDRi is read by 16bits or 32bits before the A/D conversion value is stored in ADCHnDRi+1, there is a possibility that parity error which isn't intended occur.

34.3.3.3 ADCHnDIRj — Data Supplementary Information Register j

ADCHnDIRj is a 32-bit read-only register that stores the supplementary information of ADCHnDRj and the A/D converted value. This register is readable as 32-bit or as 16-bit data.

When reading the ADCHnDIRj by 16 bits, read along with the indicated restrictions for reading of 16 bits as described in on **Section 34.3.3.2, ADCHnDRj — Data Register j**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	IDEF	WFLG	PRTY	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADCHnDRj															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 34.46 ADCHnDIRj Register Contents

Bit Position	Bit Name	Function
31 to 27	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
26	IDEF	ID error 0: There is an error 1: There is no error
25	WFLG	Write Flag Setting condition An A/D converted value is stored in the ADCHnDRj. Clearing conditions ADCHnDRj or ADCHnDIRj is read. ADCHnDRj is read via the IFC.
24	PRTY	Parity Parity bit (even parity) for ADCHnDRj and IDEF.
23 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

The IDEF bit will be 0 at the initial condition or when using a Read-and-Clear or at occurrence of an ID error. When an ID error is not detected by an A/D conversion, the IDEF bit will be 1. The timing for when this bit is set is simultaneous with the timing for which an A/D conversion result is stored in a data register (ADCHnDRj).

The WFLG is cleared when ADCHnDRj or ADCHnDIRj is read or when ADCHnDRj is read via the IFC regardless of the RDCLRE setting.

The IDEF, PRTY and ADCHnDRj[15:0] bits are cleared when ADCHnDRj or ADCHnDIRj is read or when ADCHnDRj is read via the IFC while RDCLRE is set to 1.

34.3.3.4 ADCHnADHALTR — A/D Halt Register

ADCHnADHALTR is an 8-bit write-only register that forcibly terminates the A/D conversion. The register bits are always read as 0.

Value after reset: 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	HALT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 34.47 ADCHnADHALTR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	HALT	Halt All scan groups and all AD timers are halted and initialized, and the ADC changes to the idle state. Writing 0: Scan groups and timers are not halted. Writing 1: Scan groups and timers are halted.

34.3.3.5 ADCHnADCR1 — A/D Control Register 1

ADCHnADCR1 is an 8-bit readable/writable register for ADC common control. ADCHnADCR1 is initialized to 00_H at reset.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SUSMTD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 34.48 ADCHnADCR1 Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	SUSMTD[1:0]	<p>Suspend Method</p> <p>These bits select the suspend method when a higher-priority scan group interrupts a lower-priority scan group.</p> <p>Synchronous suspend: If a request from a higher-priority SG is present while a lower-priority SG is being processed, processing for the lower-priority SG is suspended after the ongoing virtual channel processing is completed, and then processing for the higher-priority SG is executed. After processing for the higher-priority SG is completed, the suspended virtual channel processing for the lower-priority SG is resumed.</p> <p>Asynchronous suspend: If a request from a higher-priority SG is present while a lower-priority SG is being processed, the ongoing virtual channel processing is immediately suspended, and then processing for the higher-priority SG is executed. After processing for the higher-priority SG is completed, the suspended virtual channel processing for the lower-priority SG is resumed.</p> <p>0_H: Synchronous suspend 1_H: Asynchronous suspend when a higher-priority SG interrupts SG0, Synchronous suspend when a higher-priority SG interrupts a lower-priority SG (except for SG0) 2_H: Asynchronous suspend 3_H: Setting prohibited</p> <p>For details, see Figure 34.11 and Figure 34.12.</p>

CAUTION

To prevent a malfunction, there are the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.

34.3.3.6 ADCHnMPXCURCR — MPX Current Control Register

ADCHnMPXCURCR is an 8-bit readable/writable register that controls the ADCHnMPXCURR format.

ADCHnMPXCURCR is initialized to 00_H at reset.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	MSKCFMT[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.49 ADCHnMPXCURCR Register Contents

Bit Position	Bit Name	Function
7 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	MSKCFMT[3:0]	MSKC Format Specification These bits specify the MSKC[15:0] format of ADCHnMPXCURR. MSKCFMT[3] 0: MSKC[15:12] = 0000 _B 1: MSKC[15:12] = 1111 _B MSKCFMT[2] 0: MSKC[11:8] = 0000 _B 1: MSKC[11:8] = 1111 _B MSKCFMT[1] 0: MSKC[7:4] = 0000 _B 1: MSKC[7:4] = 0001 _B MSKCFMT[0] 0: MSKC[3:0] = 0000 _B 1: MSKC[3:0] = 1111 _B

CAUTION

To prevent a malfunction, there are the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.

34.3.3.7 ADCHnMPXINTER — MPX Interrupt Enable Register

ADCHnMPXINTER is an 8-bit readable/writable register that controls the interrupt (ADMPXIn) output.

ADCHnMPXINTER is initialized to 00_H at reset.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADMPXIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 34.50 ADCHnMPXINTER Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ADMPXIE	MPX interrupt enable This bits controls the ADMPXIn interrupt (ADMPXIn) output. 0: ADMPXIn is not output 1: ADMPXIn is output

CAUTION

To prevent a malfunction, there is the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update.**

34.3.3.8 ADCHnMPXCURR — MPX Current Register

ADCHnMPXCURR is a 32-bit read-only register that stores the MPX value for an external analog multiplexer. ADCHnMPXCURR is initialized to 0000 0000_H at reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MSKC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MPXCMD[7:0]							—	—	—	MPXCUR[4:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 34.51 ADCHnMPXCURR Register Contents

Bit Position	Bit Name	Function
31 to 16	MSKC[15:0]	Mask Control When a virtual channel in which CNVCLS[2:0] in ADCHnVCRj is set to 5 _H or 6 _H is started, value of the format that depends on the MSKCFMT[3:0] setting of ADCHnMPXCURCR is transferred to MSKC[15:0] in ADCHnMPXCURR. For details, see Section 34.3.3.6, ADCHnMPXCURCR — MPX Current Control Register .
15 to 8	MPXCMD[7:0]	SPI Communication Command Information A command information register to control an external analog multiplexer by using SPI communication. When a virtual channel in which CNVCLS[2:0] in ADCHnVCRj is set to 5 _H or 6 _H is started, the MPXCMD[7:0] value in ADCHnMPXCMDR is transferred to MPXCMD[7:0] in ADCHnMPXCURR.
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	MPXCUR[4:0]	Current MPX Value When a virtual channel for which CNVCLS[2:0] in ADCHnVCRj is set to 5 _H or 6 _H is started, GCTRL[4:0] in ADCHnVCRj is transferred to MPXCUR[4:0]. At this time, an interrupt request to the INTC or a DMA transfer request is generated. The DMAC transfers ADCHnMPXCURR to PSRn of the I/O port or to CSIHnTX0H of CSIH, enabling the MPX value to be sent to an external analog multiplexer. When PSRn is used, transfer the MPX value as a 32-bit value. This enables rewriting of only the necessary ports by using the format control in MSKC[15:0]. When transferring the value to CSIHnTX0H, transfer the lower 16 bits. For details, see Section 34.4.4.4, Normal A/D Conversion with the MPX and Normal A/D Conversion with the MPX in Addition Mode .

34.3.3.9 ADCHnMPXOWR — MPX Optional Wait register

ADCHnMPXOWR is an 8-bit readable/writable register that specifies the wait time to be inserted for an external analog multiplexer. ADCHnMPXOWR is initialized to 00_H at reset.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	MPXOW[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.52 ADCHnMPXOWR Register Contents

Bit Position	Bit Name	Function
7 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	MPXOW[3:0]	MPX Optional Wait These bits specify the wait time to be inserted before an A/D conversion is started after a virtual channel for which CNVCLS[2:0] in ADCHnVCRj is 5 _H or 6 _H is started. 0 _H : 0 μs 1 _H : 1 μs 2 _H : 2 μs 3 _H : 3 μs 4 _H : 4 μs 5 _H : 5 μs 6 _H : 6 μs 7 _H : 7 μs 8 _H : 8 μs 9 _H : 9 μs A _H : 10 μs B _H to F _H : Setting prohibited For details, see Section 34.4.4.4, Normal A/D Conversion with the MPX and Normal A/D Conversion with the MPX in Addition Mode.

CAUTION

To prevent a malfunction, there is the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update.**

34.3.3.10 ADCHnMPXCMDR — MPX Command Information Register

ADCHnMPXCMDR is an 8-bit readable/writable register that stores SPI communication command information to be transferred to an external analog multiplexer. ADCHnMPXCMDR is initialized to 00_H at reset.

Value after reset: 00_H

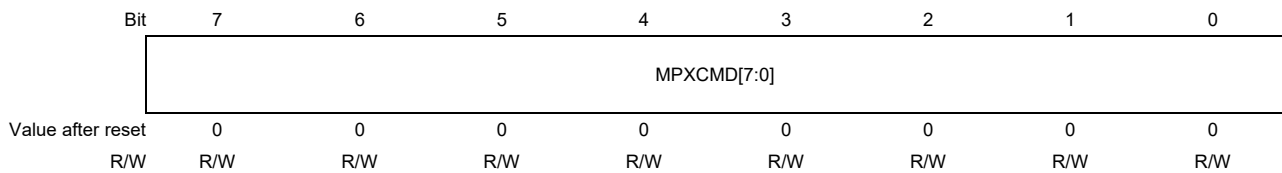


Table 34.53 ADCHnMPXCMDR Register Contents

Bit Position	Bit Name	Function
7 to 0	MPXCMD[7:0]	SPI Communication Command Information These bits store command information for controlling an external analog multiplexer by using the SPI communication. When a virtual channel for which CNVCLS[2:0] in ADCHnVCRj is set to 5 _H or 6 _H is started, MPXCMD[7:0] in ADCHnMPXCMDR is transferred to MPXCMD[7:0] in ADCHnMPXCURR and it can be read by ADCHnMPXCURR together with MPXCUR[4:0].

CAUTION

To prevent a malfunction, there is the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update.**

34.3.3.11 ADCHnADCR2 — A/D Control Register 2

ADCHnADCR2 is an 8-bit readable/writable register for ADC common control.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	DFMT[1:0]		—	—	—	ADDNT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R/W

Table 34.54 ADCHnADCR2 Register Contents

Bit Position	Bit Name	Function
7 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5 to 4	DFMT[1:0]	Data Format 00 _B : Signed 12bit fixed-point format 01 _B : Signed 12bit integer format 10 _B : Unsigned 12bit fixed-point format 11 _B : Unsigned 10bit fixed-point format For data transferred to the DFE.GTM and ASF, it is always signed fixed-point format regardless of the DFMT setting. For details of data format, see Section 34.3.3.2, ADCHnDRj — Data Register j .
3 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ADDNT	Addition Count Select 0: Add twice 1: Add 4 times This register is valid only when CNVCLS[2:0] is 4 _H or 6 _H .

CAUTION

To prevent a malfunction, there are the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.

34.3.3.12 ADCHnDFASENTSGER — DFE/ASF Entry Scan Group Enable Register

ADCHnDFASENTSGER is a 16-bit readable/writable register that enables or disables scan groups to be transferred to the DFE, and the ASF. ADCHnDFASENTSGER is initialized to 0000_H at reset.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	DFENT SG4E	DFENT SG3E	DFENT SG2E	DFENT SG1E	DFENT SG0E	—	—	—	ASENT SG4E	ASENT SG3E	ASENT SG2E	ASENT SG1E	ASENT SG0E
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

Table 34.55 ADCHnDFASENTSGER Register Contents

Bit Position	Bit Name	Function
15 to 13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	DFENTSGxE	DFE Entry Scan Group Enable 0: Entry to DFE is disabled when starting SGx. 1: Entry to DFE is enabled when starting SGx. Entry is performed for virtual channels for which DFENT in ADCHnVCRj is set to 1.
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	ASENTSGxE	ASF Entry Scan Group Enable 0: Entry to ASF is disabled when starting SGx. 1: Entry to ASF is enabled when starting SGx. Entry is performed for virtual channels for which DFENT in ADCHnVCRj is set to 1. Use the ASENTSGxE bits by setting one of these bits to 1.

CAUTIONS

- To prevent a malfunction, there is the limitations of setting update in this register.
For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.
- Do not set both DFENTSGxE and ASENTSGxE for the same scan group to 1. If it is enabled for the same scan group, entry is performed for both DFE and ASF of virtual channels for which DFENT in ADCHnVCRj is set to 1 for the scan group.

34.3.3.13 ADCHnADENDP — A/D Conversion Monitor Virtual Channel Pointer

ADCHnADENDP is an 8-bit readable/writable register that selects a virtual channel that outputs the A/D conversion timing to ADENDn. ADCHnADENDP is initialized to 00_H at reset.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0	
	—	—	ENDP[5:0]						
Value after reset	0	0	0	0	0	0	0	0	
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	

Table 34.56 ADCHnADENDP Register Contents

Bit Position	Bit Name	Function
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	ENDP[5:0]	A/D Conversion Monitor Virtual Channel Pointer Set virtual channel number which monitor ongoing A/D conversion. 00 _H to 27 _H : Virtual channel 0 to 39 Other than above: Setting prohibited When the virtual channel selected by ADCHnADENDP is started, a high level is output to ADENDn. When the virtual channel selected by ADCHnADENDP ends, a low level is output.

CAUTION

To prevent a malfunction, there is the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update.**

34.3.3.14 ADCHnSFTCR — Safety Control Register

ADCHnSFTCR is an 8-bit readable/writable register for safety control. ADCHnSFTCR is initialized to 00_H at reset.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	RDCLRE	—	OWEIE	PEIE	IDEIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R/W	R/W	R/W

Table 34.57 ADCHnSFTCR Register Contents

Bit Position	Bit Name	Function
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	RDCLRE	Read and Clear Enable 0: ADCHnDRj and ADCHnDIRj are not cleared by reading ADCHnDRj or ADCHnDIRj or by reading ADCHnDRj via the IFC. 1: ADCHnDRj and ADCHnDIRj are cleared by reading ADCHnDRj or ADCHnDIRj or by reading ADCHnDRj via the IFC. CAUTION: WFLG in ADCHnDIRj is cleared by reading ADCHnDRj or ADCHnDIRj or by reading ADCHnDRj via the IFC regardless of the RDCLRE setting.
3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	OWEIE	Overwrite Error Interrupt Enable 0: Disabled 1: Enabled
1	PEIE	Parity Error Interrupt Enable 0: Disabled 1: Enabled
0	IDEIE	ID Error Interrupt Enable 0: Disabled 1: Enabled

CAUTION

To prevent a malfunction, there is the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.

34.3.3.15 ADCHnTDCR — Pin-Level Self-Diagnosis Control Register

ADCHnTDCR is an 8-bit readable/writable register that controls the pin-level self-diagnosis. ADCHnTDCR is initialized to 00_H at reset.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	TDE	—	—	—	—	—	TDLV[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R/W	R/W

Table 34.58 ADCHnTDCR Register Contents

Bit Position	Bit Name	Function
7	TDE	Pin-Level Self-Diagnosis 0: Pin-level self-diagnosis is disabled. 1: Pin-level self-diagnosis is enabled. When TDE is set to 1, all analog pins are disconnected from the input buffer. When TDE is set to 0, all analog pins are connected to the input buffer. When TDE is set to 1, the voltage is fixed to the level specified by TDLV[1:0]. Performing an A/D conversion in this state and checking the A/D converted value allows diagnosis of the path from an analog pin to the ADC.
6 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TDLV[1:0]	Pin-Level Self-Diagnosis 0 _H : Even numbers of physical channel groups are discharged to AnVSS, and odd numbers of physical channel groups are charged to AnVCC. 1 _H : Even numbers of physical channel groups are charged to AnVCC, and odd numbers of physical channel groups are discharged to AnVSS. 2 _H : Even numbers of physical channel groups are discharged to AnVSS, and odd numbers of physical channel groups are charged to 1/2*AnVCC. 3 _H : Even numbers of physical channel groups are charged to 1/2*AnVCC, and odd numbers of physical channel groups are discharged to AnVSS.

CAUTION

To prevent a malfunction, there is the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.

34.3.3.16 ADCHnODCR — Wiring-break Detection Control Register

ADCHnODCR is a 32-bit readable/writable register that controls wiring-break detection.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ODDE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ODE	—	ODPW[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.59 ADCHnODCR Register Contents

Bit Position	Bit Name	Function
31	ODDE	Wiring-break Detection Self-diagnosis Enable 0: Wiring-break detection self-diagnosis is not enabled. 1: Wiring-break detection self-diagnosis is enabled. When ODDE is set to 1, wiring-break detection self-diagnosis is enabled for all analog pins.
30 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	ODE	Wiring-break Detection Enable 0: Wiring-break detection is not enabled. 1: Wiring-break detection is enabled. When ODE is set to 1, wiring-break detection is enabled for all analog pins. After sampling for an A/D conversion ends, analog pins for which an A/D conversion is to be performed are discharged with the pulse width specified in ODPW[5:0].
6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	ODPW[5:0]*1	Wiring-break Detection Pulse Width 04 _H : 1 state (of internal clock) 05 _H : 2 states (of internal clock) : 13 _H : 16 states (of internal clock) 14 _H : 17 states (of internal clock)

Note 1. The setting of ODPW[5:0] must be greater than 03_H and less than 15_H.

CAUTION

To prevent a malfunction, there is the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.

34.3.3.17 ADCHnECR — Error Clear Register

ADCHnECR is an 8-bit write-only register that controls error clearing. The register bits are always read as 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	OWEC	PEC	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	W	W	R

Table 34.60 ADCHnECR Register Contents

Bit Position	Bit Name	Function
7 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	OWEC	Overwrite Error Clear Writing 0: Not cleared Writing 1: Cleared
1	PEC	Parity Error Clear Writing 0: Not cleared Writing 1: Cleared
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

34.3.3.18 ADCHnOWER — Overwrite Error Register

ADCHnOWER is an 8-bit read-only register that indicates an overwrite error.

ADCHnOWER is initialized to 00_H at reset.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0	
	OWE	—	OWECAP[5:0]						
Value after reset	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	

Table 34.61 ADCHnOWER Register Contents

Bit Position	Bit Name	Function
7	OWE	Overwrite Error 0: No error is present 1: An error is present. Setting condition The A/D converted value is written to ADCHnDRj when WFLG = 1. Clearing condition A value of 1 is written to OWEC in ADCHnECR.
6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	OWECAP[5:0]	Overwrite Error Capture The virtual channel at the time when an overwrite error occurred is captured. Capturing condition The A/D converted value is written to ADCHnDRj when OWE = 0 and WFLG = 1. Clearing condition A value of 1 is written to OWEC in ADCHnECR.

CAUTION

ADCHnOWER is updated when the A/D converted value is written to ADCHnDRj.

34.3.3.19 ADCHnPER — Parity Error Register

ADCHnPER is an 8-bit read-only register that indicates a parity error. ADCHnPER is initialized to 00_H at reset.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0	
	PE	—	PECAP[5:0]						
Value after reset	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	

Table 34.62 ADCHnPER Register Contents

Bit Position	Bit Name	Function
7	PE	Parity Error 0: No errors present 1: An error is present. Setting condition A parity error is detected. Clearing condition A value of 1 is written to PEC in ADCHnECR.
6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	PECAP[5:0]	Parity Error Capture The virtual channel at the time when a parity error occurred is captured. Capturing condition A parity error is detected when PE = 0. Clearing condition A value of 1 is written to PEC in ADCHnECR.

CAUTION

ADCHnPER is updated when ADCHnDRj or ADCHnDIRj is read, but is not updated when ADCHnDRj is read via the IFC.

34.3.3.20 ADCHnVCULLMTBR0 to 6 — Virtual Channel Upper/Lower Limit Table Register 0 to 6

ADCHnVCULLMTBR0 to ADCHnVCULLMTBR6 are 32-bit readable/writable registers. ADCHnVCULLMTBR0 to ADCHnVCULLMTBR6 are selected from ADCHnVCRj.VCULLMTBS[2:0]. These registers are initialized to 7FFE 0000_H by reset.

Value after reset: 7FFE 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VCULMTB[15:0]															
Value after reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VCLLMTB[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 34.63 ADCHnVCULLMTBR Register Contents 0 to 6

Bit Position	Bit Name	Function
31 to 16	VCULMTB[15:0]	<p>Upper Limit Table</p> <p>These bits specify the upper-limit value of an A/D converted value.</p> <p>Setting range: 0000_H to 7FFE_H</p> <p>Be sure to set it to the signed fixed-point format.</p> <p>VCULMTB[15] and VCULMTB[0] are always fixed to 0.</p> <p>When in normal-A/D-conversion-in-addition-mode (ADCHnVCRj.CNVCLS[2:0] = 4_H), or normal-A/D-conversion-with-the-MPX-in-addition-mode (ADCHnVCRj.CNVCLS[2:0] = 6_H), set the comparative value to the addition sum.</p>
15 to 0	VCLLMTB[15:0]	<p>Lower Limit Table</p> <p>These bits specify the lower-limit value of an A/D converted value.</p> <p>Setting range: 0000_H to 7FFE_H</p> <p>Be sure to set it to the signed fixed-point format.</p> <p>VCLLMTB[15] and VCLLMTB[0] are always fixed to 0.</p> <p>When in normal-A/D-conversion-in-addition-mode (ADCHnVCRj.CNVCLS[2:0] = 4_H), or normal-A/D-conversion-with-the-MPX-in-addition-mode (ADCHnVCRj.CNVCLS[2:0] = 6_H), set the comparative value to the addition sum.</p>

34.3.3.21 ADCHnVCLMINTER1 — Virtual Channel Upper/Lower Limit Error Interrupt Enable Register 1

ADCHnVCLMINTER1 is a 32-bit readable/writable register for output control of the upper/lower limit error interrupt for virtual channels 0 to 31. ADCHnVCLMINTER1 is initialized to 0000 0000_H at reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADUL3 1IE	ADUL3 0IE	ADUL2 9IE	ADUL2 8IE	ADUL2 7IE	ADUL2 6IE	ADUL2 5IE	ADUL2 4IE	ADUL2 3IE	ADUL2 2IE	ADUL2 1IE	ADUL2 0IE	ADUL1 9IE	ADUL1 8IE	ADUL1 7IE	ADUL1 6IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADUL1 5IE	ADUL1 4IE	ADUL1 3IE	ADUL1 2IE	ADUL1 1IE	ADUL1 0IE	ADUL0 9IE	ADUL0 8IE	ADUL0 7IE	ADUL0 6IE	ADUL0 5IE	ADUL0 4IE	ADUL0 3IE	ADUL0 2IE	ADUL0 1IE	ADUL0 0IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.64 ADCHnVCLMINTER1 Register Contents

Bit Position	Bit Name	Function
31 to 0	ADULjIE	Virtual channel upper / lower limit error enable 0: The upper / lower limit error interrupt (INT_ULn) is not output by result of the virtual channel j. 1: The upper / lower limit error interrupt (INT_ULn) is output by the result of the virtual channel j.

CAUTION

To prevent a malfunction, there is the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update.**

34.3.3.22 ADCHnVCLMINTER2 — Virtual Channel Upper/Lower Limit Error Interrupt Enable Register 2

ADCHnVCLMINTER2 is a 32-bit readable/writable register for output control of the upper/lower limit error interrupt for virtual channels 32 to 39. ADCHnVCLMINTER2 is initialized to 0000 0000_H at reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ADUL3 9IE	ADUL3 8IE	ADUL3 7IE	ADUL3 6IE	ADUL3 5IE	ADUL3 4IE	ADUL3 3IE	ADUL3 2IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.65 ADCHnVCLMINTER2 Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	ADULjIE	Virtual channel upper / lower limit error enable 0: The upper / lower limit error interrupt (INT_ULn) is not output by the result of the virtual channel j. 1: The upper / lower limit error interrupt (INT_ULn) is output by the result of the virtual channel j.

CAUTION

To prevent a malfunction, there is the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.

34.3.3.23 ADCHnVCLMSR1 — Virtual Channel Upper/Lower Limit Excess Status Register 1

ADCHnVCLMSR1 is a 32-bit read-only register that indicates an upper/lower limit excursion of virtual channel 0 to 31. ADCHnVCLMSR1 is initialized to 0000 0000_H at reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VC31L MS	VC30L MS	VC29L MS	VC28L MS	VC27L MS	VC26L MS	VC25L MS	VC24L MS	VC23L MS	VC22L MS	VC21L MS	VC20L MS	VC19L MS	VC18L MS	VC17L MS	VC16L MS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VC15L MS	VC14L MS	VC13L MS	VC12L MS	VC11L MS	VC10L MS	VC09L MS	VC08L MS	VC07L MS	VC06L MS	VC05L MS	VC04L MS	VC03L MS	VC02L MS	VC01L MS	VC00L MS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 34.66 ADCHnVCLMSR1 Register Contents

Bit Position	Bit Name	Function
31 to 0	VCjLMS	Virtual channel j upper / lower limit status 0: Limit has not been passed. 1: Limit has been passed. Setting condition The upper limit or the lower limit excursion has been detected. Clearing condition A value of 1 is written to VCjLMSC of ADCHnVCLMSR1.

34.3.3.24 ADCHnVCLMSR2 — Virtual Channel Upper/Lower Limit Excess Status Register 2

ADCHnVCLMSR2 is a 32-bit read-only register that indicates an upper/lower limit excursion of virtual channel 32 to 39. ADCHnVCLMSR2 is initialized to 0000 0000_H at reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	VC39L MS	VC38L MS	VC37L MS	VC36L MS	VC35L MS	VC34L MS	VC33L MS	VC32L MS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 34.67 ADCHnVCLMSR2 Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	VCjLMS	Virtual channel j upper / lower limit excursion status 0: The limit has not been passed. 1: The limit has been passed. Setting condition The upper limit or the lower limit excursion has been is detected. Clearing condition A value of 1 is written to VCjLMSC of ADCHnVCLMSR2.

34.3.3.25 ADCHnVCLMSR1 — Virtual Channel Upper/Lower Limit Excess Status Clear Register 1

ADCHnVCLMSR1 is a 32-bit write-only register that clears virtual channel upper/lower limit excursion status register1 (ADCHnVCLMSR1). The register bits are always read as 0.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VC31L MSC	VC30L MSC	VC29L MSC	VC28L MSC	VC27L MSC	VC26L MSC	VC25L MSC	VC24L MSC	VC23L MSC	VC22L MSC	VC21L MSC	VC20L MSC	VC19L MSC	VC18L MSC	VC17L MSC	VC16L MSC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VC15L MSC	VC14L MSC	VC13L MSC	VC12L MSC	VC11L MSC	VC10L MSC	VC09L MSC	VC08L MSC	VC07L MSC	VC06L MSC	VC05L MSC	VC04L MSC	VC03L MSC	VC02L MSC	VC01L MSC	VC00L MSC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 34.68 ADCHnVCLMSR1 Register Contents

Bit Position	Bit Name	Function
31 to 0	VCjLMS	Virtual channel j upper / lower limit excursion status clear 0: Not cleared. 1: VCjLMS of ADCHnVCLMSR1 is cleared.

34.3.3.26 ADCHnVCLMSCR2 — Virtual Channel Upper/Lower Limit Excess Status Clear Register 2

ADCHnVCLMSCR2 is a 32-bit write-only register that clears the virtual channel upper/lower limit excursion status register2 (ADCHnVCLMSR2). The register bits are always read as 0.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	VC39L MSC	VC38L MSC	VC37L MSC	VC36L MSC	VC35L MSC	VC34L MSC	VC33L MSC	VC32L MSC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 34.69 ADCHnVCLMSCR2 Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	VCjLMSC	Virtual channel j upper / lower limit excess status clear 0: Not cleared. 1: VCjLMS of ADCHnVCLMSR2 is cleared.

34.3.4 Scan Group Specific Registers

This section describes registers provided for each scan group.

34.3.4.1 ADCHnSGSTCRx — Scan Group x Start Control Register

ADCHnSGSTCRx is an 8-bit write-only register that controls the start of scan group x. The register bits are always read as 0.

Value after reset: 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SGST
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 34.70 ADCHnSGSTCRx Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SGST	Scan Group Start Condition for starting scan group x: A value of 1 is written to SGST when SGACTION = 0 Writing 0 to this bit is ignored.

34.3.4.2 ADCHnSGSTPCR_x — Scan Group x Stop Control Register

ADCHnSGSTPCR_x is an 8-bit write-only register that controls the stop of scan group x. The register bits are always read as 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SGSTP
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 34.71 ADCHnSGSTPCR_x Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SGSTP	Scan Group Stop Condition for stopping scan group x: A value of 1 is written to SGSTP when SGACTION = 1 When SGACTION = 0, writing 1 to this bit is ignored. Writing 0 to this bit is ignored.

34.3.4.3 ADCHnADTSTCRy — A/D Timer y Start Control Register

ADCHnADTSTCRy is an 8-bit write-only register that controls the start of A/D timer y. The register bits are always read as 0.

Value after reset: 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADTST
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 34.72 ADCHnADTSTCRy Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ADTST	A/D Timer Start Condition for starting A/D timer y: A value of 1 is written to ADTST when ADTACT = 0 Writing 0 to this bit is ignored.

34.3.4.4 ADCHnADTENDCRy — A/D Timer y End Control Register

ADCHnADTENDCRy is an 8-bit write-only register that controls stop of A/D timer y. The register bits are always read as 0.

Value after reset: 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADTEND
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 34.73 ADCHnADTENDCRy Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ADTEND	A/D Timer End Condition for finishing A/D timer y: A value of 1 is written to ADTEND when ADTACT = 1 Writing 0 to this bit is ignored.

34.3.4.5 ADCHnSGCRx — Scan Group x Control Register

ADCHnSGCRx is an 8-bit readable/writable register that controls scan group x. ADCHnSGCRx is initialized to 00_H at reset.

Value after reset: 00_H

• When x = 0 to 2

Bit	7	6	5	4	3	2	1	0
	—	ADSTARTE	SCANMD	ADIE	—	—	—	TRGMD
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R	R/W

Table 34.74 ADCHnSGCRx Register Contents (x = 0 to 2)

Bit Position	Bit Name	Function
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	ADSTARTE	Scan Group Synchronization Start Enable 0: ADSTART is disabled. 1: ADSTART is enabled.
5	SCANMD	Scan Mode 0: Multicycle scan mode 1: Continuous scan mode In multicycle scan mode, scans are repeated as many times as specified in ADCHnSGMICYCRx. In continuous scan mode, scans are repeated with no limit of times.
4	ADIE	Scan End Interrupt Enable 0: ADInx is not output at the end of scan for SGx. 1: ADInx is output at the end of scan for SGx. ADIE of ADCHnSGCRx is independent of ADIE of ADCHnVCRj. For details, see Section 34.4.6.1, Scan End Interrupt Request .
3 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	TRGMD	Trigger Mode 0: Trigger input to SGx is disabled. 1: The SGx_TRG hardware trigger is selected for the trigger input to SGx.

• When x = 3 or 4

Bit	7	6	5	4	3	2	1	0
	ADTSTARTE	ADSTARTE	SCANMD	ADIE	—	—	TRGMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Table 34.75 ADCHnSGCRx Register Contents (x = 3 or 4)

Bit Position	Bit Name	Function
7	ADTSTARTE	A/D Timer Synchronization Start Enable 0: ADTSTART is disabled. 1: ADTSTART is enabled.
6	ADSTARTE	Scan Group Synchronization Start Enable 0: ADSTART is disabled. 1: ADSTART is enabled.
5	SCANMD	Scan Mode 0: Multicycle scan mode 1: Continuous scan mode In multicycle scan mode, scans are repeated as many times as specified in ADCHnSGMICYCRx. In continuous scan mode, scans are repeated with no limit of times.
4	ADIE	Scan End Interrupt Enable 0: ADInx is not output at the end of scan for SGx. 1: ADInx is output at the end of scan for SGx. ADIE of ADCHnSGCRx is independent of ADIE of ADCHnVCRj. For details, see Section 34.4.6.1, Scan End Interrupt Request .
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TRGMD[1:0]	Trigger Mode 0 _H : Trigger input to SGx is disabled. Trigger input to A/D timer x is disabled. 1 _H : SGx_TRG hardware trigger is selected for the trigger input to SGx. Trigger input to A/D timer x is disabled. 2 _H : A/D timer trigger x is selected for the trigger input to SGx. Trigger input to A/D timer x is disabled. 3 _H : A/D timer trigger x is selected for the trigger input to SGx. The SGx_TRG hardware trigger is selected for the trigger input to A/D timer x.

CAUTIONS

- To prevent a malfunction, there are limitations of setting update in this register.
For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.
- If a higher-priority scan group is set to continuous scan mode (SCANMD = 1), a lower-priority scan group can not operate.
- Hardware trigger (scan group x start) input to a started scan group x is ignored.

34.3.4.6 ADCHnSGVCSPx — Scan Group x Start Virtual Channel Pointer

See **Section 34.3.4.12, ADCHnSGVCPRx — Scan Group x Virtual Channel Pointer Register** together.

ADCHnSGVCSPx is an 8-bit readable/writable register that specifies the start pointer of a virtual channel.

ADCHnSGVCSPx is initialized to 00_H at reset.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0	
	—	—	VCSP[5:0]						
Value after reset	0	0	0	0	0	0	0	0	
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	

Table 34.76 ADCHnSGVCSPx Register Contents

Bit Position	Bit Name	Function
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	VCSP[5:0]	Start Virtual Channel Pointer These bits select the virtual channel from which the scan is to be started. When an SGx is started, processing for the virtual channels from ADCHnSGVCSPx to ADCHnSGVCEPx is executed.

CAUTIONS

- Do not set the value greater than 39.
- ADCHnSGVCSPx must be equal to or less than ADCHnSGVCEPx.
- This register has a mirror structure as follows. While scan group x is executing, if it is necessary to rewrite SGVCSP and SGVCEP, be sure to use a mirror register.

This Register	Mirror Register
ADCHnVCSPx.VCSP[5:0]	ADCHnVCPRx.VCSP[5:0]

- It is possible to set same virtual channel n (ADCHnVCRj) in more than one scan group x by setting the start / end virtual channel pointer.
But, when a virtual channel is overwritten while one scan group is operating, the setting for A/D conversion might be different from the desired setting for this scan group.

Make sure to use caution of the information below to avoid this problem.

- When designing a system, always specify the same value for the ADCHnVCRj setting when it is used by multiple scan groups for the target channel.
- Rewrite ADCHnVCRj after stopping all scan groups of a target channel.

When these are not obeyed, A/D conversion results can't be guaranteed.

- To prevent a malfunction, there is the limitations of setting update in this register.
For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.

34.3.4.7 ADCHnSGVCEPx — Scan Group x End Virtual Channel Pointer

See **Section 34.3.4.12, ADCHnSGVCPRx — Scan Group x Virtual Channel Pointer Register** together.

ADCHnSGVCEPx is an 8-bit readable/writable register that specifies the end pointer of a virtual channel.

ADCHnSGVCEPx is initialized to 00_H at reset.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0	
	—	—	VCEP[5:0]						
Value after reset	0	0	0	0	0	0	0	0	
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	

Table 34.77 ADCHnSGVCEPx Register Contents

Bit Position	Bit Name	Function
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	VCEP[5:0]	End Virtual Channel Pointer These bits select the virtual channel at which the scan is to be ended. When SGx is started, processing for the virtual channels from ADCHnSGVCSPx to ADCHnSGVCEPx is executed.

CAUTIONS

- This register has a mirror structure as follows. While scan group x is executing, if it is necessary to rewrite SGVCSP and SGVCEP, be sure to use a mirror register.

This Register	Mirror Register
ADCHnVCEPx.VCSP[5:0]	ADCHnVCPRx.VCEP[5:0]

See **Section 34.3.4.6, ADCHnSGVCSPx — Scan Group x Start Virtual Channel Pointer** for other cautions.

- To prevent a malfunction, there is the limitations of setting update in this register.
For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.

34.3.4.8 ADCHnSGMCYCRx — Scan Group x Multicycle Register

ADCHnSGMCYCRx is an 8-bit readable/writable register that specifies the number of scan cycles in multicycle scan mode. ADCHnSGMCYCRx is initialized to 00_H at reset.

Value after reset: 00_H

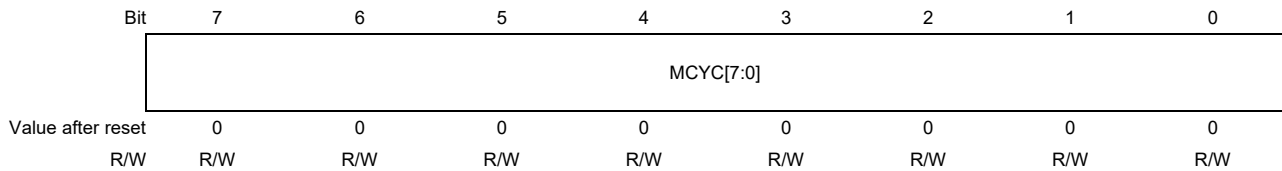


Table 34.78 ADCHnSGMCYCRx Register Contents

Bit Position	Bit Name	Function
7 to 0	MCYC[7:0]	Multicycle Specification These bits specify the number of scan cycles in multicycle scan mode. Number of scan cycles = MCYC[7:0] + 1 When SGx is started, scans are repeated for virtual channels from ADCHnSGVCSPx to ADCHnSGVCEPx for many cycles as specified in ADCHnSGMCYCRx.

CAUTION

To prevent malfunction, there is the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.

34.3.4.9 ADCHnSGSRx — Scan Group x Status Register

ADCHnSGSRx is an 8-bit read-only register that indicates the status of scan group x.

Value after reset: 00_H

• When x = 0 to 2

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SGACT	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 34.79 ADCHnSGSRx Register Contents (x = 0 to 2)

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SGACT	Scan Group Status 0: There is no source in SGx. 1: There is a source in SGx. (The period from the reception of a starting trigger until A/D conversion completion, or until aborting by ADHALT or SGSTP.)
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

• When x = 3 or 4

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADTACT	SGACT	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 34.80 ADCHnSGSRx Register Contents (x = 3, 4)

Bit Position	Bit Name	Function
7 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	ADTACT	A/D Timer Status 0: A/D timer x is in idle state. 1: A/D timer x is running.
1	SGACT	Scan Group Status 0: There is no source in SGx. 1: There is a source in SGx. (The period from the reception of a starting trigger until A/D conversion completion, or until aborting by ADHALT or SGSTP.)
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

34.3.4.10 ADCHnADTIPRy — A/D Timer y Initial Phase Register

ADCHnADTIPRy is a 32-bit readable/writable register that sets the initial count of A/D timer y. ADCHnADTIPRy is initialized to 0000 0000_H at reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	ADTIP[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADTIP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.81 ADCHnADTIPRy Register Contents

Bit Position	Bit Name	Function
31 to 21	-	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20 to 0	ADTIP[20:0]	A/D Timer Initial Phase These bits set the initial phase of A/D timer y. (1) After A/D timer y is started, ADCHnADTIPRy is loaded to A/D timer y and the timer counts down. (2) After A/D timer y becomes 0, A/D timer trigger y is output for one cycle, ADCHnADTPRRy is loaded to A/D timer y, and the timer counts down again. After that, (2) is repeated. For details, see Section 34.4.5.3, A/D Timer Trigger .

CAUTION

To prevent a malfunction, there is the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.

34.3.4.11 ADCHnADTPRRy — A/D Timer y Cycle Register

ADCHnADTPRRy is a 32-bit readable/writable register that sets the cycle count of A/D timer y. ADCHnADTPRRy is initialized to 001F FFFF_H at reset.

Value after reset: 001F FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	ADTPR[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADTPR[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.82 ADCHnADTPRRy Register Contents

Bit Position	Bit Name	Function
31 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20 to 0	ADTPR[20:0]	A/D Timer Cycle These bits set the cycle of A/D timer y. (1) After A/D timer y is started, ADCHnADTIPRy is loaded to A/D timer y and the timer counts down. (2) After A/D timer y becomes 0, A/D timer trigger y is output for one cycle, ADCHnADTPRRy is loaded to A/D timer y, and the timer counts down again. After that, (2) is repeated. For details, see Section 34.4.5.3, A/D Timer Trigger .

CAUTION

To prevent a malfunction, there is the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update**.

34.3.4.12 ADCHnSGVCPRx — Scan Group x Virtual Channel Pointer Register

ADCHnSGVCPRx is a 16-bit readable/writable register that specifies the start/end pointer of a virtual channel. For this register, write as 16-bit.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	VCEP[5:0]						—	—	VCSP[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.83 ADCHnSGVCPRx Register Contents

Bit Position	Bit Name	Function
15 to 14	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	VCEP[5:0] ^{*1}	End Virtual Channel Pointer (mirror of ADCHnSGVCEP)
7 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	VCSP[5:0] ^{*1}	Start Virtual Channel Pointer (mirror of ADCHnSGVCSP)

Note 1. For details of this function, see **Section 34.3.4.6, ADCHnSGVCSPx — Scan Group x Start Virtual Channel Pointer** and **Section 34.3.4.7, ADCHnSGVCEPx — Scan Group x End Virtual Channel Pointer**.

34.3.4.13 ADCHnSGVCOWRx — Scan Group x Virtual Channel Optional Waiting Times Register

ADCHnSGVCOWRx is a 16-bit readable/writable register that specifies an optional wait time in an execution interval of virtual channel j and virtual channel j+1.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	VCOW[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.84 ADCHnSGVCOWRx Register Contents

Bit Position	Bit Name	Function
15 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	VCOW[11:0]	Wait Time Specification Between Virtual Channels of same scan group 000 _H : No wait. 001 _H to FFF _H : set value x 4 ADCLK Note: For example, to set wait times to 1us, set VCOW[11:0] to 00A _H . Wait time, please set the VCOW value in the range of 1...400us at 1us intervals.

CAUTION

To prevent a malfunction, there is the limitations of setting update in this register.

For the limitations of setting update, refer to **Section 34.4.1.4, Limitations of Setting Update.**

34.4 Operation

A/D conversion is performed by scan group.

Virtual channels that are assigned by the scan group start/end pointer are scanned in ascending order.

When multiple scan groups start up, the processing is performed from the scan group with the highest priority.

After the initial settings, scan starts by input of the start trigger.

Scan ends when A/D conversion ends for the specified count or by forced termination.

Information about the settings, functions, and operations is described below.

34.4.1 Setting Procedure

34.4.1.1 Initial Settings

The initial settings must be specified while the trigger of all the scan groups is disabled and all the scan groups and all accumulation channels stop.

If such items shown above are running, perform the termination procedure.

Use the value after reset for the register setting value of the function that is not to be used.

(1) Registers that must be set

The control bit that is required to be set at minimum for executing A/D conversion is defined as a required setting bit.

The required setting bit contains the following bits:

- Bits that determine the operating mode of the SAR-ADC (SM).
- Bits that determine physical channels and virtual channels to be subject to A/D conversion.

Registers that contain the required setting bits are as below:

Table 34.85 Required Registers which must be set for proper operation

Classification	Register	Indispensable Setting Bit	Explanation	Addition
Required	ADCHnADCR1	SUSMTD	Suspend method	—
	ADCHnADCR2	DFMT	Data format	—
	ADCHnVCRj	CNVCLS, GCTRL	Conversion class, General control	Setting is unnecessary for ADCHnVCRj which are not used.
	ADCHnSGCRx	SCANMD	Scan mode	Setting is unnecessary for SGx which are not used.
	ADCHnSGMCYCRx	MCYC[7:0]	The number of multi-cycle-scan times	Setting is unnecessary for SGx that a continuous mode is chosen.
	ADCHnSGVCOWRx	VCOM[11:0]	Optional waiting times of between virtual channels	Setting is unnecessary for SGx that the waiting times aren't inserted between the virtual channels.
At least one of these must be set	ADCHnSGVCSPx, ADCHnSGVCEPx	VCSP, VCEP	Starting / end pointer	Setting is unnecessary for SGx which isn't used.
	ADCHnSGVCPRx			

Please set registers other than these properly according to the function to use.

(2) About Setting Order

Any register setting sequence is acceptable regardless of whether the register contains the required setting bit.

Set registers in the desired order.

The initial setting flow is shown below. The setting order in the flow is for convenience.

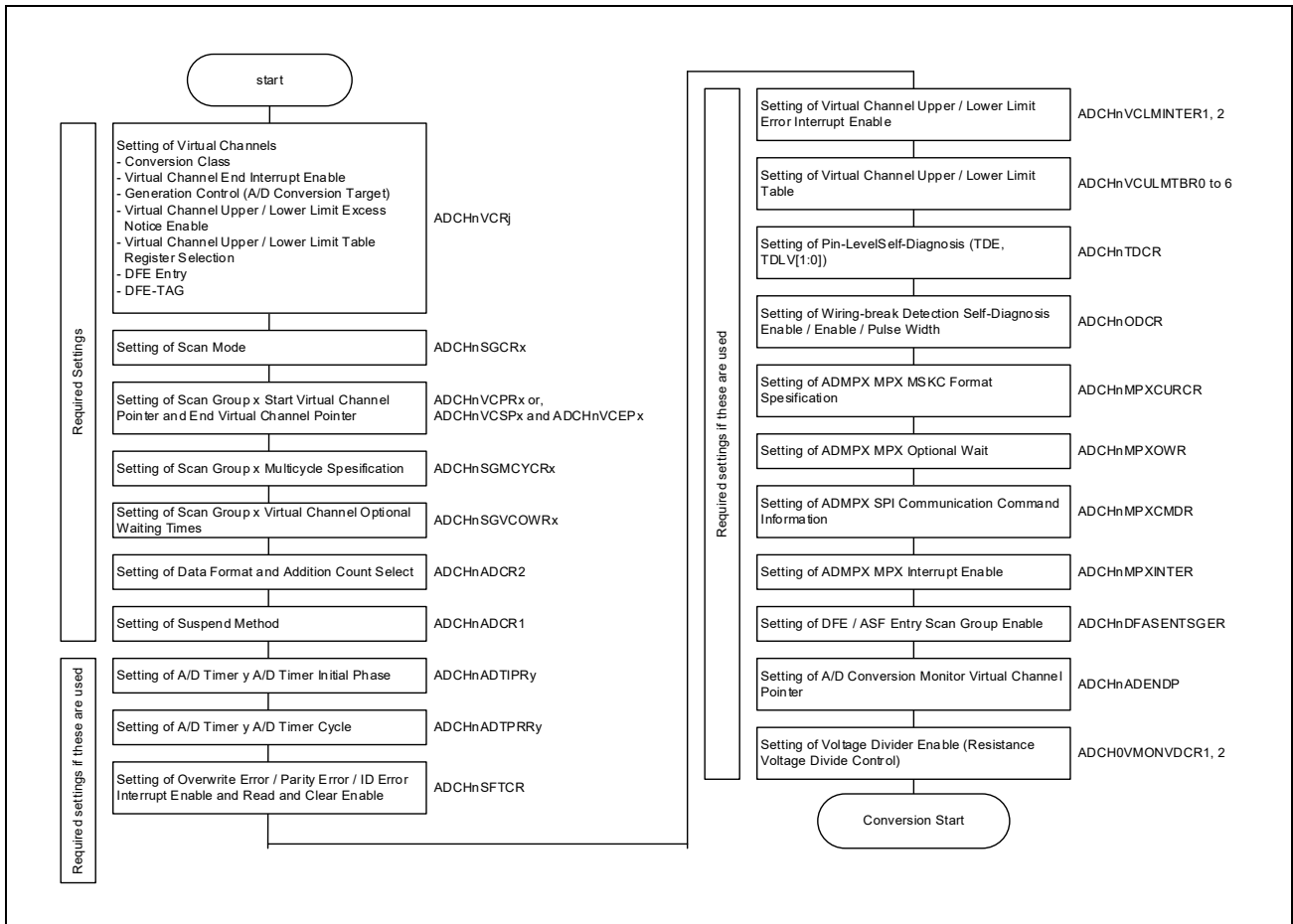


Figure 34.5 Initial Setting Flow

34.4.1.2 Startup Method

This section describes how to start up scan group x, AD timer x.

The procedure differs depending on the startup target.

Perform the startup procedure during the initial setting, or while scan group x to be started is stopped (ADCHnSGSRx.SGACT = 0) after the settings have been updated.

The startup procedure is ignored if the target is already operating.

Table 34.86 Startup Method of Scan Group x, AD Timer y and ASF

Starting target	Trigger type	Starting method
Scan group x	S/W trigger	Write 1 to ADCHnSGSTCRx.SGST.
	H/W trigger	1. Set 1 _H to SGCRx.TRGM [1:0]. 2. Input a High pulse signal of 1ADCLK width to the SGx_TRG terminal.
A/D conversion synchronous start	S/W trigger	1. Set SGCRx.ADSTARTE to 1 for scan group x of ADCH0. 2. Set SGCRx.ADSTARTE to 1 for scan group x of ADCH2. 3. Set SGCRx.ADSTARTE to 1 for scan group x of ADCH1. 4. Set SGCRx.ADSTARTE to 1 for scan group x of ADCH3. 5. Write 1 to ADSYNSTCR.ADSTART.
A/D timer y	S/W trigger	Write 1 to ADTSTCRx.ADTST.
	H/W trigger	1. Set SGCRx.TRGM [1:0] to 3H. 2. Input the H pulse signal with 1 clkad width to the SGx_TRG pin.
A/D timer synchronous start	S/W trigger	1. Set SGCRx.ADTSTARTE to 1 for scan group x of ADCH0. 2. Set SGCRx.ADTSTARTE to 1 for scan group x of ADCH2. 3. Set SGCRx.ADTSTARTE to 1 for scan group x of ADCH1. 4. Set SGCRx.ADTSTARTE to 1 for scan group x of ADCH3. 5. Write 1 to ADTSYNSTCR.ADTSTART.
Setting of ASF	—	See Section 34.8, ADC Summation Function (ASF) .

34.4.1.3 Terminating Procedure

This section describes the procedure to terminate scan groups.

There are three termination methods: termination of all scan groups (forced termination), termination of scan group x 1 and termination of scan group x 2.

For the detailed description of ASFrCTL1 register in each flow, please see **Section 34.8, ADC Summation Function (ASF)**.

(1) To Terminate All Scan Groups (Forced Termination)

This procedure is to terminate all scan groups.

Please clear the multiplication counter in ASF and all intermediate buffers as ASFrCTL1.ASFrST = 0 of ASF when either of scanning group is using ASF.

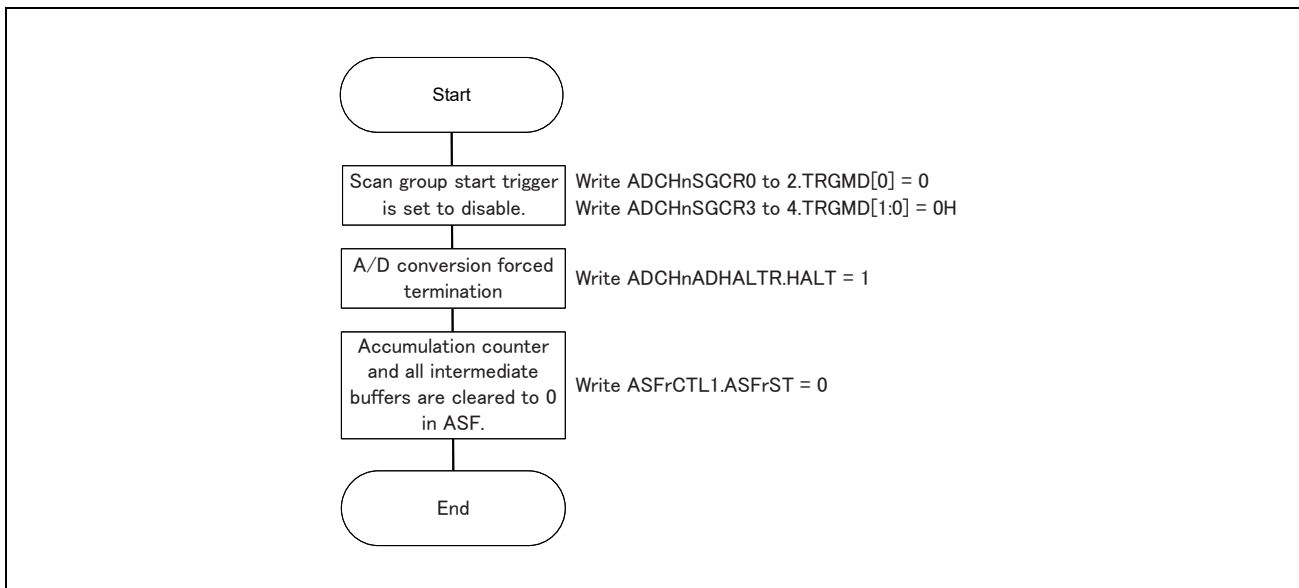


Figure 34.6 All Scan Group (A/D halt) Flow

For the detailed description of termination timing, please see **Section 34.4.14.1, Example of All Scan Group Stop (A/D Halt)**.

(2) Scan Group x Termination Procedure 1

This procedure is to terminate scan group x individually.

To terminate multiple scan groups, perform this termination procedure for each scan group.

This procedure can be applied to scan groups that are operating in multicycle scan mode.

When terminating a scan group that is operating in continuous scan mode, please see **Section 34.4.1.3(1), To Terminate All Scan Groups (Forced Termination)** or **Section 34.4.1.3(3), Scan Group x Termination Procedure 2**.

Please clear the multiplication counter in ASF and all intermediate buffers as `ASFrCTL1.ASFrST=0` of ASF when the scanning group that stops is using ASF.

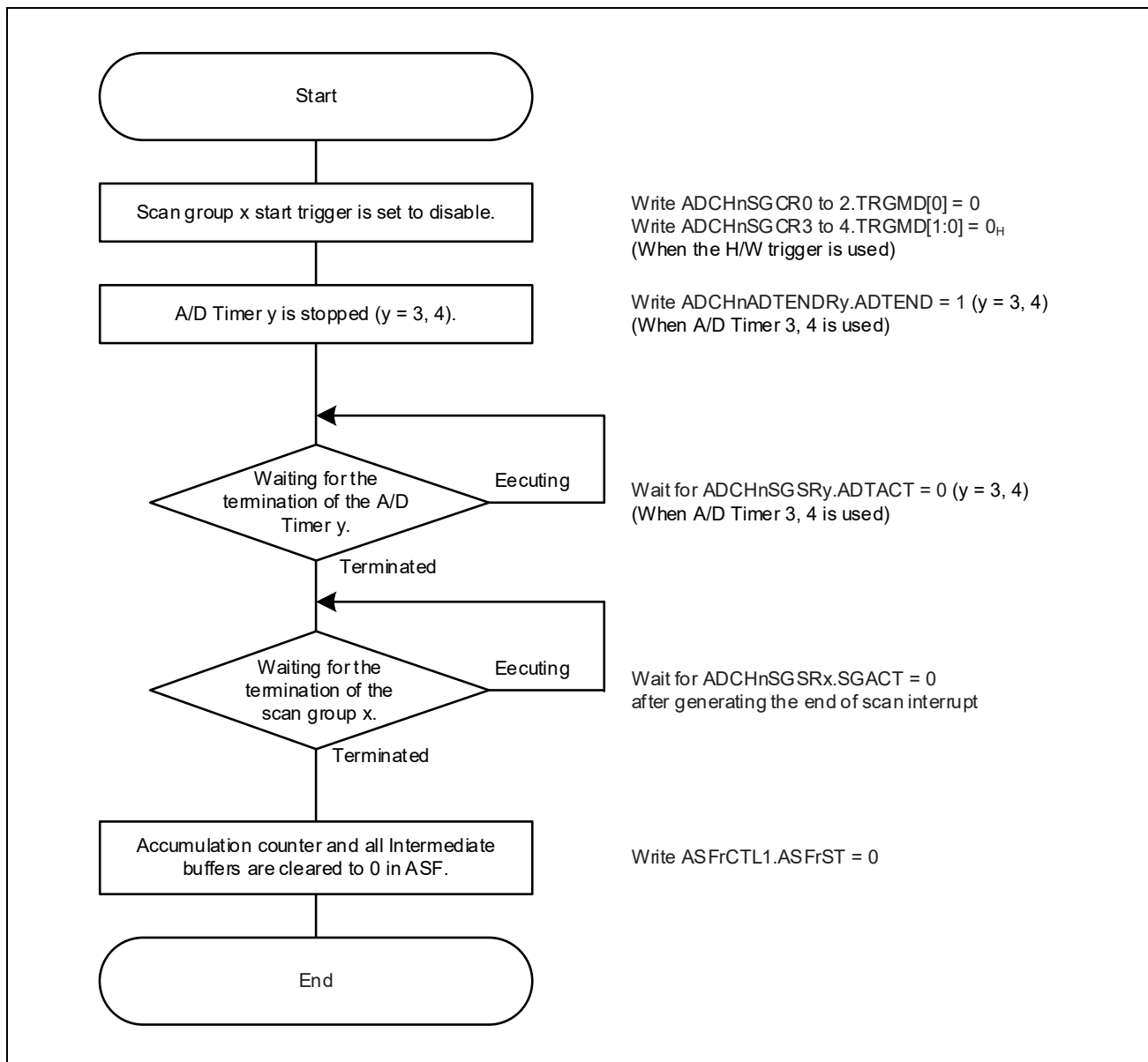


Figure 34.7 Scan Group x Termination Flow 1

(3) Scan Group x Termination Procedure 2

This procedure stops individual scanning group x.

Please execute the shutdown procedure to each scanning group when stopping two or more scanning groups.

This procedure can be applied to the scanning group operating by either of the multi cycle scanning mode/mode of a continuous scanning.

When scan group x is stopped by this procedure, it operates as the following.

When scan group x stop control register is written as 1, an ongoing virtual channel is ended with waiting completion of the remaining processing.

When an ongoing virtual channel is ended, scan group x is ended.

Please clear the multiplication counter in ASF and all intermediate buffers as $ASF1nCTL1.ASF1nST=0$ of ASF when the scanning group that stops is using ASF.

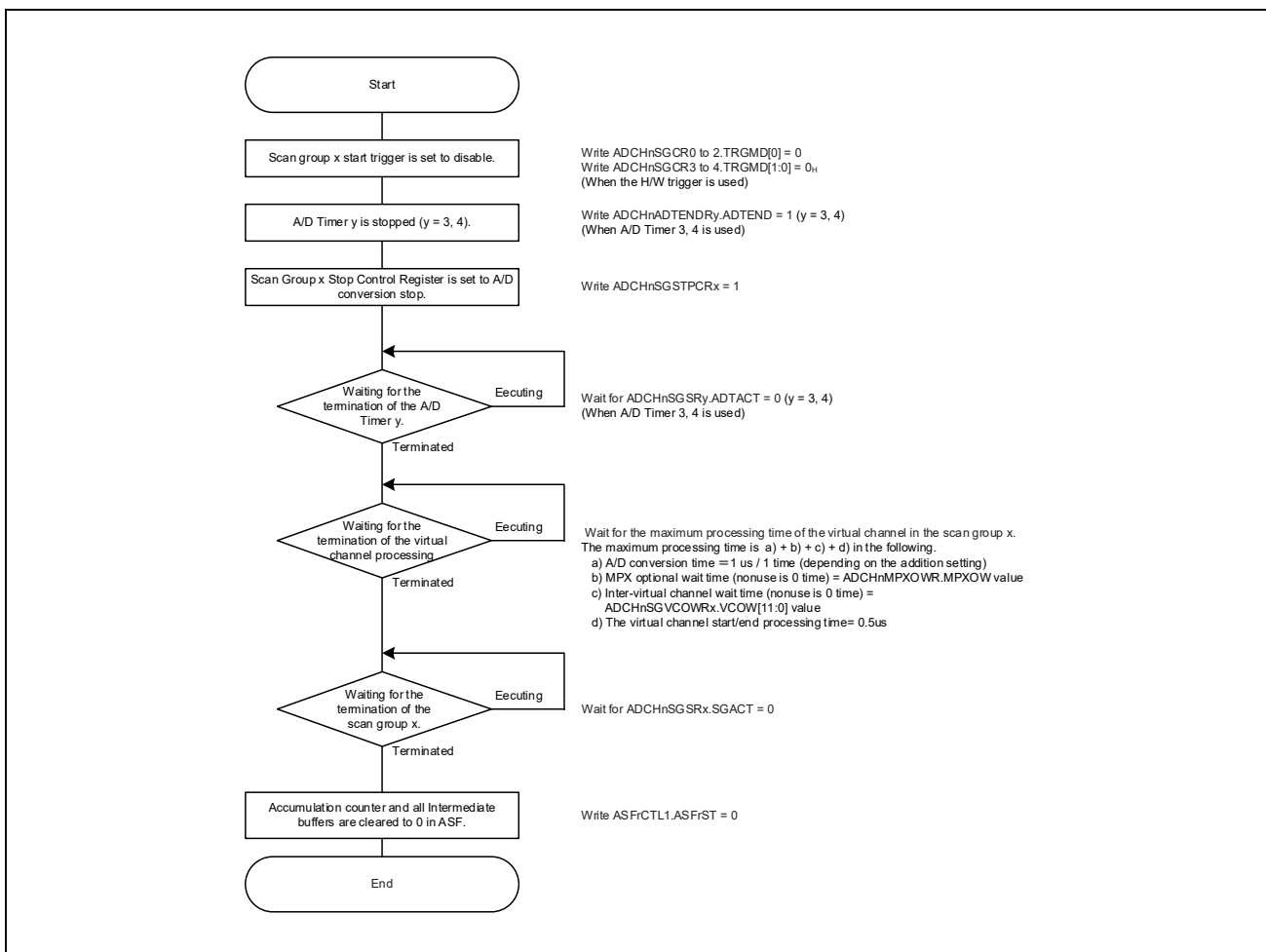


Figure 34.8 Scan Group x Termination Flow 2

When scan group x stop control register is written as 1 in near the end of the A/D conversion, scan group which corresponds changes to IDLE as intended.

But if interrupt-enable-bit (ADCHnSGCRx.ADIE and ADCHnVCRj.ADIE) is set as 1, there is potential which the scan end interrupt request (ADInx) is output.

So if this is dissatisfied, set interrupt-enable-bit (ADCHnSGCRx.ADIE and ADCHnVCRj.ADIE) as 0 before scan group x stop control register is written as 1.

For the detailed description of termination timing, please see **Section 34.4.14.2, Example of Scan Group x Stop.**

34.4.1.4 Limitations of Setting Update

A period to update the setting is limited for the registers shown in the initial setting flow.

To update:

Scan group stop → Register setting update → Scan group startup

Note 1. When a scan group of a stop target is using H/W trigger, invalidate H/W trigger before stopping.

The period for updating differs depending on registers as follows.

Update settings after the specified termination procedure according to the following table:

Table 34.87 Possible Period of Setting Update of Register for Initial Setup

Register	Bits	Update Possible Period
ADCHnADCR1	All bits	1) All scan groups stop
ADCHnADCR2		
ADCHnSFTCR		
ADCHnTDCR		
ADCHnODCR		
ADCHnMPXCURCR		
ADCHnMPXOWR		
ADCHnMPXINTER		
ADCHnDFASENTSGER		
ADCHnADENDP		
ADCHnVMONVDCR1, 2		
ADCHnVCLMINTER1, 2		
ADCHnSGVCOWRx		
ADCHnSGCRx		
ADCHnSGMICYCRx	All bits	
ADCHnSGVCSPx		
ADCHnSGVCEPx		
ADCHnVCR00 to 39		
ADCHnSGCRx	ADTSTARTE, ADSTARTE, TRGMD[1:0] *, ADIE (Clear only)	3) Always updatable
ADCHnSGVCPRx	All bits	
ADCHnVCULLMTBR0 to 6		
ADCHnMPXCMDR	All bits	
ADCHnADTIPR3 to 4		4) Setting target AD timer y stop
ADCHnADTPRR3 to 4		

Note: For TRGMD[1:0], ADTSTARTE and ADSTARTE, see **Section 34.4.1.2, Startup Method** and **Section 34.4.1.3, Terminating Procedure** also.

The updated setting value is applied at the following timing.

ADCHnSGCRx: Immediately

ADCHnSGVCPR: Next time the corresponding scan group x starts up

ADCHnVCULLMTBR0~6: Immediately

ADCHnMPXCMDR: Start time of normal A/D conversion with the MPX or normal A/D conversion with the MPX in addition mode

34.4.2 Scan Mode

There are two scan modes.

34.4.2.1 Multicycle Scan Mode

A/D conversion for virtual channels assigned to the target SGx is repeated for a specified count (1 to 256 times.)

Set the SGCRx.SCANMD bit to 0 to start multicycle scan mode.

Specify the scan count in the SGMCYCR register. The setting range of the values is from 00H to FFH. The setting value should be a value obtained by subtracting 1 from the desired execution count.

The following figure shows an operation example when the scan count is 2.

- Conversion type: Normal A/D conversion mode (CNVCLS[2:0] = 0H)
- Scan group 0
- Virtual channel: ADCHnVCR0 to ADCHnVCR3

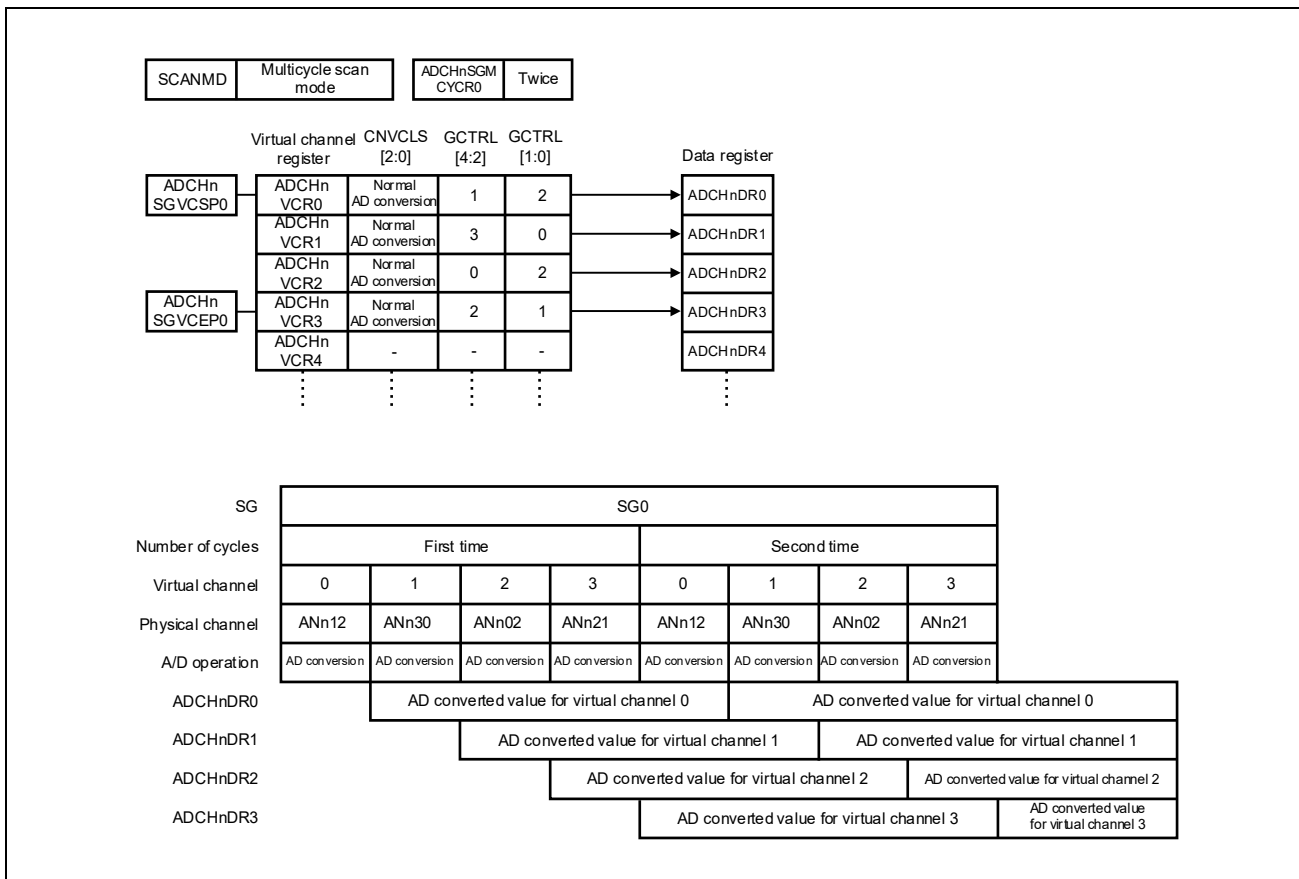


Figure 34.9 Example of Operation in Multicycle Scan Mode

34.4.2.2 Continuous Scan Mode

A/D conversion for virtual channels assigned to the target SGx is repeated with unlimited count.

Set the SGCRx.SCANMD bit to 1 to start continuous scan mode. The setting value of the SGMCYCR register is not effective.

The following figure shows an operation example.

- Conversion type: Normal A/D conversion mode (CNVCLS[2:0] = 0H)
- Scan group 0
- Virtual channel: ADCHnVCR0 to ADCHnVCR3

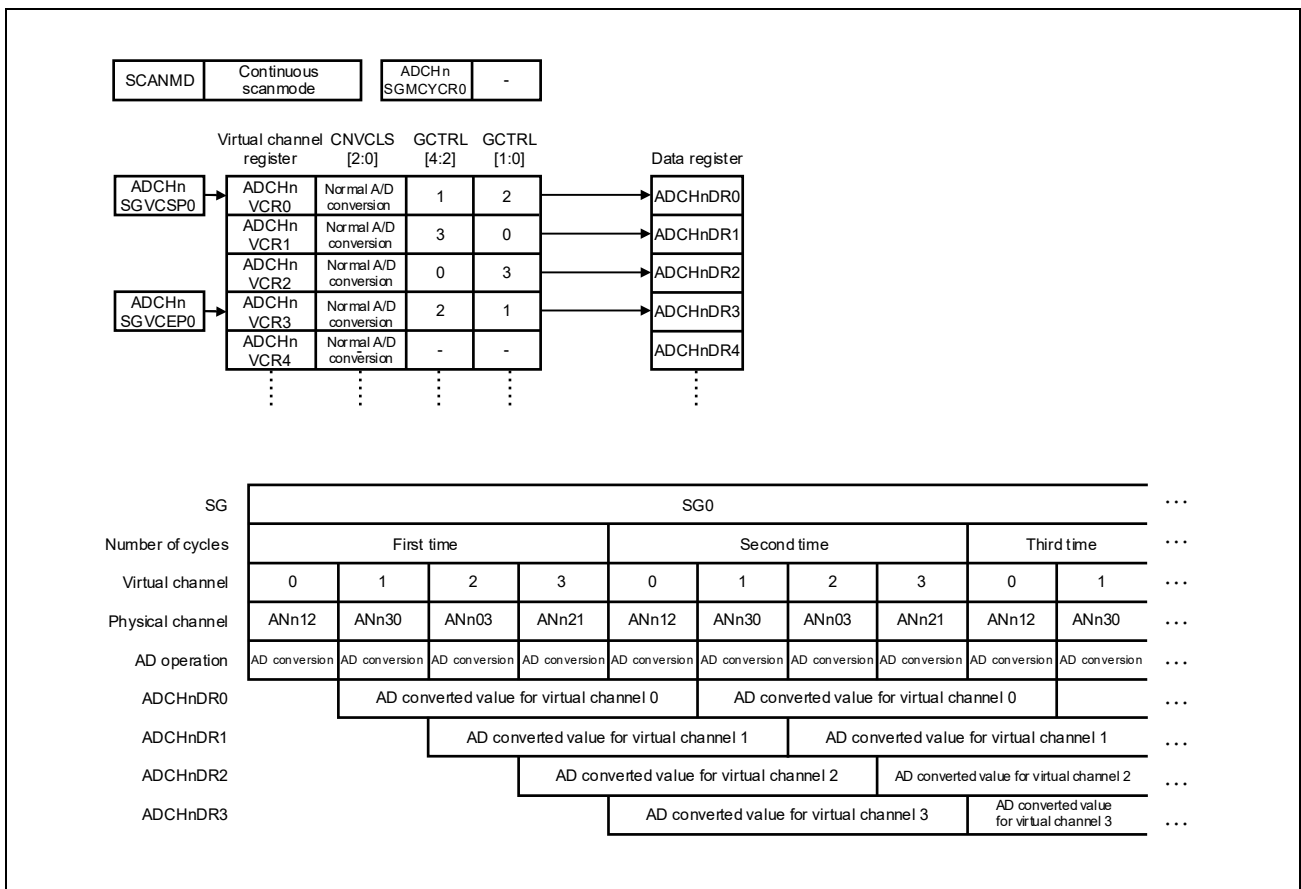


Figure 34.10 Example of Operation in Continuous Scan Mode

34.4.3.1 Synchronous Suspend

The synchronous suspend function interrupts the processing after A/D conversion operation for the virtual channel in progress ends and then starts A/D conversion for the scan group with higher priority.

After A/D conversion for the scan group with higher priority is completed, A/D conversion for the interrupted virtual channel resumes.

The following figure shows an operation example of synchronous suspend.

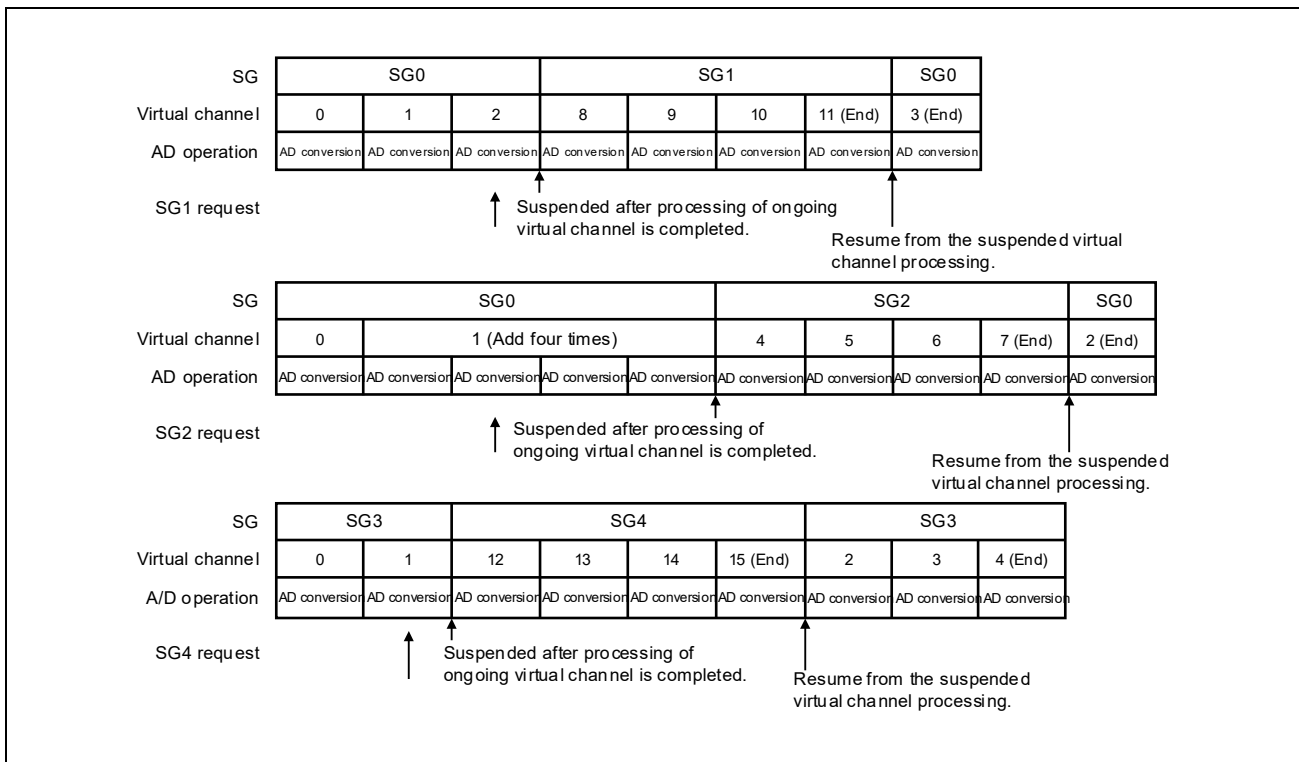


Figure 34.11 Example of Synchronous Suspend and Resume Operation

34.4.3.2 Asynchronous Suspend

The asynchronous suspend function immediately interrupts the processing without waiting until A/D conversion operation for the virtual channel in progress ends and then starts A/D conversion for the scan group with higher priority. After A/D conversion for the scan group with higher priority is completed, A/D conversion for the interrupted virtual channel resumes from the beginning.

The following figure shows an operation example of asynchronous suspend.

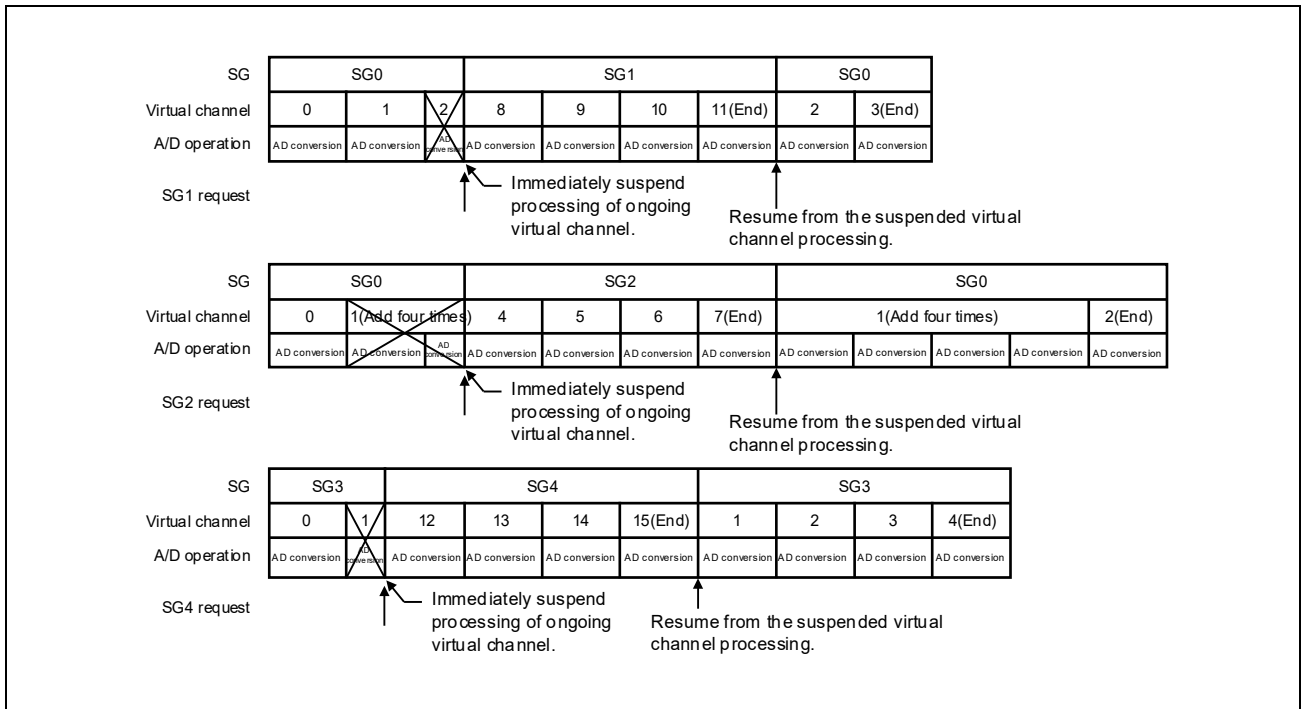


Figure 34.12 Example of Asynchronous Suspend and Resume Operation

34.4.3.3 Synchronous/Asynchronous Mixture Type Suspend

When a scan group with higher priority interrupts scan group 0, the asynchronous suspend method is applied. When a scan group with higher priority interrupts scan groups other than scan group 0, the synchronous suspend method is applied.

The following figure shows an operation example of synchronous/asynchronous-mixed suspend.

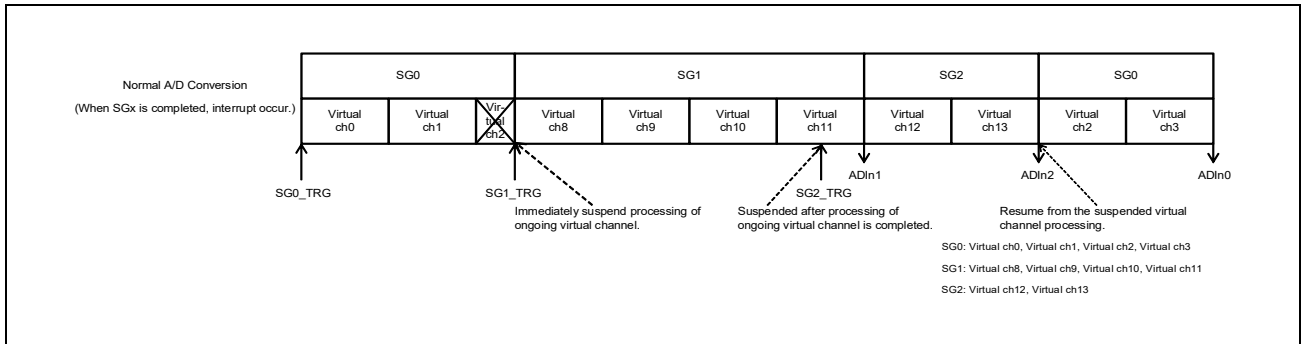


Figure 34.13 Synchronous/Asynchronous-Mixed Suspend

34.4.4 Conversion Class

There are 5 conversion types. The conversion type must be set for each virtual channel, see **Section 34.4.13, Optional Waiting Times Insertion Function of Between Virtual Channels.**

34.4.4.1 Normal A/D Conversion

Normal A/D conversion converts an analog signal in the physical channel to a digital signal as-is.

For details about the operation, see **Section 34.4.2, Scan Mode.**

34.4.4.2 A/D Converter Self-Diagnosis

This conversion method is one of the safety functions.

A/D converter self-diagnosis A/D conversion separates the physical channel and applies the specified voltage to the analog input pin of the A/D converter to perform A/D conversion.

For details about the operation, see **Section 34.4.11.2, A/D Converter Self-Diagnosis.**

34.4.4.3 Example of Addition Mode A/D Conversion Operation

Addition mode A/D conversion converts an analog signal in the physical channel to a digital signal two or four times continuously, and then stores the addition value in the data register. The number of addition times (two or four times) is common among all the virtual channels.

The following figure shows an example of the operation:

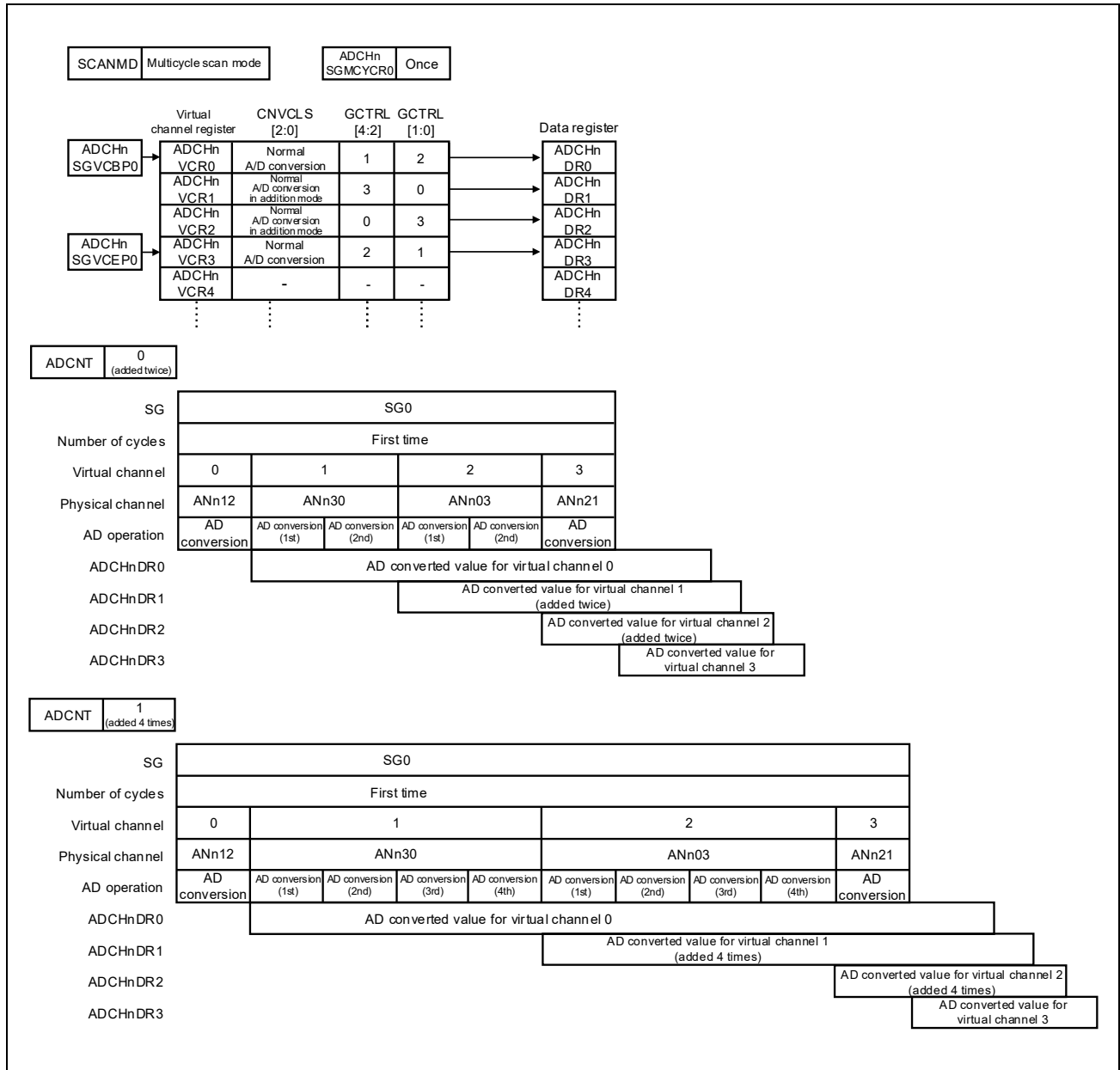


Figure 34.14 Example of Normal A/D Conversion Operation in Addition Mode

For operation when suspend occurs during execution of addition mode A/D conversion, refer to **Section 34.4.3, Suspend Method**.

34.4.4.4 Normal A/D Conversion with the MPX and Normal A/D Conversion with the MPX in Addition Mode

These functions perform A/D conversion by selecting the MPX input channel (MPX value) in the physical channel for which the MPX is connected to the external of the LSI.

There are two methods: normal A/D conversion (without addition) and addition mode A/D conversion.

To transfer the MPX value to the external of the LSI, use functions of the DMAC, IO port, CSIH, and ADMPXn4 to ADMPXn0.

The following registers must be set in advance. See the specifications of each function to set the following registers:

- MPX current control register (ADCHnMPXCURCR)
- MPX command information register (ADCHnMPXCMDR)
- MPX physical channel number of the virtual channel register n (ADCHnVCRj.GCTRL[4:0])
- MPX optional wait register (ADCHnMPXOWR)
- MPX interrupt enable register (ADCHnMPXINTER)

The ADCHn operates as follows:

- (1) Stores the values of ADCHnMPXCURCR, ADCHnMPXCMDR, ADCHnVCRj.GCTRL[4:0] in the MPX current register (ADCHnMPXCURR) when starting the virtual channel and outputs an interrupt signal (ADMPXIn).
- (2) Output the ADCHnMPXCURR.MPXCUR[4:0] value to ADMPXn4 to ADMPXn0 when ADMPXIn outputs.
- (3) Inserts wait states specified in MPXOWR.
(Waits until the MPX value is transferred by the DMAC, IO port, or CSIH and MPX output is settled.)
- (4) Performs A/D conversion for the physical channel connected with the MPX.

Wait states must be set so that the MPX value transfer time and MPX output settling time can be secured.

Wait states can be set within the range from 0 to 10. 1 wait state is 40 ADCLK.

See the IO port specifications to set the mask setting value of the MPX current control register.

This setting value is used for IO port output and is not used for CSIH output.

Set the CSIH command information in the SPI command information register.

This setting value is used for CSIH output and is not used for IO port output.

For operation when suspend occurs during execution of Normal A/D Conversion with the MPX and Normal A/D Conversion with the MPX in Addition Mode, refer to **Section 34.4.3, Suspend Method**.

(1) Using I/O Port Output

The following figure shows a structure and operation example when the DMA and IO port output are used.

- Normal A/D conversion w/MPX
- MPX wait state = 2 wait states
- ADCHnVCR2 to ADCHnVCR5 are used.

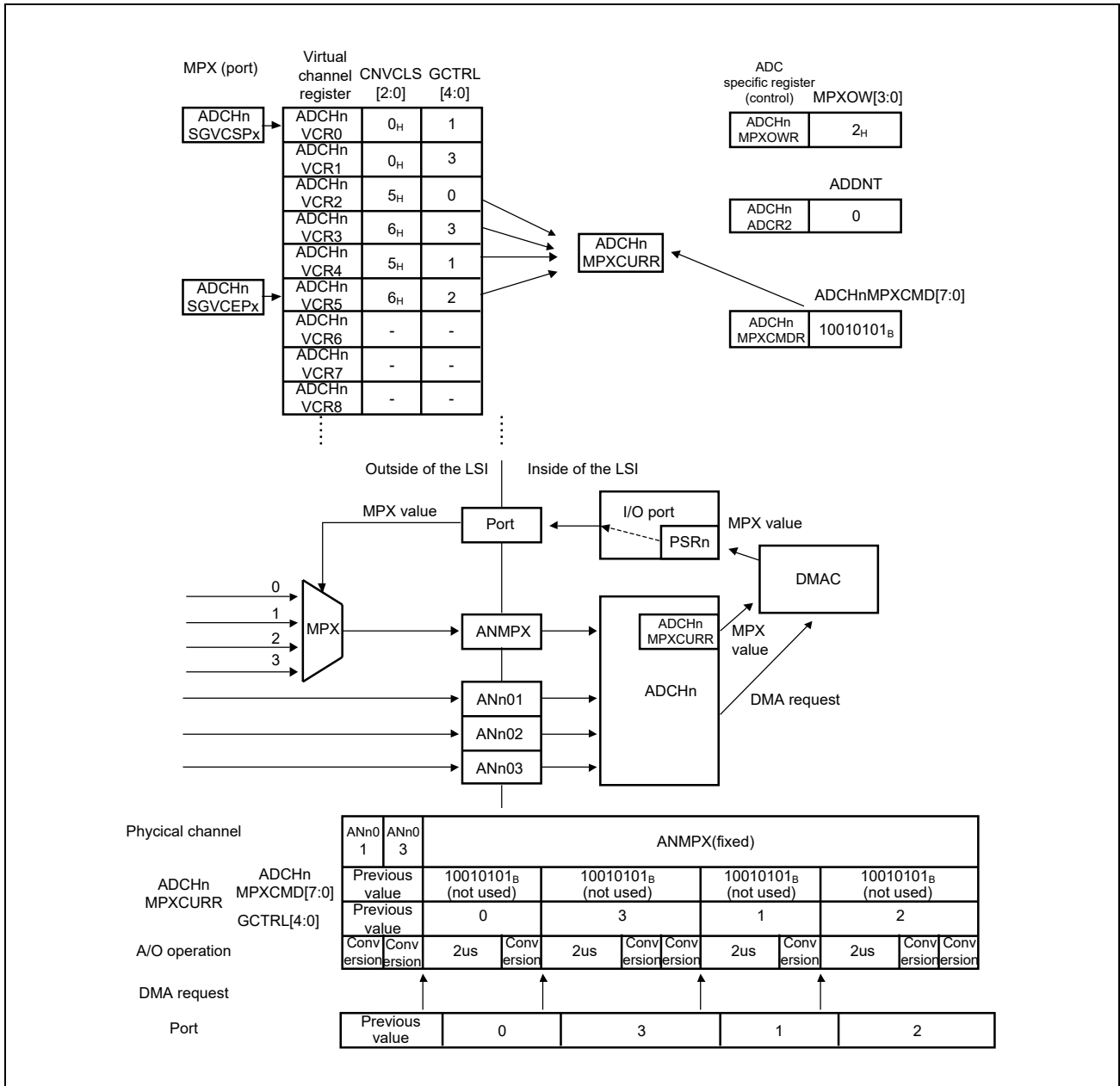


Figure 34.15 Example of Using an External Analog Multiplexer — I/O Port Output

NOTE

ADCH0: ANMPX = AN031; ADCH2: ANMPX = AN243; ADCH1: ANMPX = AN100; ADCH3: ANMPX = AN300

(2) Using CSIH Output

The following figure shows a structure and operation example when the DMA and CSIH output are used.

- Addition mode A/D conversion w/MPX
- MPX wait state = 2 wait states
- ADCHnVCR2 to ADCHnVCR5 are used.

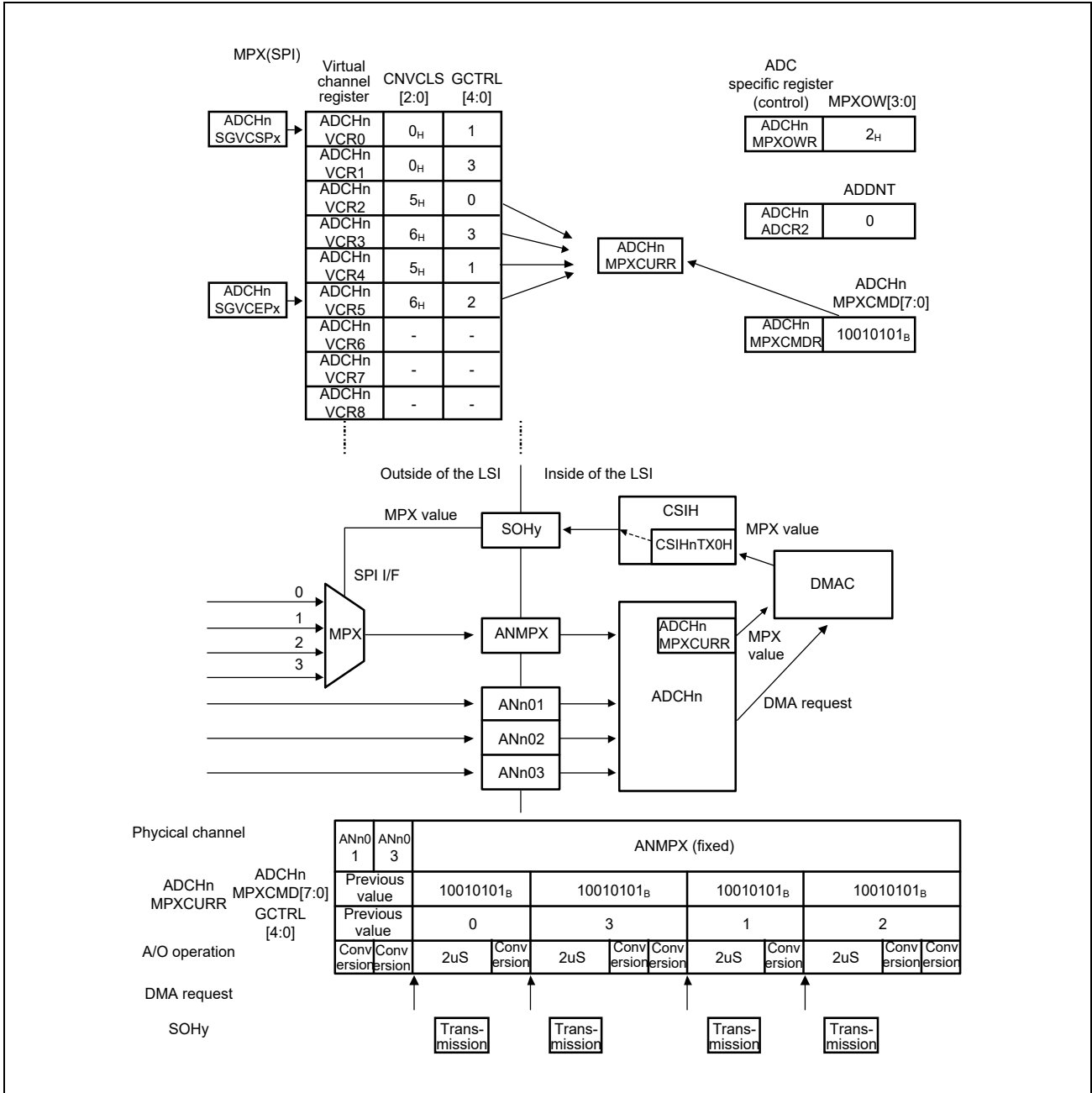


Figure 34.16 Example of Using an External Analog Multiplexer — CSIH Output

NOTE

ADCH0: ANMPX = AN031; ADCH2: ANMPX = AN243; ADCH1: ANMPX = AN100; ADCH3: ANMPX = AN300

34.4.5 Trigger Functions

The ADCH_n has an S/W and H/W trigger.

When the scan group is operating (ADCH_nSGSRx.SGACT = 1), start trigger for the same scan group is ignored.

When the AD timer is operating (ADCH_nSGSRx.ADTACT = 1), start trigger for the same AD timer is ignored.

The following table shows availability of the trigger function in each scan group:

Table 34.88 List of trigger functions for each a scan group

Classification	Function	Register bit name or terminal name	SG0	SG1	SG2	SG3	SG4
S/W Trigger	Scan Group x Starting	ADCH _n SGSTCRx.SGST	✓	✓	✓	✓	✓
	Scan Group x End	ADCH _n SGSTPCRx.SGSTP	✓	✓	✓	✓	✓
	A/D Conversion Synchronous Start	ADCH _n ADSYNSTCR.ADSTART	✓	✓	✓	✓	✓
	A/D Abort	ADCH _n ADHALTR.HALT	✓	✓	✓	✓	✓
	AD Timer Start	ADCH _n ADTSTCRy.ADTST	—	—	—	✓	✓
	AD Timer Synchronous Start	ADCH _n ADSYNSTCR.ADTSTART	—	—	—	✓	✓
	AD Timer end	ADCH _n ADTENDCRx.ADTEND	—	—	—	✓	✓
H/W Trigger	SGx Trigger	SGx_TRG	✓	✓	✓	✓	✓

Note: ✓: Function present
—: Function not present

34.4.5.1 S/W Trigger

A S/W trigger occurs when a bit of a register is set to 1. Each S/W trigger possesses a single function.

(1) Scan Group x Starting Trigger

The A/D conversion of scan group x is started.

This trigger is always valid.

(2) Scan Group x Stop Trigger

The A/D conversion of scan group x is stopped.

This trigger is always valid.

(3) A/D Conversion Synchronous Start Trigger

The A/D conversion of scan group x in each ADCH is started at the same time.

This trigger is valid for each scan group x where an A/D conversion synchronous starting enable bit of scan group x control register (ADCH_nSGCRx.ADSTARTE) of each ADCH is set 1.

(4) A/D Halt Trigger

Operation of all scan groups of the whole A/D conversion operation (including the scan group which is suspended) is halted and the internal state machine is initialized.

It is not possible to halt only a specific scan group x.

When scan group x has been started once again, scan group x cannot be resumed from the virtual channel set at the end of the prior conversion.

Processing is started from the virtual channel indicated by a start pointer.

This trigger is always valid.

(5) A/D Timer Starting Trigger

The Count operation of AD timer y is started.

It is possible to start scan group x at desired periods.

This trigger is always valid.

For details, see **Section 34.4.5.3, A/D Timer Trigger**.

(6) A/D Timer Synchronous Starting Trigger

The Count operation of A/D timer y in each ADCH is started at the same time.

It is possible to start scan group y at the desired periods.

This trigger is valid to scan group x where the A/D timer synchronous starting enable bit of scan group y control register (SGCRy.ADTSTARTE) of each ADCH is set 1.

For details, see **Section 34.4.5.3, A/D Timer Trigger**.

(7) A/D Timer Stop Trigger

The Count operation of AD timer y is stopped.

This trigger is always valid.

For details, see **Section 34.4.5.3, A/D Timer Trigger**.

34.4.5.2 H/W Trigger

A H/W trigger is a rising edge of SGx_TRG.

There are external triggers (ADTRGn) as well as ATU and GTM as a trigger sources of SGx_TRG of each ADCHn, and they are chosen by the PIC. For details, refer to **Section 33, Peripheral Interconnection (PIC)**.

It functions as scan group start/AD timer start.

The function is chosen by a register setting.

(1) H/W Trigger List

The H/W trigger function chosen by each setting is indicated in the following table.

Table 34.89 H/W Trigger Functions List

SGx_TRG					ADCHnSGCRx		SGx_TRG Functions	Notes
4	3	2	1	0	TRGMD[1]	TRGMD[0]		
✓	✓	✓	✓	✓	0	0	Invalid	Scan of SGx is not started.
						1	SGx scan start	
✓	✓	—	—	—	1	0	Invalid	Scan of SGx is not started.
						1	AD timer y start	

Note: ✓: Function present —: Function not present

(2) H/W Trigger Route

The route of H/W trigger input is illustrated using figure.

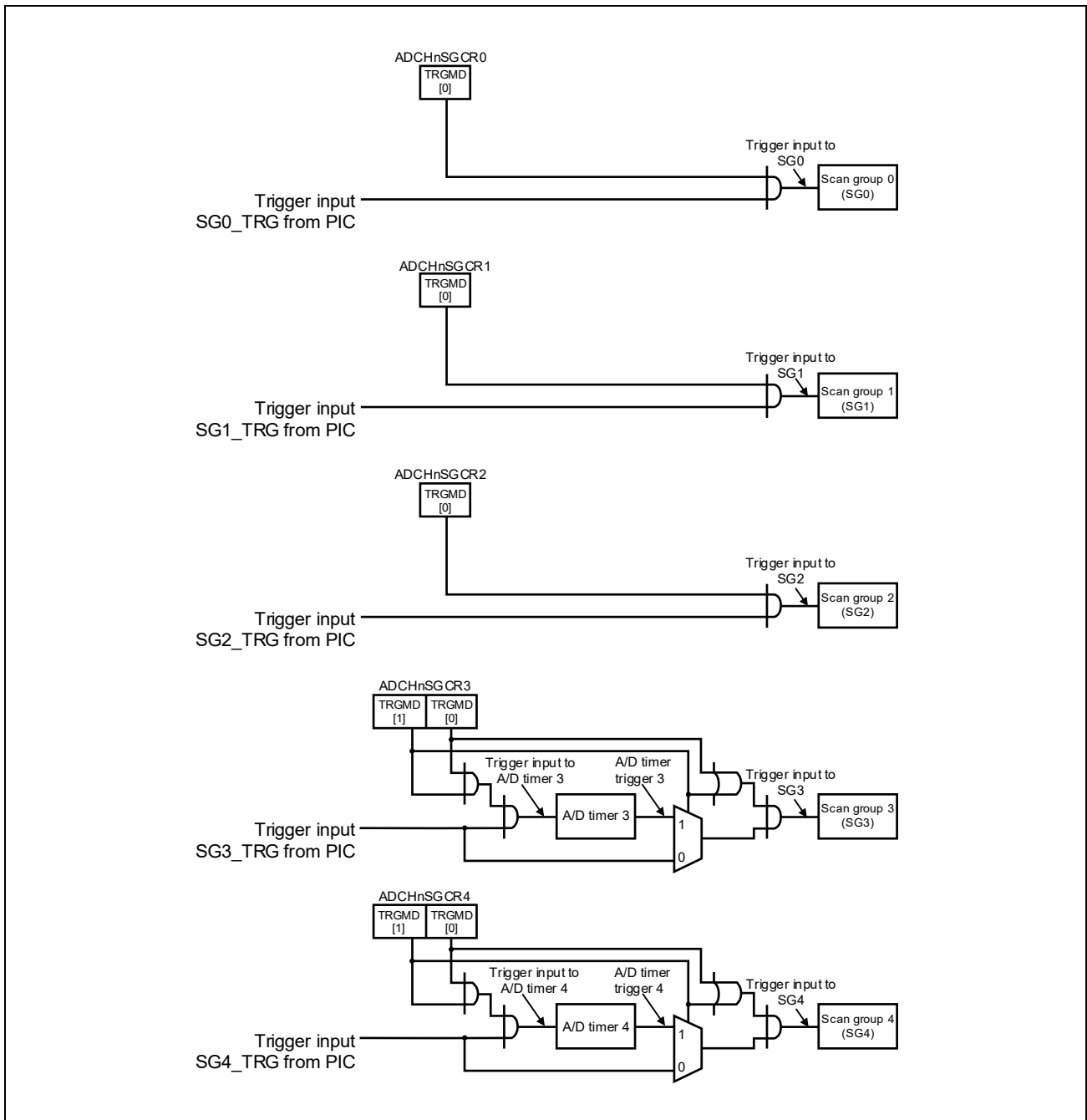


Figure 34.17 Route of Trigger Input

34.4.5.3 A/D Timer Trigger

It is possible to start a scan group y at a constant interval by using an A/D timer trigger.

The A/D timer counts in synchronization with the ADCLK.

[Recommendation flow]

The A/D timer y initial phase register (ADCHnADTIPRy) and A/D timer y cycle register (ADCHnADTPRR) is set according to the initial settings.

The range of these registers is 00_0000_H to 1F_FFFF_H.

Set these register so that the period of the A/D timer trigger does not become shorter than the execution time of scan group y .

The AD timer y is started by an AD timer starting trigger.

The AD timer y is stopped by an AD timer stop trigger or A/D conversion halt (ADCHnADHALTR.HALT = 1).

An operation example is shown in the following.

- (1) When A/D timer y is started, ADCHnADTIPRx is loaded in A/D timer y and the timer counts down.
- (2) When A/D timer y reaches 0, A/D timer trigger y is output in 1 ADCLK period, and ADCHnADTPRRx is loaded in A/D timer y and the timer counts down again.

After this, Step (2) will be repeated until A/D timer y is stopped.

When A/D timer y accepts the A/D timer starting trigger after A/D timer y is stopped, it is started from step (1).

It is possible to begin an AD timer of each ADCHn at the same time by an AD timer synchronous start trigger.

The operation after a synchronous start is same as above.

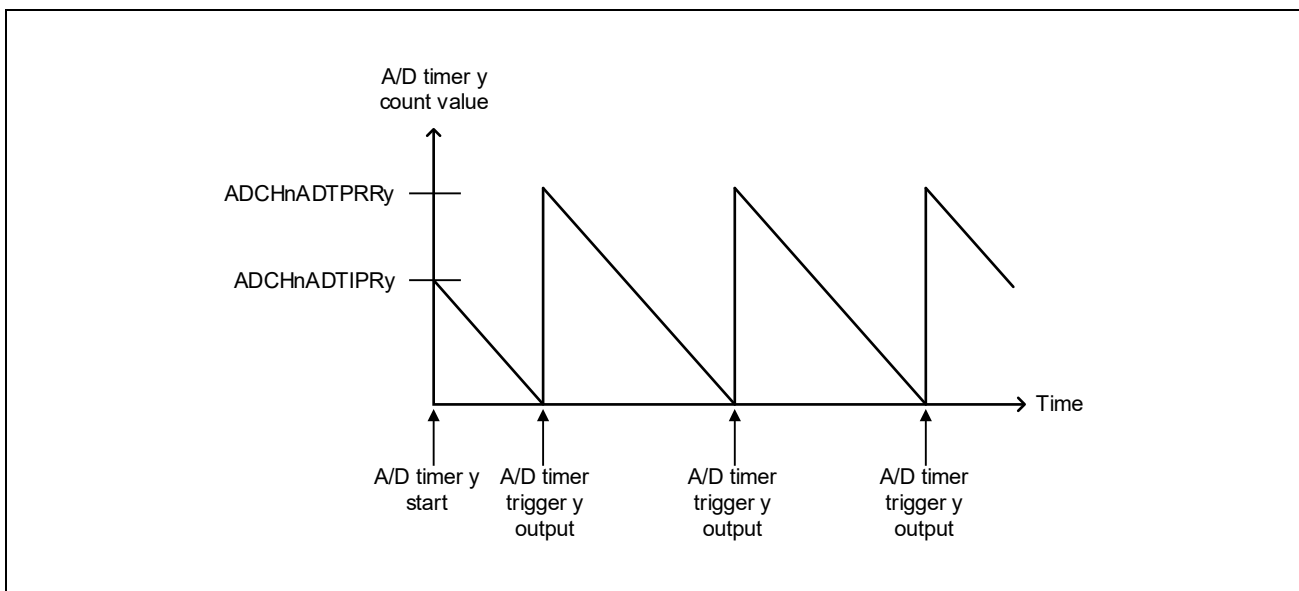


Figure 34.18 Example of A/D Timer Operation

34.4.6 Interrupt Request Functions

There is a scan end interrupt, A/D error interrupt, parity error interrupt, MPX interrupt, and upper/lower limit error interrupt.

The interrupt request is output in 1 register access clock period.

34.4.6.1 Scan End Interrupt Request

When scan group x (*1), or ADCHnVCRj has ended, a scan end interrupt request (ADInx) is generated.

A DMAC can be started by a scan end interrupt.

Output of ADInx is set by the permissions determined by the following bit.

- Scan end interrupt enable bit (ADCHnSGCRx.ADIE) in scan group x
- Virtual channel end interrupt enable bit (ADCHnVCRj.ADIE) in virtual channel register n

Table 34.90 ADInx Generation Permission / No Permission Settling List

ADCHnSGCRx ADIE	ADCHnVCRj ADIE	ADInx Output	Generation Condition
0	0	Not permitted	—
	1	Permitted	Virtual channel j is ended
1	0	Permitted	Scan group x is ended (*1)
	1	Permitted	Scan group x is ended (*1), or virtual channel j is ended

Note 1. The meaning of 'the end the time of scan group x' is shown in the following.

- When an A/D conversion of the virtual channel indicated by 'the Scan Group x End Virtual Channel Pointer (ADCHnSGVCEPx)' is completed.
- When an A/D conversion of the virtual channel indicated by 'VCEP of the Scan Group x Virtual Channel Pointer Register (ADCHnSGVCPRx)' is completed.

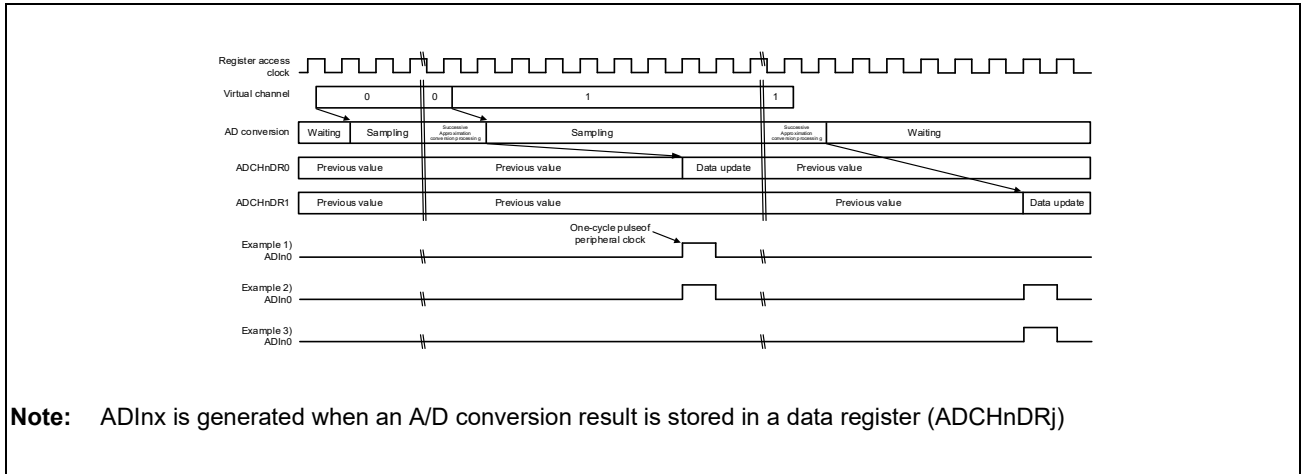


Figure 34.19 Scan Conversion End Interrupt Occurrence Timing

34.4.6.2 A/D Error Interrupt Request

When the following A/D error occurs, an A/D error interrupt (ADEn) is generated.

The generation timing of the A/D error interrupt is same as the generation timing of the scan end interrupt.

Clear the error status in the interrupt handler, after the interrupt is generated.

A/D error interrupt is generated when the OR-condition of the interrupt-enable-bit of the safety-control-register (ADCHnSFTCR) is valid.

Table 34.91 ADEn Generation Permitted List

ADCHnSFTCR		ADEn	Generation Condition
OWEIE	IDEIE		
0	0	Not permitted	—
0	1	Permitted	ID error is occurred
1	0	Permitted	Overwrite error is occurred
1	1	Permitted	Overwrite error is occurred, ID error is occurred

Each error detection is always valid.

Error status information is stored in the error register for each error.

When overwrite error were detected, the error flag bit is set 1, and the virtual channel number for the error generated is captured in the capture bit.

When ID error is detected, the error bit of the data supplementary information register j (ADCHnDIRj) which corresponds with the virtual channel (ADCHnVCRj) where an ID error is detected is set to 0. When an ID error is not detected, the error bit is set to 1.

Error status information except for that in the ADCHnDIRj.IDEF is maintained until it's cleared by the clear bit of each error flag of an error-clear-register. When the Read-and-Clear-Enable (RDCLRE) is set 1, ADCHnDIRj.IDEF maintains its status until the flag bit of the relevant error is read.

When the same error occurs once again without being cleared, the error is ignored.

If the interrupt-enable-bit is set to a valid state, when an error of following type occurs, the AD error interrupt is output.

Table 34.92 ADEn Error Factor Register List

Error Name	Error Register		Error Clear Register
	Flag Bit	Capture Bit	Clear Bit
Overwrite error	ADCHnQWER.OWE	ADCHnOWER.OWECAP[5:0]	ADCHnECR.OWEC
ID error	ADCHnDIRj.IDEF	no function.	no function.*1

Note 1. When the Read-and-Clear-Enable (RDCLRE) is set 1, and the data supplementary information register j (ADCHnDIRj) is read, the Error Clear Register clear bit set to 0.

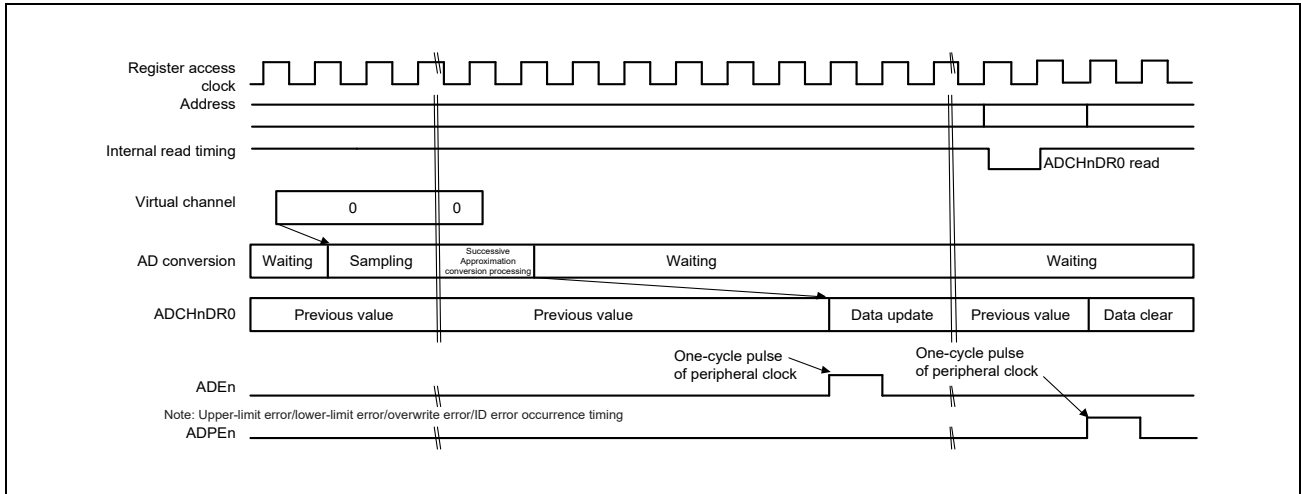


Figure 34.20 Timing of A/D Error Interrupt and Parity Error Interrupt Generation

34.4.6.3 Overwrite Error

This error indicates that the new A/D conversion result is overwritten to the data register before the data register (ADCHnDRj) is read.

When the write flag (ADCHnDIRj.WFLG) of the data accessory information register is 1 and the new A/D conversion result is stored in the corresponding data register j (ADCHnDRj), the error is detected.

34.4.6.4 ID Error

This error indicates that the physical channel specified in the virtual channel register (ADCHnVCRj) is inconsistent with the actually-converted physical channel, that is, an operational failure occurred in the internal analog switch of the IO buffer/RRAMP.

ID Error detection result is stored in the ID error bit of Data Supplementary Information Register j (ADCHnDIRj) that is corresponded to detect the Virtual Channel Register n (ADCHnVCRj) that detected ID error.

The signal level of feedback from IO buffer/RRAMP is checked.

The setting of ADCHnVCRj.GCTRL[5:0] is compared with IO buffer/RRAMP number that is activated, then when the result is not equal, the error is detected. Because IO buffer/RRAMP are controlled by one-hot, if feedback signal is not one-hot signal, the error is detected.

The ID error is detected when any of the following condition is met.

- (1) Mismatched between selecting physical channel number and activated I/O buffer number.
- (2) Mismatched between physical channel number that is assigned to the MPX and the I/O buffer number that is activated.*1
- (3) Mismatched between activated I/O buffer number and activated RRAMP number.*2
- (4) More than two of I/O buffer are activated (One-hot error) when A/D conversion is executed.
- (5) More than two of RRAMP are activated (One-hot error) when A/D conversion is executed.
- (6) More than one of RRAMP are activated in the period that all RRAMP output should off when A/D conversion is executed.

Note 1. Because there are no feedback from MPX, the check cannot be done comparing with MPXvalue.

Note 2. The RRAMP number of physical channel number change by ADCHn. For a detail description, please see **Section 34.1.6, External Input/Output Signals.**

The detection condition changes by the detection factor.

Table 34.93 ID Error Detect List

Class of Conversion	(1)	(2)	(3)	(4)	(5)	(6)
Normal A/D conversion	✓	—	✓	✓	✓	✓
Normal A/D conversion in addition mode	✓	—	✓	✓	✓	✓
Normal A/D conversion with MPX	—	✓	✓	✓	✓	✓
Normal A/D conversion with MPX in addition mode	—	✓	✓	✓	✓	✓
A/D convertor self-diagnosis	—	—	—	—	—	—

34.4.6.5 Parity Error Interrupt Request

The ADCH_n can issue a Parity Error notification (ADPE_n) to the ECM.

The Parity Error is detected in second cycle when ADCH_mDR_j or ADCH_mDIR_j is read.

The Parity Error notification is generated by parity error when ADCH_nSFTCR.PEIE is set to 1.

The Parity Error can be checked by reading ADCH_nDIR_j.PRTY.

The parity bit is even parity for ADCH_nDR_j and ADCH_nDIR_j.IDEF.

The error status information is stored in PER register. When parity error is detected, PER.PE is set to 1 and PER.PRCAP[5:0] is set the virtual channel number that generate parity error.

The error status information is cleared by writing to 1 to ADCH_nECR.PEC.

When PER.PE is set, next error is ignored, but when ADCB_nSFTCR.PEIE is set to 1, the parity error is output to ECM if next error is generated.

34.4.6.6 MPX Interrupt Request

The ADCH_n can issue an MPX interrupt request (ADMPXIn) to the INTC.

If the MPX-interrupt-enable-bit is set to a valid (ADCH_nMPXINTER.ADMPXIE = 1) level, ADMPXIn is generated when a virtual channel for which normal-A/D-conversion-with-MPX (ADCH_nVCRj.CNVCLS[2:0] = 5_H) or normal-A/D-conversion-with-MPX-in-addition-mode (ADCH_nVCRj.CNVCLS[2:0] = 6_H) is started.

In both cases such as MPX wait and wait between virtual channels, ADMPXIn is output 4 register access clocks after the conversion is started.

The DMAC can be activated when an ADMPXIn occurs.

If the MPX-interrupt-enable-bit was set to an invalid state (ADCH_nMPXINTER.ADMPXIE = 0), ADMPXIn is not output.

However, the ADMPX_{n0} to ADMPX_{n4} of the output ports is updated irrespective of the value of ADCH_nMPXINTER.ADMPXIE.

ADMPX_{n0} to ADMPX_{n4} is output just as ADCH_nMPXCURR.MPXCUR[4:0] is transferred, so the update timing is the same as ADCH_nMPXCURR.MPXCUR[4:0].

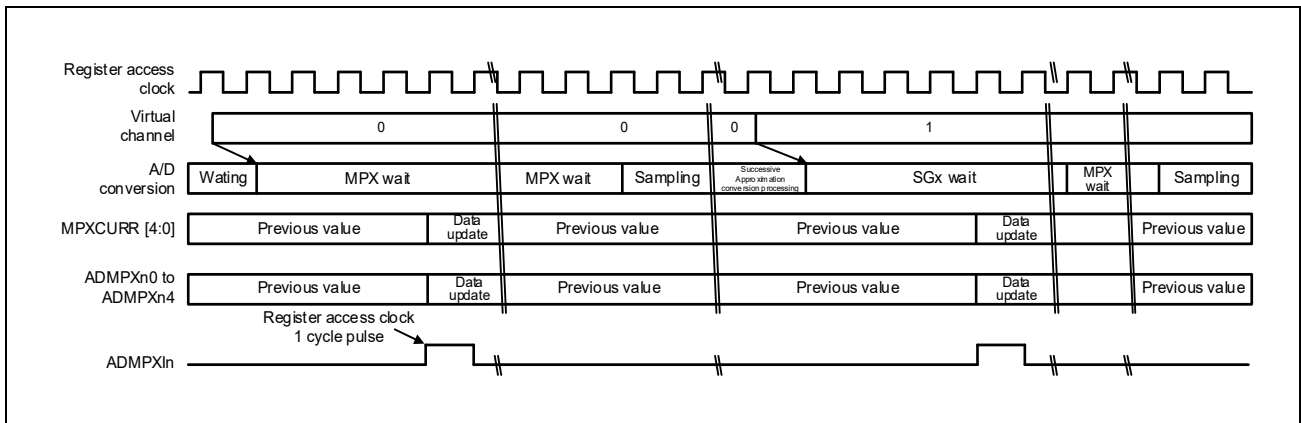


Figure 34.21 Timing of an MPX Interrupt Occurrence

34.4.6.7 Virtual Channel Upper/Lower Limit Error Interrupt

ADCH_n detects that the A/D conversion result is above the upper limit or below the lower limit for each virtual channel, and generates an interrupt request (INT_UL_n).

(1) Setting Registers

INT_UL_n is to set the upper/lower-limit-error-interrupt-enable-bit (ADCH_nVCLMINTER1.ADUL_jIE, ADCH_nVCLMINTER2.ADUL_jIE) and the virtual-channel-upper-limit-excess-notice-enable-bit (ADCH_nVCR_j.VCULME) or virtual-channel-lower-limit-excess-notice-enable-bit (ADCH_nVCR_j.VCLLME) as 1, and this function is set to valid or invalid in each virtual channel.

When it is set as valid, the result which compares the A/D conversion result with the virtual-channel-upper/lower-limit-table-register (ADCH_nVCULLMTBR0 to ADCH_nVCULLMTBR6) is output to INT_UL_n.

When ADUL_nIE is set as “0” or both of VCULME and VCLLME are set as “0”, INT_UL_n is not output (the output is fixed low.)

Table 34.94 INT_UL_n, v_{ce}nd[j], v_{cu}lmo and v_{cl}lmo Output Setting

ADUL _j IE	VCULME	VCLLME	INT_UL _n	v _{cu} lmo	v _{cl} lmo	v _{ce} nd[j]
0	0	0	—	—	—	—
		1	—	—	✓	✓
	1	0	—	✓	—	✓
		1	—	✓	✓	✓
1	0	0	—	—	—	—
		1	✓	—	✓	✓
	1	0	✓	✓	—	✓
		1	✓	✓	✓	✓

Note: For details of v_{ce}nd[j], v_{cu}lmo and v_{cl}lmo, first see **Figure 34.27** and see **Section 34.4.9, Virtual Channel Upper/Lower Limit Excess Notice**.

✓: Valid —: Invalid

The output condition of the upper/lower limit error interrupt signal (INT_UL_n) is as follows.

Precondition: When the A/D conversion value is greater than the upper limit value or less than the lower limit value.

- (1) INT_UL_n is not output irrespective of the setting of VCULME and VCLLME if ADUL_jIE=0.
- (2) INT_UL_n is not output irrespective of the setting of ADUL_jIE if VCULME=0 and VCLLME=0.
- (3), (4), (5) When ADUL_jIE = 1 and both VCULLME, VCLLME are set to “1” or VCULLME = 1 or VCLLME = 1, INT_UL_n is output at same timing as scan end interrupt (ADIn).

When the class of conversion is “normal A/D conversion in addition mode”, INT_UL_n is output only after the end of the addition number of times of the designated A/D virtual channel.

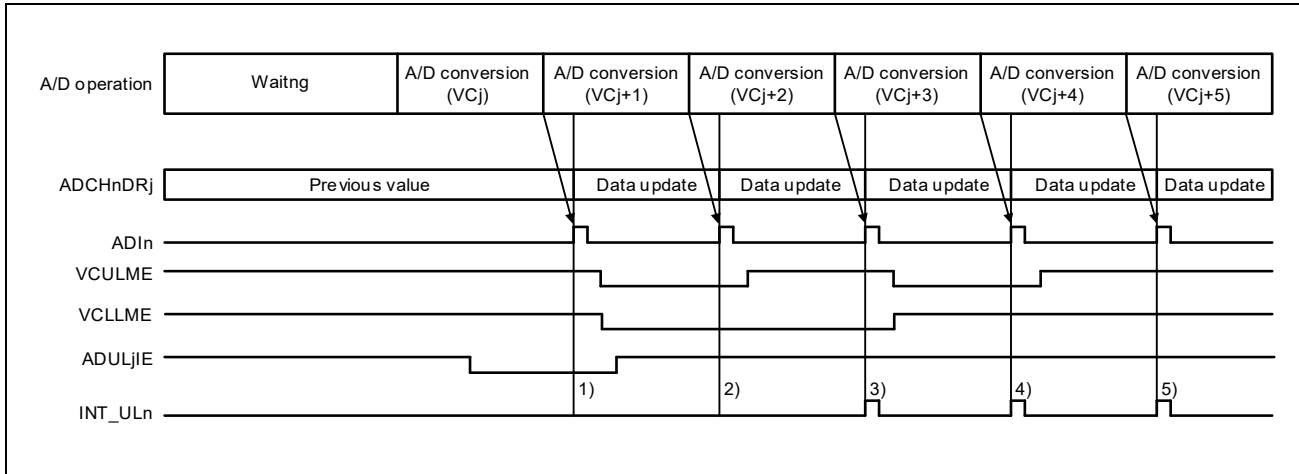


Figure 34.22 Output Condition of Upper / Lower Limit Error Interrupt

Table 34.95 INT_ULn Output Permitted List for Each Conversion Class

ADCHnVCRj. CONVCLS[2:0]	Conversion Class	Valid / Invalid	Notes
0 _H	Normal A/D conversion	✓	
1 _H	Setting prohibited	—	
2 _H	Setting prohibited	—	
3 _H	A/D converter self-diagnosis	✓	
4 _H	Normal A/D conversion in addition mode	✓	It is asserted only at the end of the designated addition counts.
5 _H	Normal A/D conversion with the MPX	✓	
6 _H	Normal A/D conversion with the MPX in addition mode	✓	It is asserted only at the end of the designated addition counts.
7 _H	Setting prohibited	—	

Note: ✓: Valid —: Invalid

(2) Table Registers

See **Section 34.4.9.2, Table Registers**.

(3) Interface Signals

When this function is set as valid, every time A/D conversion in a virtual channel ends, a table register is compared with an A/D conversion result, and INT_ULn is output.

INT_ULn is a common signal for all virtual channels.

INT_ULn is one-shot-pulse signal output by a register access clock.

It's possible to determine the detected number of the virtual channel by reading the virtual-channel-upper/lower-limit-excess-status-register 1 (ADCHnVCLMSR1) and virtual-channel-upper/lower-limit-excess-status-register 2 (ADCHnVCLMSR2).

(4) Operating Timing

INT_ULn signals are synchronized with the register access clock.

Asserting-timing of this signal is the same as the assert-timing of a scan end interrupt signal (ADIn).

The operating timing chart is indicated below.

- 1st comparison period: A/D conversion result = 7FF8_H, INT_ULn is generated by upper limit value excess and flag is set.
- 2nd comparison period: A/D conversion result = 4FF8_H, INT_ULn is generated by lower limit value excess and flag is set.

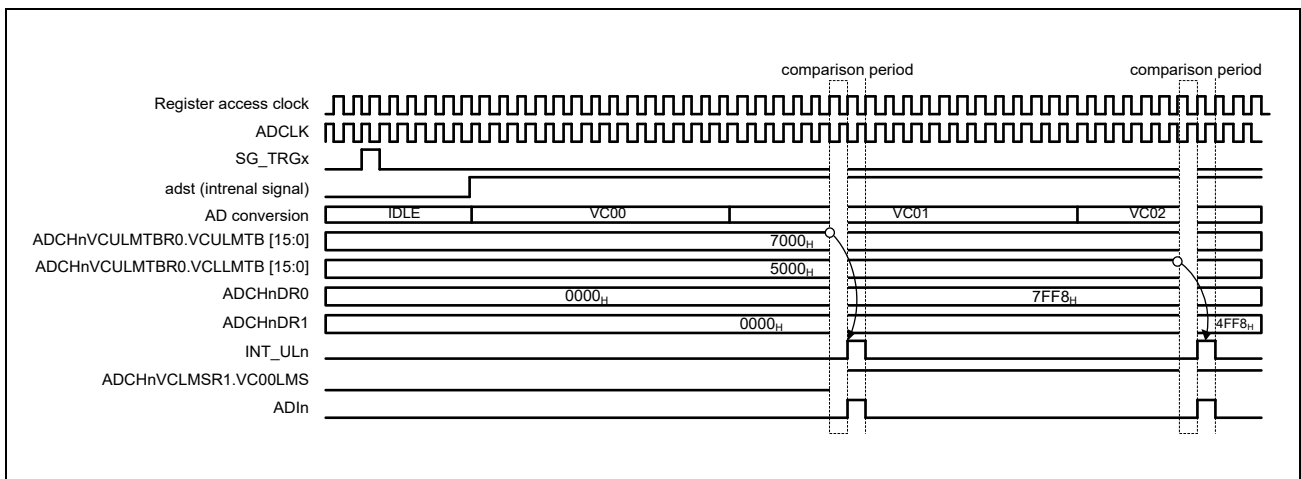


Figure 34.23 Timing Chart of Upper / Lower Limit Error Interrupt

34.4.7 DFE/GTM/ASF Entry Function

The ADCH_n can issue a request for entry to the DFE or GTM and a request for entry to the ASF for DFE, GTM, and ASF according to the settings of DFENT in ADCH_nVCR_j and DFENTSG_xE and ASENTSG_xE in ADCH_nDFASENTSGER. At the same time, ADCH_n outputs the TAG information and A/D conversion data that are set in DFTAG in ADCH_nVCR_j. For the format of the output A/D conversion data, see **Section 34.3.3.2, ADCH_nDR_j — Data Register j**.

For allocation of each ADCH_n and each channel of DFE, refer to **Section 37, Digital Filter Engine (DFE)**.

For allocation of each ADCH_n and each channel of GTM, refer to **Section 31, Generic Timer Module (GTM)**.

For allocation of ADCH₀ and each channel of ASF, refer to **Section 34.8, ADC Summation Function (ASF)**.

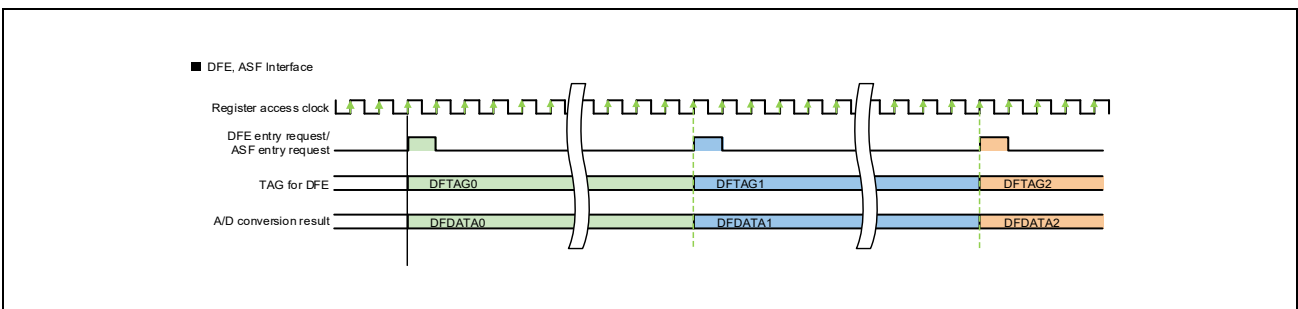


Figure 34.24 DFE/ASF Entry Timing

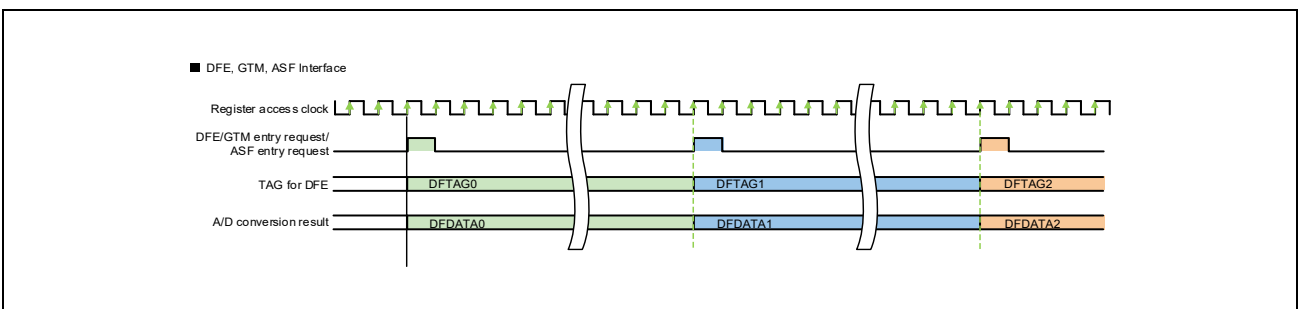


Figure 34.25 DFE/GTM/ASF Entry Timing

34.4.8 Monitoring Function Using the A/D Conversion Monitor Pin

ADENDn can be used to monitor the processing timing of the virtual channel specified by ADCHnADENDP. For details of pin settings, see **Section 2, Pin Function**.

Figure 34.26 shows the A/D conversion monitor timing.

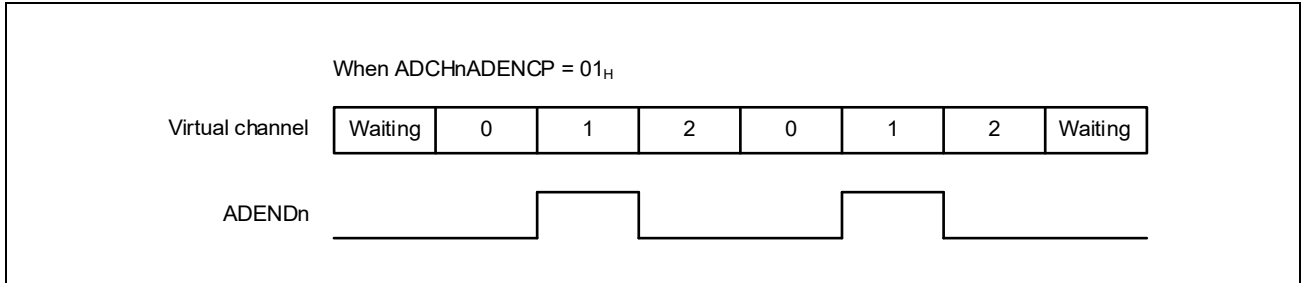


Figure 34.26 A/D Conversion Monitor Timing

CAUTION

If the high-level voltage is output from ADENDn in a lower-priority scan group and a higher-priority scan group suspends (asynchronous suspend) the processing of the lower-priority scan group, the low-level voltage is output from ADENDn. Since the suspended virtual channel processing for the lower-priority scan group resumes after that, the high-level voltage is output again from ADENDn.

34.4.9 Virtual Channel Upper/Lower Limit Excess Notice to ADC VMON Secondary Error Generator (AVSEG), or to ADC Boundary Flag Generator (ABFG)

The ADCH_n can detect that the A/D conversion result is greater than the upper limit value or less than the lower limit value for each virtual channel, and it is possible to notify the ADC VMON secondary error generator (AVSEG), or ADC boundary flag generator (ABFG).

34.4.9.1 Setting Registers

It is possible to set this function as valid or invalid in each virtual channel if the virtual-channel-upper-limit-excess-notice-enable-bit (ADCH_nVCR_j.VCULME) or virtual-channel-lower-limit-excess-notice-enable-bit (ADCH_nVCR_j.VCLLME) is set.

When it is set as valid, the A/D conversion result is compared with the virtual-channel-upper/lower-limit-table-register (ADCH_nVCULLMTBR0 to ADCH_nVCULLMTBR6) and is output to v_{ce}nd[*j*] (A/D conversion end notice), v_{cul}mo (virtual channel upper limit excess notice) and v_{cll}mo (virtual channel lower limit excess notice).

When both of VCULME and VCLLME are set to “0”, v_{cul}mo, v_{cll}mo, and v_{ce}nd[*j*] are not output (output is fixed low).

For details of v_{cul}mo, v_{cll}mo, and v_{ce}nd[*j*], see **Figure 34.27**.

Table 34.96 v_{ce}nd[*j*], v_{cul}mo and v_{cll}mo Output Setting

VCULME	VCLLME	Virtual Channel Upper Limit Excess Notice v _{cul} mo	Virtual Channel Lower Limit Excess Notice v _{cll} mo	A/D Conversion End Notice v _{ce} nd[<i>j</i>]
0	0	—	—	—
	1	—	✓	✓
1	0	✓	—	✓
	1	✓	✓	✓

Note: ✓: Valid —: Invalid

Table 34.97 v_{ce}nd[*j*], v_{cul}mo and v_{cll}mo Notice Function Valid/ Invalid List for Each Conversion Class

ADCH _n VCR _j . CONVCLS[2:0]	Conversion Class	Valid / Invalid	Notes
0 _H	Normal A/D conversion	✓	
1 _H	Setting prohibited	—	
2 _H	Setting prohibited	—	
3 _H	A/D converter self-diagnosis	✓	
4 _H	Normal A/D conversion in addition mode	✓	It is asserted only at the end of the designated addition counts.
5 _H	Normal A/D conversion with the MPX	✓	
6 _H	Normal A/D conversion with the MPX in addition mode	✓	It is asserted only at the end of the designated addition counts.
7 _H	Setting prohibited	—	

Note: ✓: Valid —: Invalid

34.4.9.2 Table Registers

The virtual-channel-upper/lower-limit-table-register (ADCHnVCULLMTBR0 to ADCHnVCULLMTBR6) are specified by the VCULLMTBS [2:0] bits of the virtual channel register (ADCHnVCRj).

The upper limit value table and the lower limit table are sets.

Therefore it isn't possible to choose the lower limit table for the register number different from the upper limit value table in one virtual channel.

Set the upper limit value to the upper-side 16 bits of the chosen ADCHnVCULLMTBR and set the lower limit value to the lower-side 16 bits of the chosen ADCHnVCULLMTBR.

Write using a 32 bit width. This register cannot be written using a 16 bit width or 8 bit width.

ADCHnVCULLMTBR0 to ADCHnVCULLMTBR6 can be rewritten at any time irrespective of the timing before scan group start and after scan group start.

Table 34.98 Correlation between ADCHnVCRj.VCULLMTBS[2:0] and ADCHnVCULLMTBR0 to ADCHnVCULLMTBR6

ADCHnVCRj.VCULLMTBS[2:0]	Register Name	Upper Limit Side Condition	Lower Limit Side Condition
0 _H	ADCHnVCULLMTBR0	A/D conversion value > VCULLMTB[15:0]	A/D conversion value < VCULLMTB[15:0]
1 _H	ADCHnVCULLMTBR1		
2 _H	ADCHnVCULLMTBR2		
3 _H	ADCHnVCULLMTBR3		
4 _H	ADCHnVCULLMTBR4		
5 _H	ADCHnVCULLMTBR5		
6 _H	ADCHnVCULLMTBR6		
7 _H	Setting prohibited		

34.4.9.3 Interface Signals

When this function is used, every time an A/D conversion in a virtual channel has ended, a table register is compared with an A/D conversion result, and `vcend[j]` (A/D conversion end notice), `vculmo` (virtual channel upper limit excess notice) and `vcllmo` (virtual channel lower limit excess notice) are output.

`vcend[j]`, `vculmo` and `vcllmo` are one-shot-pulse signals output by the register access clock.

These signals are output when the A/D conversion ends.

`vcend[j]` is output irrespective of the result of the comparison.

Only when the upper limit value excess is detected, `vculmo` is output, and only when the lower limit value excess is detected, `vcllmo` is output.

`vculmo` and `vcllmo` are common signals for all virtual channels.

It's possible to determine the detected number of the virtual channel by using a pair of `vcend[j]` and `vculmo`, or using a pair of `vcend[j]` and `vcllmo`.

For details of `vculmo`, `vcllmo`, and `vcend[j]`, see **Figure 34.27**.

Table 34.99 `vcend[j]`, `vculmo` and `vcllmo` Output Timing

Signal Name	Means	Operation	Synchronizing Clock
<code>vcend[j]</code>	Virtual channel A/D conversion end notice	It is output at the A/D conversion end.	register access clock
<code>vculmo</code>	Virtual channel upper limit excess notice	When it occurs, it is output at the time of A/D conversion end.	
<code>vcllmo</code>	Virtual channel lower limit excess notice	When it occurs, it is output at the time of A/D conversion end.	

34.4.9.4 Operating Timing

The operation timing chart is indicated below.

vcend[j], vculmo and vcllmo are the signals which are with the register access clock.

Assert-timing of these signals is same as the assert-timing of the scan end interrupt signal (ADIn).

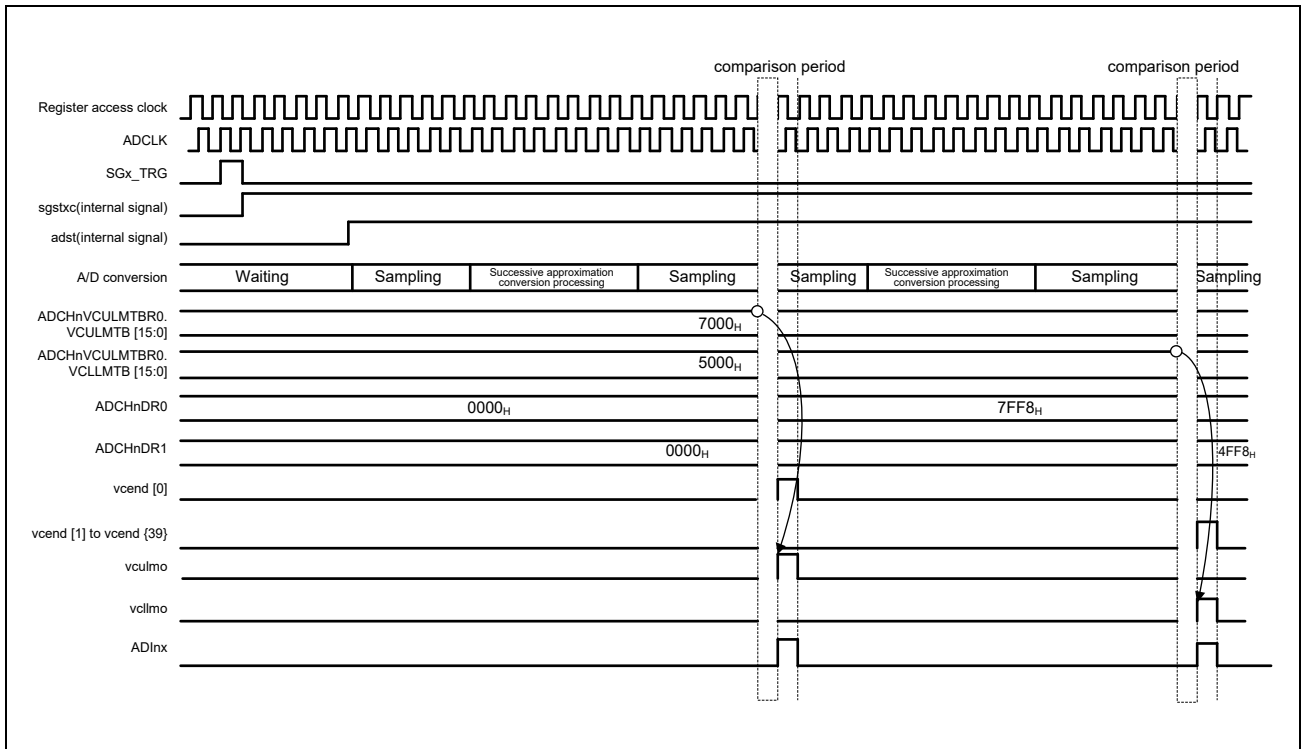


Figure 34.27 Output Timing Chart of Virtual Channel Upper / Lower Limit Excess Notice

The following diagram shows the timing chart when ADCHnDRj storage and ADCHnVCLLMTBR rewrite compete against each other.

Comparison is performed 1 register access clock period before ADCHnDRj storage. Therefore, if ADCHnDRj storage and ADCHnVCULLMTBR rewrite are performed simultaneously, the previous value of ADCHnVCULLMTBR is used for comparison.

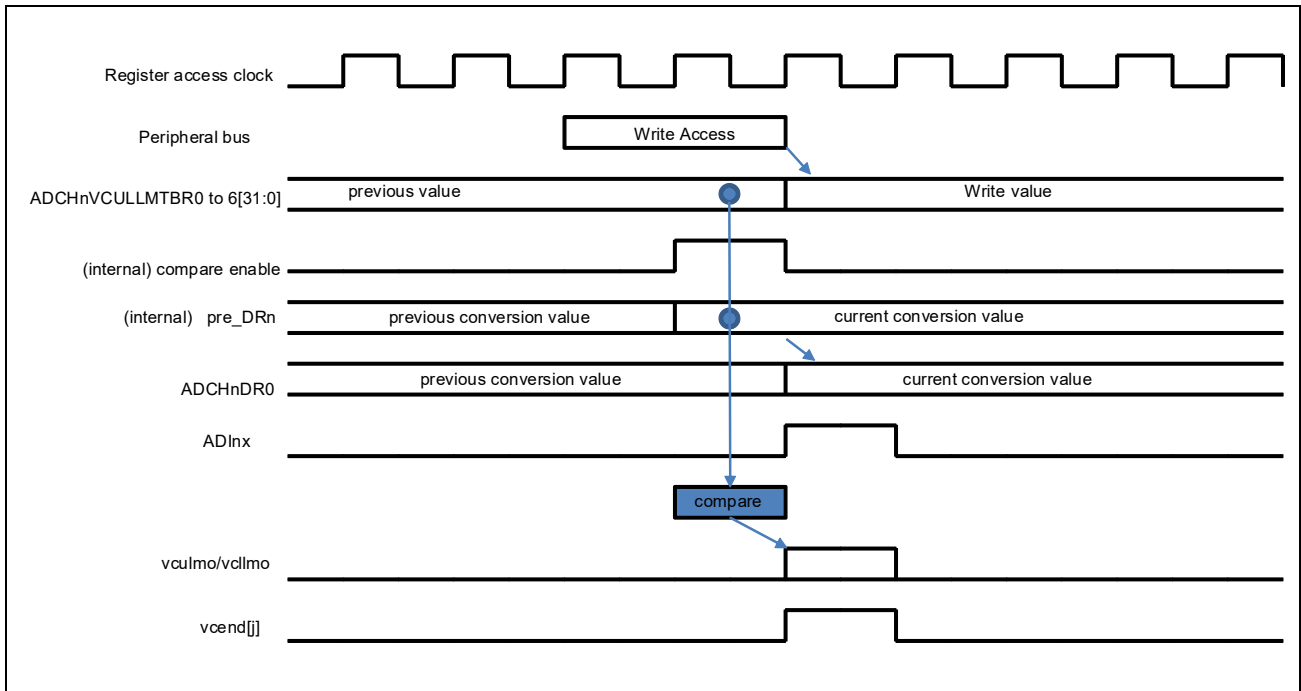


Figure 34.28 Timing Chart of Competition of Storing of ADCHnDRj and Rewriting of ADCHnVCULLMTBR

34.4.10 Wiring-Break Detection Function

The wiring-break detection function detects a wiring-break of the ANI. If a wiring-break is present, the conversion result attenuates to approximately 0 V, and an abnormal value is detected in the conversion result by repeating the A/D conversion several times. This can be determined as a wiring-break. For details timing, refer to **Figure 34.29**.

[Feature]

- (1) Users can select desired physical channels for which the wiring-break is to be detected.

[Recommendation flow]

- (1) Set registers according to the initial settings (**Figure 34.5**).
- (2) Set CNVCLS[2:0] in the virtual channel register ADCHnVCRj to 0_H and set GCTRL[5:0] to select desired channels.
- (3) Set ODE in the wiring-break detection control register to 1_H and set ODPW[5:0] to specify the desired wiring-break detection pulse width.
- (4) Assert SG0 to SG4 trigger signals to perform an A/D conversion.
- (5) When the read A/D conversion result has attenuated to approximately 0V after repeating A/D conversion several times, judge that a wiring-break has occurred.

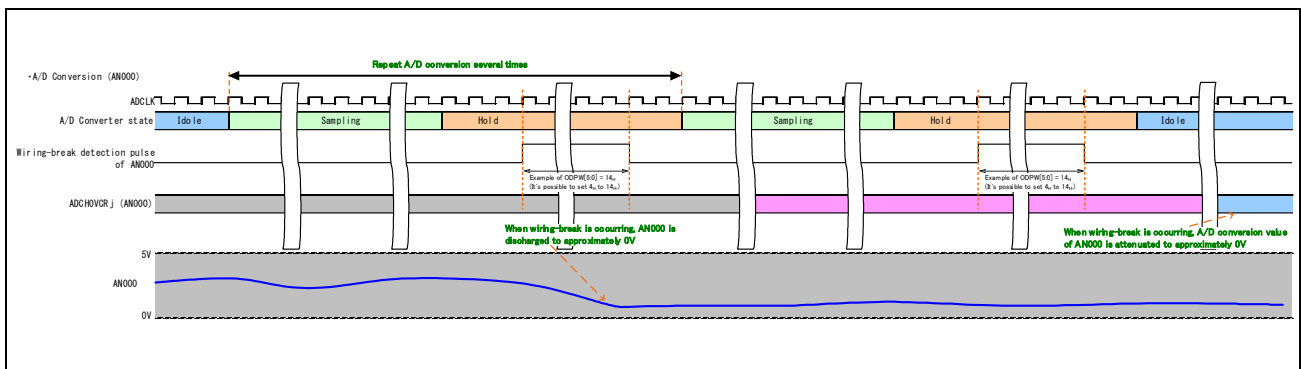


Figure 34.29 Timing Chart of Wiring-Break Detection Function

Note: About prohibiting using the pin level self-diagnosis function and the wiring-break-detection functions at same time, refer to **Section 34.4.11.1, Pin-Level Self-Diagnosis**.

34.4.11 Self-Diagnostic Functions

The ADC is equipped with the following four self-diagnostic functions.

- Pin-level self-diagnosis
- A/D conversion circuit self-diagnosis
- Wiring-break detection self-diagnosis
- Secondary power supply voltage monitor self-diagnosis

34.4.11.1 Pin-Level Self-Diagnosis

The pin-level self-diagnosis performs an A/D conversion that is set to a different voltage for even-numbered physical channel groups and odd-number physical channel groups to check for an abnormal path from the analog input.

When an A/D conversion result is not a static value, there is a possibility that the A/D converter is broken.

[Analog input route]

I/O Buffer → RRAMP → A/D Converter

[Recommendation flow]

Set the virtual channel register j (ADCHnVCRj) and pin level self-diagnosis register (ADCHnTDCR) according to the initial setting flow.

ADCHnVCRj

VCR number	CNVCLS[2:0]	GCTRL[5:0]
N	0H	ANlxx number

ADCHnTDCR

ADCHnTDCR		Injection Voltage to Physical Channel Group	
TDE	TDLV[1:0]	Even Group	Odd Group
0	*	—	—
1	0H	AnVSS	AnVCC
1	1H	AnVCC	AnVSS
1	2H	AnVSS	1/2 x AnVCC
1	3H	1/2 x AnVCC	AnVSS

Allocate the above-mentioned virtual channel n to scan group x.

Note: About prohibiting using the pin level self-diagnosis function and the wiring-break-detection functions at same time, set ADCHnODCR.ODE, ADCHnODCR.ODDE and ADCHnTDCR.TDE as the following.

Setting			Pin function			Operation
ODDE	ODE	TDE	Analog input	Discharge	Voltage injection	
0	0	0	✓	—	—	Normal
0	0	1	—	—	✓	Pin level self-diagnosis
0	1	0	✓	✓	—	Wiring-break detection
1	1	0	—	✓	—	Wiring-break detection self-diagnosis
Other than above						Setting prohibited

✓: Valid —: Invalid

34.4.11.2 A/D Converter Self-Diagnosis

The A/D converter self-diagnosis is used to verify that A/D conversion operates correctly.

When the A/D conversion value is different from the expected value, there is a possibility that the A/D converter is broken.

The voltage value setting is made in GCTRL[4:0] when CNVCLS[2:0] = 3_H, and can be converted for $AnVREFH \times 1$, $AnVREFH \times 3/4$, $AnVREFH \times 1/2$, $AnVREFH \times 1/4$, and $AnVREFH \times 0$.

Features of the A/D conversion circuit self-diagnosis are described below.

[Recommendation flow]

- (1) Set registers according to the initial settings (**Figure 34.5**).
- (2) Set CNVCLS[2:0] in the virtual channel register ADCHnVCRj to 3_H and set GCTRL[5:0] to specify the desired self-diagnosis voltage level.
- (3) Set registers required for A/D conversion according to the initial settings (**Figure 34.5**).
- (4) Assert SG0 to SG4 trigger signals to perform A/D conversions.

34.4.11.3 Wiring-Break Detection Self-Diagnosis

The self-diagnosis function of wiring-break detecting function is used to check that the wiring-break detecting function works normally.

A/D conversion execute in disabling ANI input and pull-down the ANI by ADCHnODCR setting. A/D conversion repeat several times, then the A/D conversion result of pull-down attenuates to approximately 0 V, the wiring-break detecting function is judged to be working normally. For the detail, refer to **Figure 34.30**.

[Feature]

- 1) Users can select desired physical channels for which self-diagnosis of wiring-break detection function is performed by setting virtual channels.

[Recommended flow]

- 1) Set the initial value (refer to **(Figure 34.5)**)
- 2) Set the physical channel by setting of ADCHnVCRj of CNVCLS[2:0] = 0_H and GCTRL[5:0].
- 3) Set ODE and ODDE in the wiring-break detection control register(ADCHnODCR) to 1 and set ODPW[5:0] to specify the desired wiring-break detection pulse width.
- 4) Execute the A/D conversion by asserting SG0-4 trigger.
- 5) After repeat the A/D conversion several times, read the result of A/D conversion value. If the value attenuates to approximately 0V, the wiring-break detecting function is judged to be working normally.

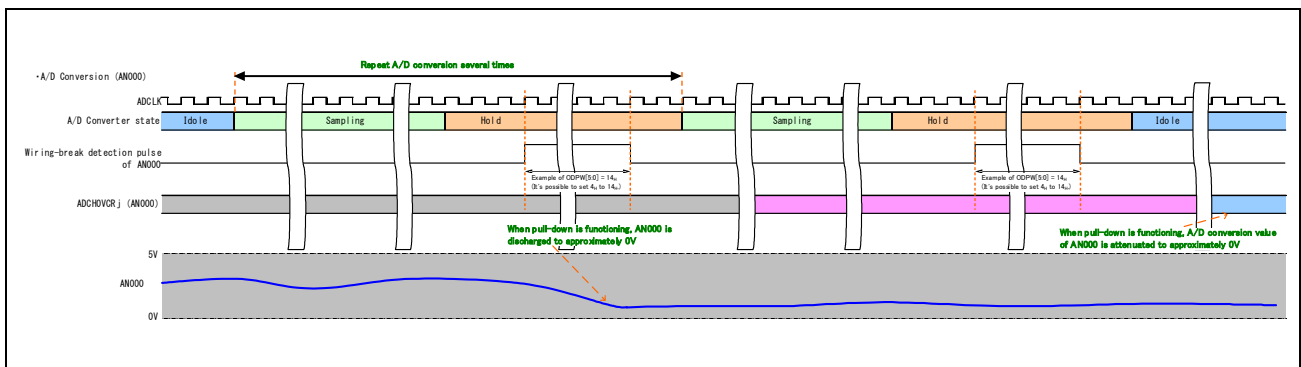


Figure 34.30 Timing Chart of Wiring-Break Detection Self-Diagnosis

Note: About prohibiting using the pin level self-diagnosis function and the wiring-break-detection functions at same time, refer to **Section 34.4.11.1, Pin-Level Self-Diagnosis**.

34.4.11.4 Secondary Power Supply Voltage Monitor Self-Diagnosis

The self-diagnosis function of secondary power supply voltage monitor function is used to check that the secondary power supply voltage monitor function works normally.

A/D conversion is performed when divided resistance is off by VMONVDCR1 and VMONVDCR2. Then the A/D conversion result becomes proximity 0V, the function is judged to be working normally.

When secondary power supply voltage monitor self-diagnosis function is used, A/D conversion of power supply and secondary power supply voltage monitor self-diagnosis should alternately be performed each power supply. For operation example, refer to **Figure 34.31**.

[Recommended flow]

- 1) Set the initial value (refer to **(Figure 34.5)**)
- 2) Set the secondary power supply voltage monitor channel by setting of ADCH0VCRj of CNVCLS[2:0] = 0_H and GCTRL[5:0] (VCC=28, EVCC=29, VDD=30).
- 3) Set VDE1 to 1, set VDE2 to 1.
- 4) A/D conversion is performed, then check that the result is same as power supply voltage.
- 5) Set VDE1 to 0, set VDE2 to 0.
- 6) A/D conversion (secondary power supply voltage monitor self-diagnosis) is performed, then check that the result is approximately 0V.

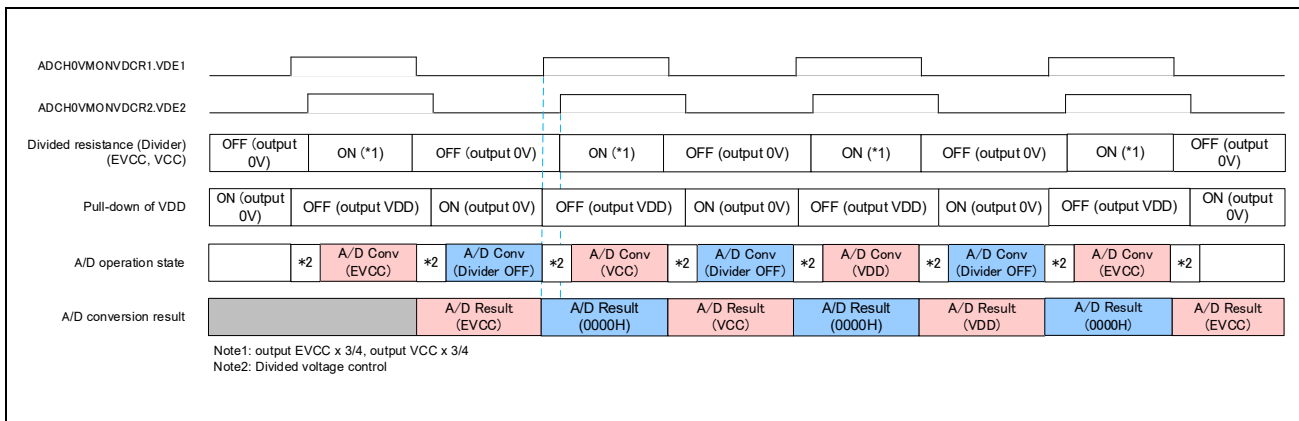


Figure 34.31 Example of Operation of Secondary Power Supply Voltage Monitor Self-Diagnosis

34.4.12 Timing Regulations

The operating timing of ADCH_n is shown below.

Table 34.100 Timing in Normal A/D Conversion

Item	Symbol	Period	Unit
Scan group start delay time	tD	(2 to 4) × Pφ + 5 × Iφ	Pφ (register access clock) Iφ (ADCLK)
Sampling time	tSPL	18 × Iφ	Iφ (ADCLK)
Successive approximation conversion processing time	tSAR	22 × Iφ	Iφ (ADCLK)
Scan group end delay time	tED	(2 to 4) × Iφ + 3 × Pφ	Pφ (register access clock) Iφ (ADCLK)
Scan group processing time	tSG	47 × Iφ + 5 × Pφ to 49 × Iφ + 7 × Pφ	Pφ (register access clock) Iφ (ADCLK)

The processing time for a scan group (tSG) can be obtained from the following formula, where the number of virtual channels is *i* and the number of multicycles is *j* in multicycle scan mode:

$$tSG = tD + (tSPL + tSAR) \times i \times j + tED$$

The processing time for the first cycle of a scan in continuous scan mode is as follows:

$$tD + (tSPL + tSAR) \times i$$

The processing time for the second (or subsequent) cycle of a scan in continuous scan mode is as follows:

$$(tSPL + tSAR) \times i$$

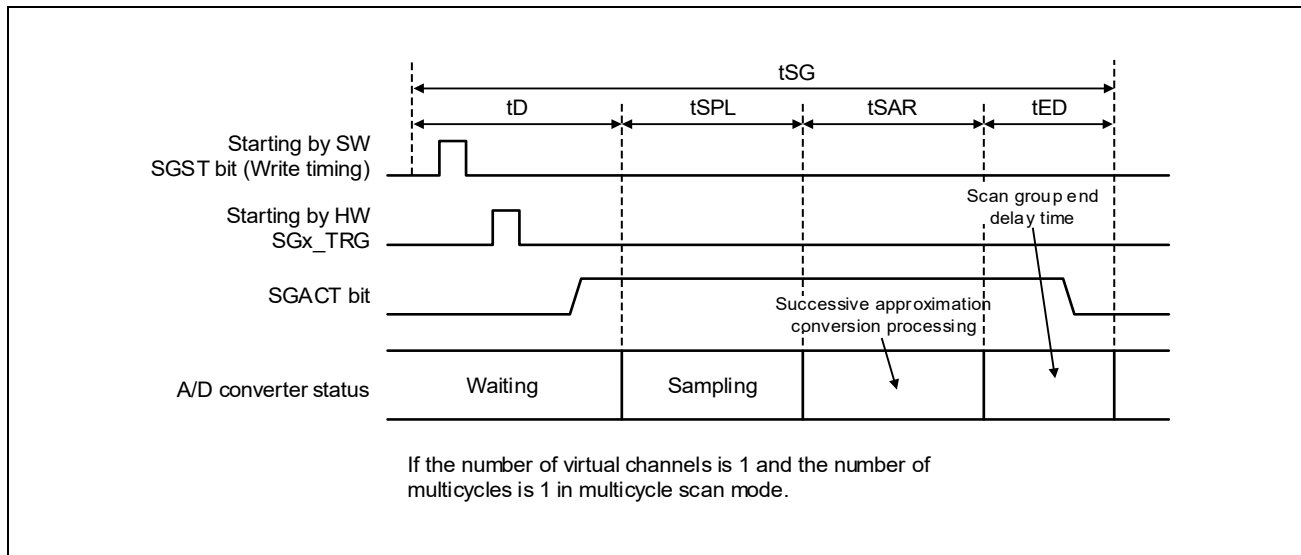


Figure 34.32 Timing Chart of Normal A/D Conversion Operation

34.4.13 Optional Waiting Times Insertion Function of Between Virtual Channels

The wait function is the function to insert any wait time in the processing between a virtual channel processing and a virtual channel processing. The wait time can set each scan group, but cannot set difference time in same scan group.

This function inserts waiting times before virtual channel (other than first virtual channel of first scan) processing of the following.

The conversion class is:

- Normal A/D conversion
- Addition mode A/D conversion
- Normal A/D conversion with the MPX
- Normal A/D conversion with the MPX in addition mode

This function doesn't insert waiting times before virtual channel processing of the following.

The conversion class is:

- A/D converter self-diagnosis

The waiting times is specified by ADCHnSGVCOWRx.VCOW[11:0] to each scan group.

34.4.13.1 Waiting Times Insertion in Normal A/D Conversion

When the normal A/D conversion is executed by using the wait function, a wait is inserted between virtual channel processes regardless of using MPX wait function.

A case by which the number of scan is 3 times is shown in **Figure 34.33**.

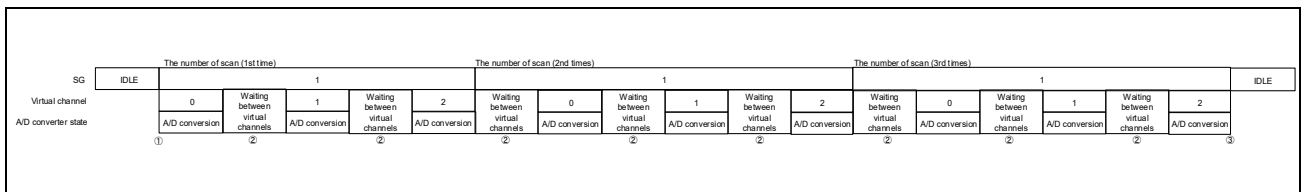


Figure 34.33 Example of Operation of Waiting Timing Insertion in Normal A/D Conversion (the number of scan is 3 times)

It is shown in the following for ① to ③ in **Figure 34.33**.

- ① Waiting times is not inserted before the first virtual channel processing of the first scan. The continuous scan mode is also same.
- ② Waiting times is inserted between virtual channel processing. The continuous scan mode is also same.
- ③ Waiting times is not inserted after the last virtual channel processing of the last scan.

A case by which the number of scan is 1 time is shown in **Figure 34.34**.

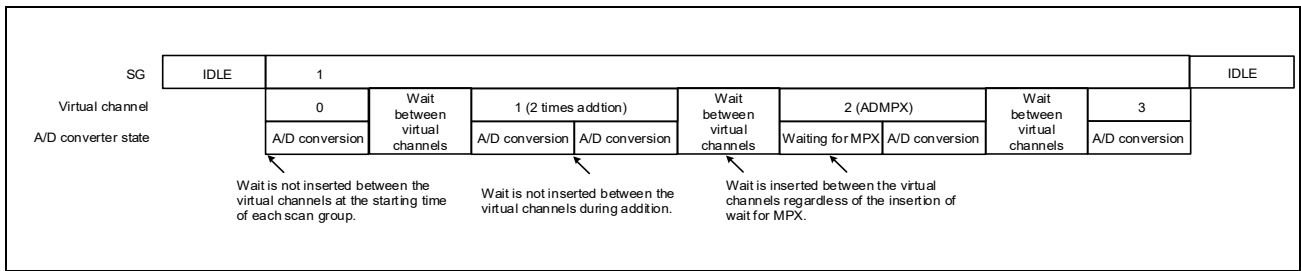


Figure 34.34 Example of Operation of Waiting Timing Insertion in Normal A/D Conversion (the number of scan is 1 time)

When using “Addition mode A/D conversion” or “Normal A/D conversion with the MPX in addition mode”, the waiting times is not inserted in ongoing addition processing (within completion of second A/D conversion in 2 times addition mode, or within completion of fourth A/D conversion in 4 times addition mode).

34.4.13.2 Waiting Time Insertion in Synchronous Suspend Movement

There are two cases that scan group with higher priority (SG2) interrupt to scan group with lower priority (SG1).

<Case 1> Processing a virtual channel of scan group with lower priority (SG1)

<Case 2> Waiting between virtual channels of scan group with lower priority (SG1)

A sample case shows in **Figure 34.35**.

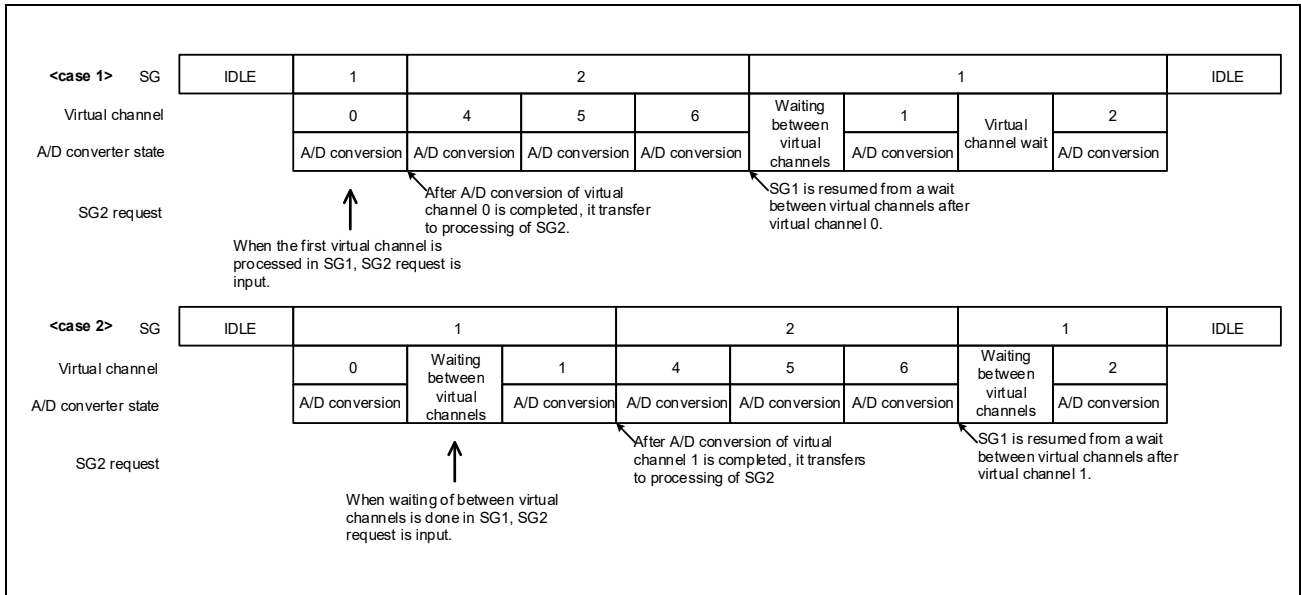


Figure 34.35 Example of Operation of Waiting Timing Insertion in Synchronous Suspend Movement

34.4.13.3 Waiting Times Insertion in Asynchronous Suspend Movement

There are two cases that scan group with higher priority (SG2) interrupt to scan group with lower priority (SG1).

<Case 1> First virtual channel processing of scan group with lower priority (SG1)

<Case 2> A virtual channel processing other than first of scan group with lower priority (SG1).

<Case 3> Waiting times between virtual channels processing of scan group with lower priority (SG1).

When a scan group with lower priority (SG1) restart, a wait is not asserted in <Case 1>, but a wait is asserted in <Case 2> and <Case 3>.

A sample case shows in **Figure 34.36**.

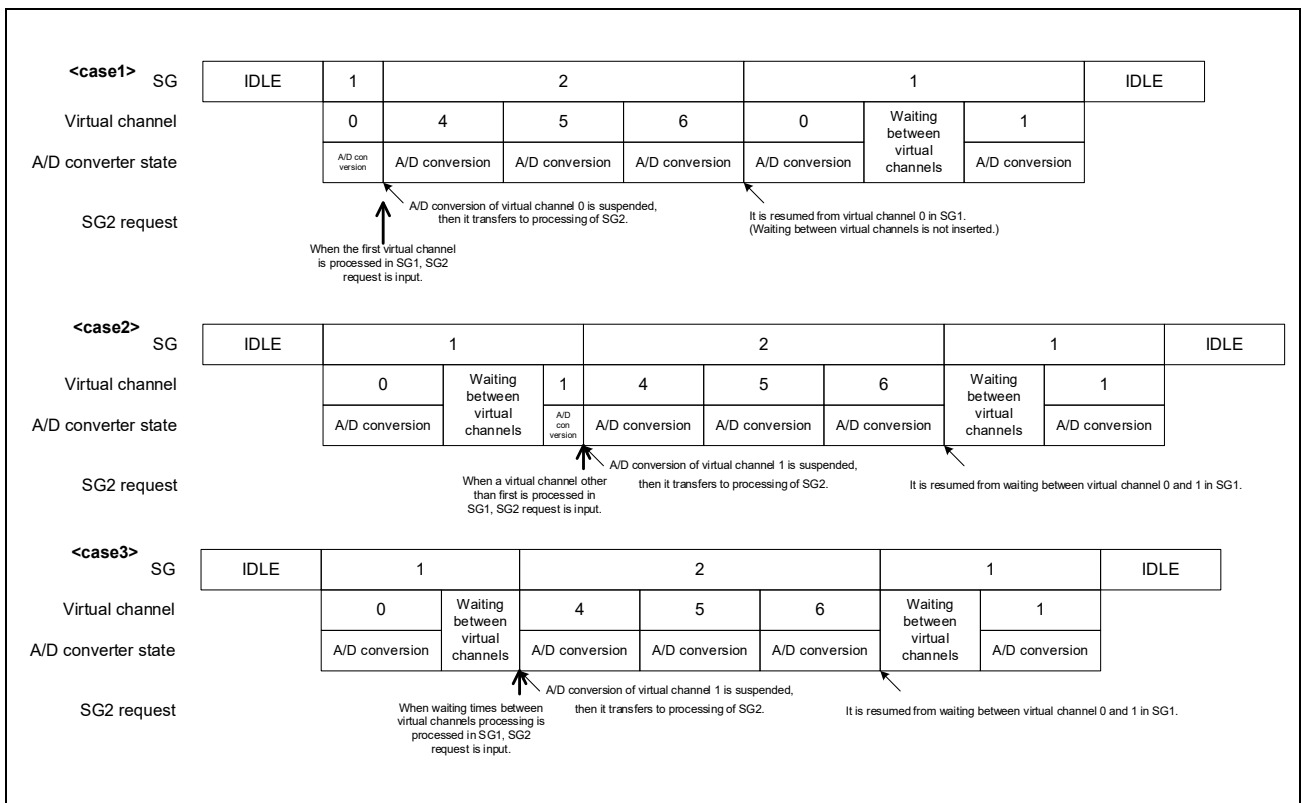


Figure 34.36 Example of Operation of Waiting Timing Insertion in Asynchronous Suspend Movement

34.4.14 Example of Scan Group Stop

The example shows about the function of all scan groups stop (ADCHnADHALTR.HALT) and each scan group stop (ADCHnSGSTPCRx.SGSTP).

34.4.14.1 Example of All Scan Group Stop (A/D Halt)

When ADCHnADHALTR.HALT is set to 1, all scan groups and all AD timers are halted and initialized, and the ADC becomes the idle state (ADCHnSGSRx.SGACT = 0) immediately (by at most 4 register access clocks).

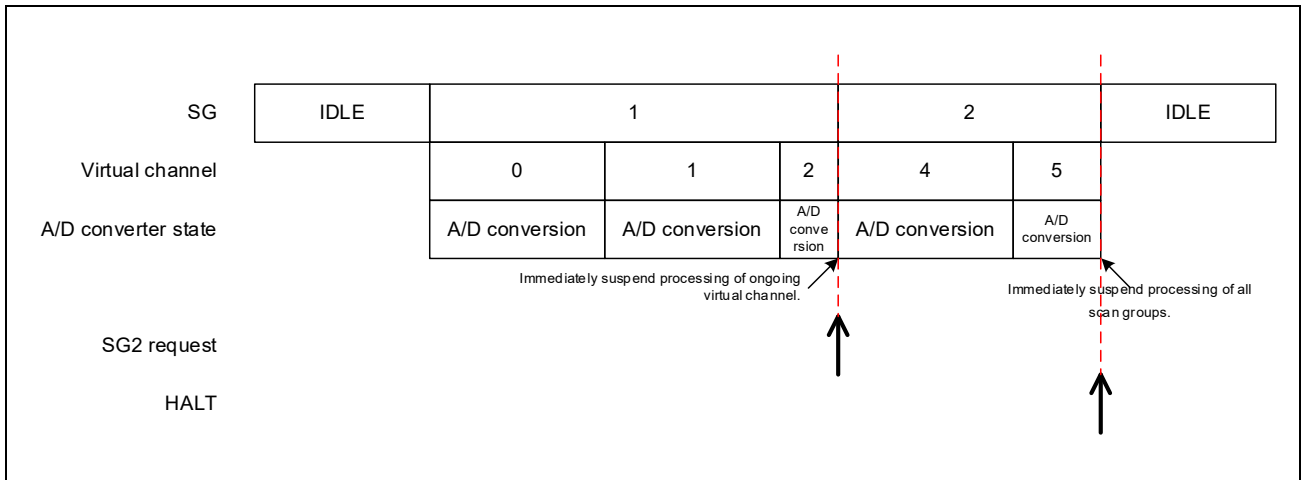


Figure 34.37 Example of All Scan Group Stop (A/D Halt)

34.4.14.2 Example of Scan Group x Stop

When ADCHnSGSTPCR_x.SGSTP is set to 1, corresponding scan group become to be idle state (ADCHnSGSR_x.SGACT = 0) after finishing current virtual channel process. A sample case shows in **Figure 34.38**.

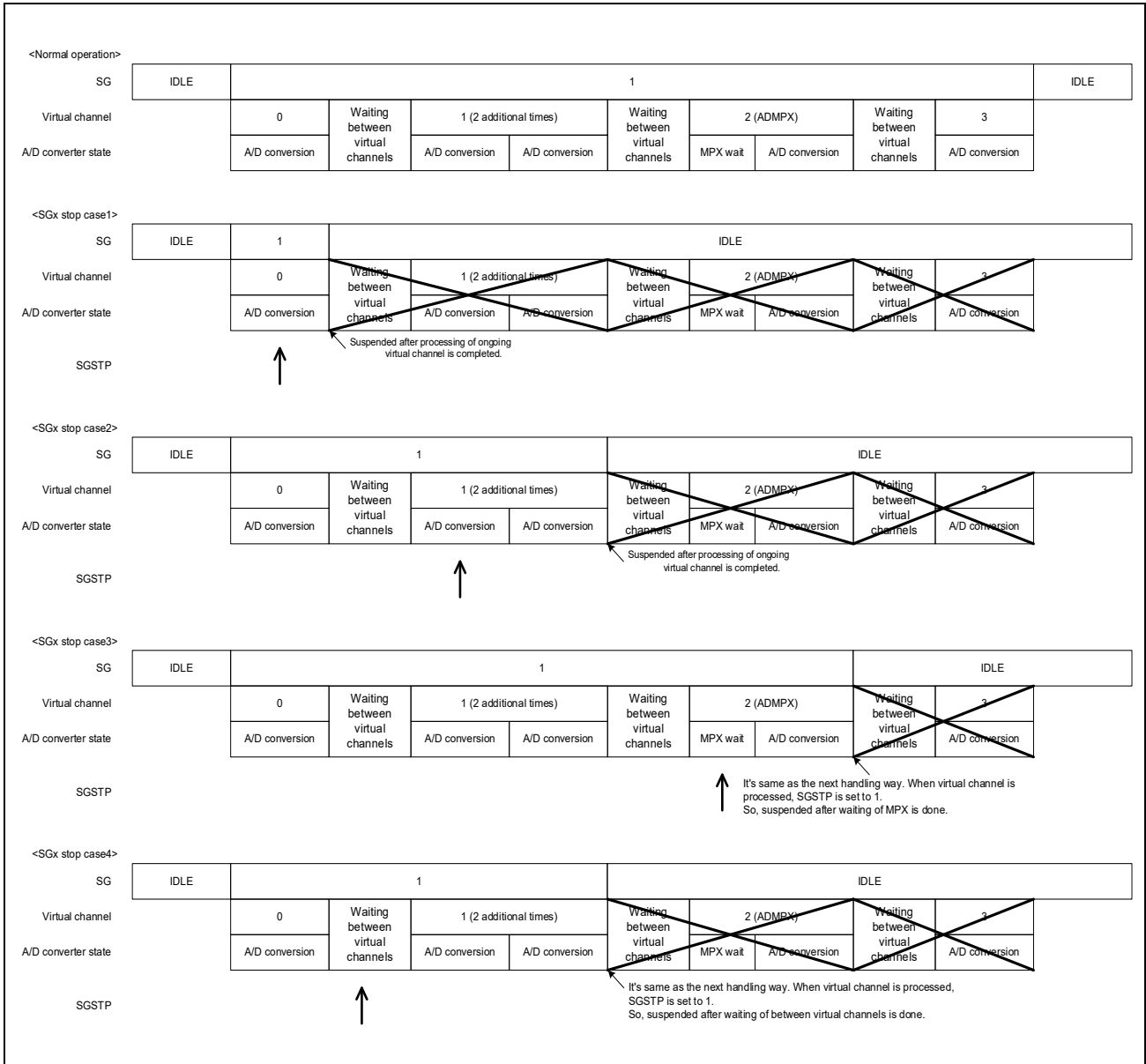


Figure 34.38 Example of Scan Group x Stop 1

ADCHnSGSTPCR_x.SGSTP is set to 1, and when processing of ongoing virtual channel is suspended by asynchronous suspend function of higher priority scan group, processing of ongoing virtual channel is stopped and transits to idle state (SGACT = 0) (Case 1 of **Figure 34.39**).

When ADCHnSGSTPCR_x.SGSTP is set to 1 in suspended state by synchronous suspend function or asynchronous suspend function, ADCH is not resumed and transits to idle state (SGACT = 0) (Case 2 of **Figure 34.39**).

When synchronous suspend is occurred by higher priority scan group during the A/D conversion process, ADCH transits to idle state (SGACT = 0) after the A/D conversion for the virtual channel is finished.

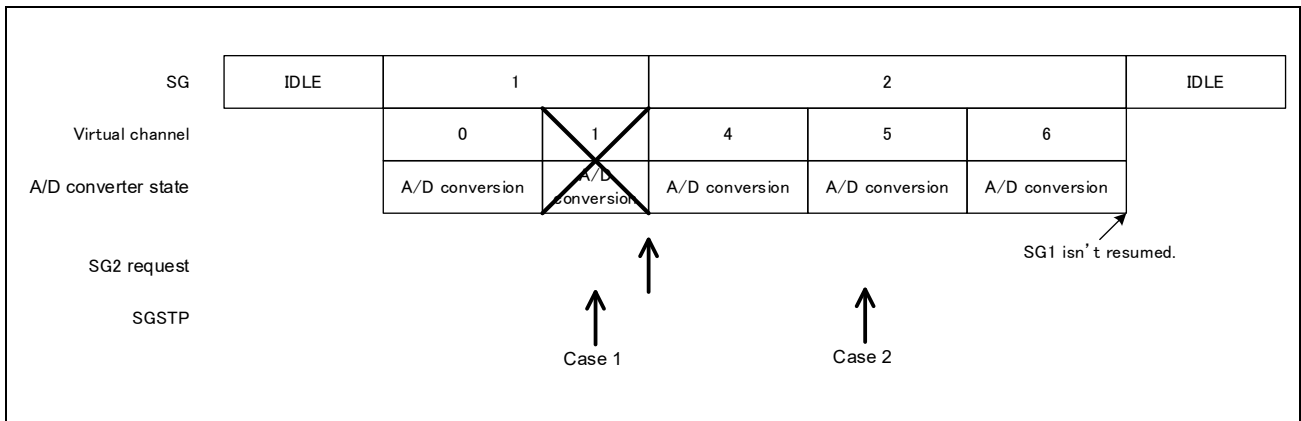


Figure 34.39 Example of Scan Group x Stop 2

34.4.15 Secondary Power Supply Voltage Monitor Function

ADCH0 can convert the voltage of VCC, EVCC, and the VDD power supply in AD. When voltage divider enable bit 1 and 2 (VDE1 and VDE2) of voltage monitor voltage divider control register 1 and 2 (ADCH0VMONVDCR1 and ADCH0VMONVDCR2) are set as 1, it is possible to perform A/D conversion of each power supply voltage.

About VCC and EVCC, A/D conversion of divided voltage are performed.

For the self-diagnosis flow of the secondary power supply voltage monitor function, refer to **Section 34.4.11.4 Secondary Power Supply Voltage Monitor Self-Diagnosis**.

[Recommendation flow]

- 1) Set the initial value (refer to **Figure 34.5**)
- 2) Set CNVCLS[2:0] = 0H, and set each power supply voltage (VCC=28, EVCC=29, VDD=30) to GCTRL[5:0] of ADCH0VCR0, ADCH0VCR1, and ADCH0VCR2. (When they are three consecutive ADCH0VCRj, arbitrary setting is permitted.)
Additionally, set VCULME=1, VCLLME=1 and VCULLMTBS[2:0] to the corresponding values defined in ADCH0VCULLMTBR.
- 3) Set VDE1 to 1, set VDE2 to 1.
- 4) Set upper limit value and lower limit value of each power supply voltage to ADCH0VCULLMTBR0, ADCH0VCULLMTBR1 and ADCH0VCULLMTBR2 (any of ADCH0VCULLMTBR (ADCH0VCULLMTBR0 to ADCH0VCULLMTBR6) can be used as voltage monitor). For reference value of upper / lower limit value, refer to **Figure 34.40**.
- 5) Allocate AVSEGVCCCHSCR (for VCC), AVSEGEVCCCHSCR (for EVCC) and AVSEGVDDCHSCR (for VDD) a virtual channels which corresponds to each power source. For details of this setting, refer to **Section 34.9 ADC VMON Secondary Error Generator (AVSEG)**.
- 6) Set configuration of a filter to AVSEGVCCCNTCR (for VCC), AVSEGEVCCCNTCR (for EVCC) and AVSEGVDDCNTCR (for VDD). For details of this setting, refer to **Section 34.9 ADC VMON Secondary Error Generator (AVSEG)**.
- 7) Start SG, and perform A/D conversion.

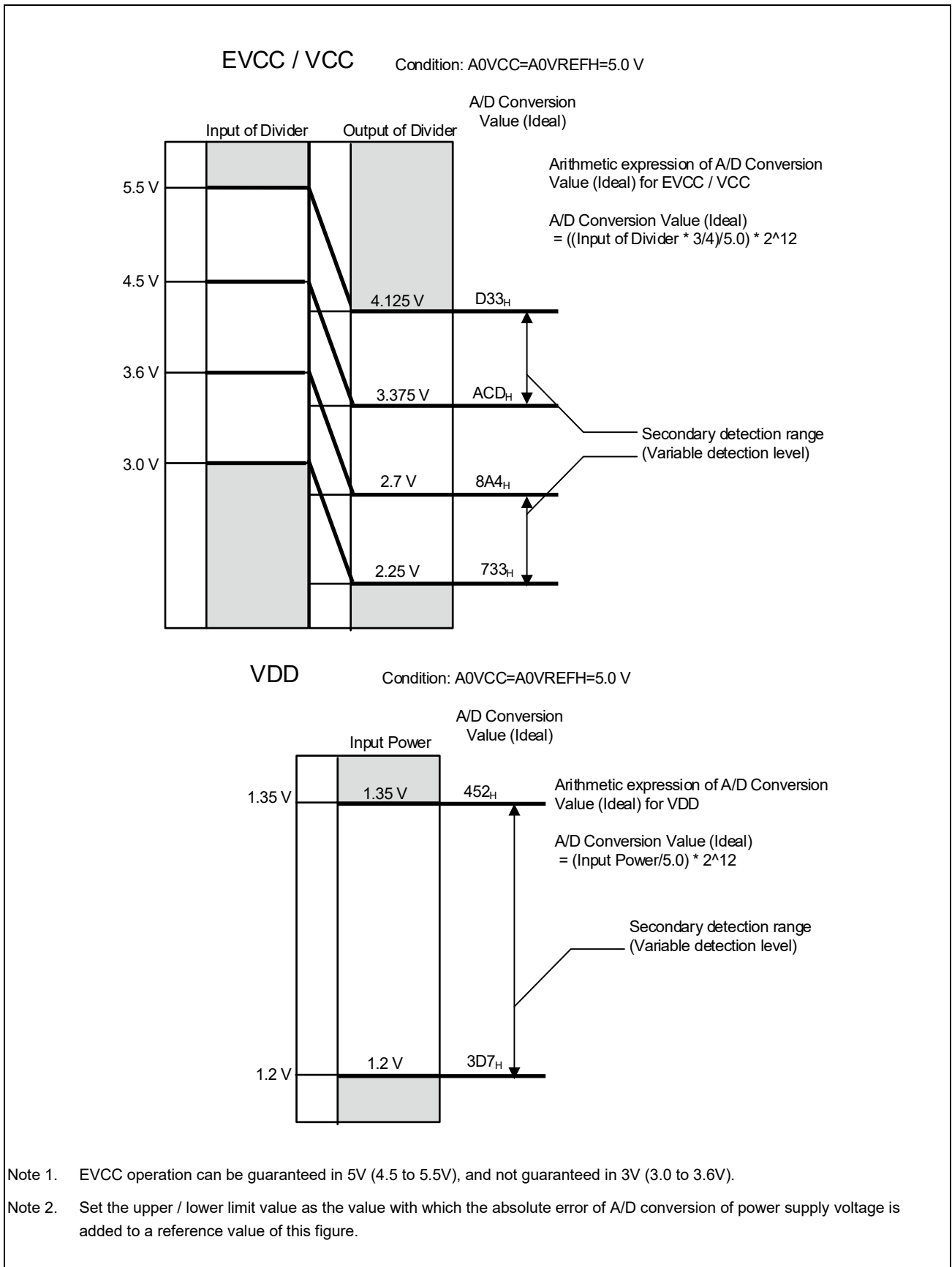


Figure 34.40 Reference Value of Upper / Lower Limit Value of Secondary Power Supply Voltage Monitor

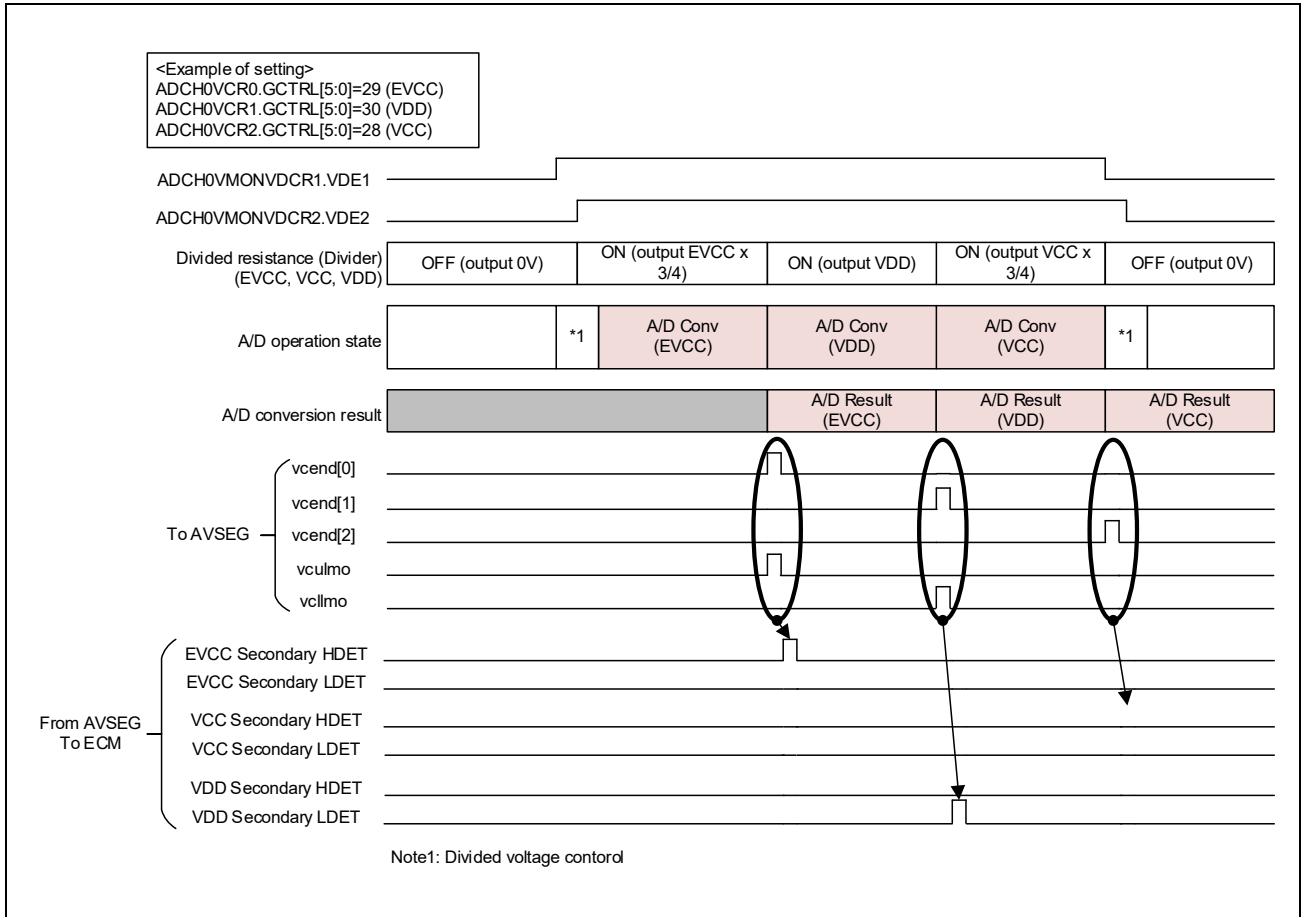


Figure 34.41 Example of Operation of Secondary Power Supply Voltage Monitor

NOTE

- When making it to the module standby, set VDE1 and VDE2 as 0 respectively, before it changes to the module standby.
- Please remove voltage fluctuation of A0VREFH as much as possible.

34.5 Definition of A/D Conversion Accuracy

A/D conversion accuracy is defined below.

- Resolution
Number of digital output codes of the A/D converter
- Quantization error
An error essentially contained in A/D converters, which is given as 1/2LSB (**Figure 34.42**).
- Offset error
Deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from the minimum voltage value 000_H to 001_H. However, the quantization error is not included (**Figure 34.42**).
- Full-scale error
Deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from FFE_H to FFF_H. However, the quantization error is not included (**Figure 34.42**).
- DNL (Differential nonlinear error)
Deviation between the ideal digital output code width (V_q) and the actual digital output code width (V_a), which is given as $(V_a - V_q)/V_q$. However, the offset error, the full-scale error, and the quantization error are not included (**Figure 34.42**).
- INL (Integral nonlinear error)
Deviation of the actual value from the ideal A/D conversion characteristics, from the zero voltage to the full-scale voltage, which is given as an integral of DNL from 000_H to a digital output code. However, the offset error, the full-scale error, and the quantization error are not included (**Figure 34.42**).
- TUE (Total unadjusted error)
Deviation between the digital value and the analog input value. The offset error, the full-scale error, the quantization error, DNL, and INL are included (**Figure 34.42**).

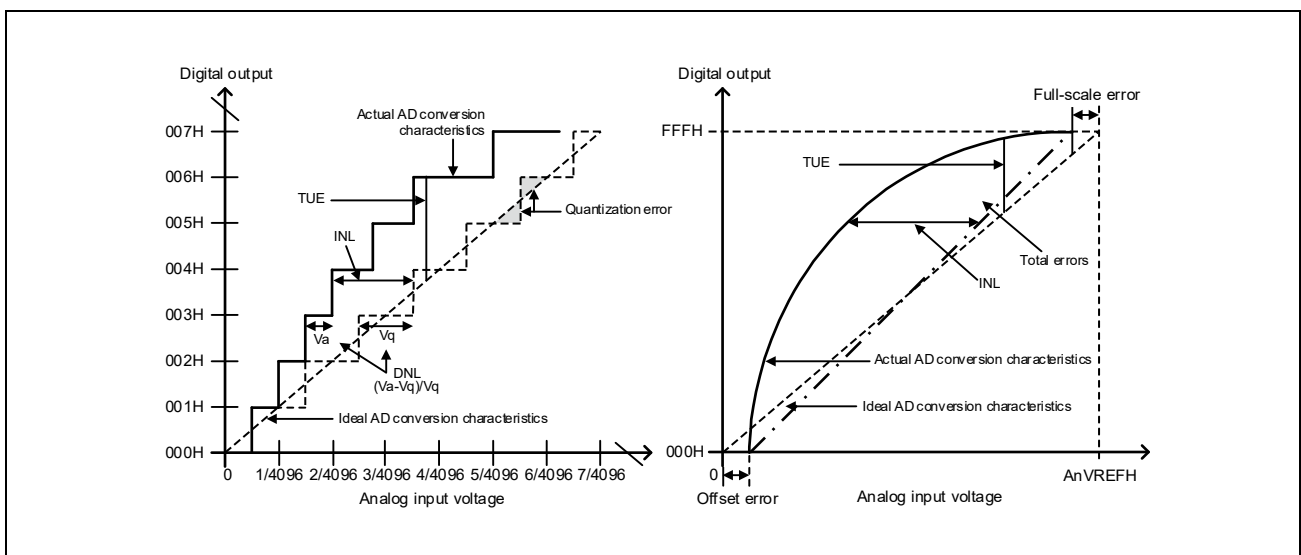


Figure 34.42 Definition of A/D Conversion Accuracy

34.6 Usage Notes

34.6.1 Notes on Using an External Analog Multiplexer

When using an external multiplexer, observe the following notes so as not to cause system issues.

Other than for the exceptions below, set the MPX wait as follows:

- When an MPX value is transferred to a port: Insert a wait of at least 1 μ s.
- When an MPX value is transferred with the SPI interface: Insert a wait of at least the SPI transmission time + 1 μ s.

Exception 1) When an external multiplexer is used for one scan group

When an external multiplexer is used for SG0 with SUSMTD[1:0] set to 1_H, or when an external multiplexer is used for any of SG0 to SG3 with SUSMTD[1:0] set to 2_H, set the MPX wait as follows:

- When an MPX value is transferred to a port: Insert a wait of at least 1 μ s.
- When an MPX value is transferred with the SPI interface: Insert a wait of at least SPI transmission time \times 2.

Exception 2) When an external multiplexer is used for multiple scan groups

When an external multiplexer is used with SUSMTD[1:0] set to 1_H or 2_H, observe the following notes.

- For the starting virtual channel of each scan group, setup so as not to use an external multiplexer. (However, for the start virtual channel of the scan group whose priority is lowest among the scan groups for which the external multiplexer is used, settings to use an external multiplexer does not cause any problem.)
- Up to two scan groups can be transferred with the SPI interface.

Furthermore, set the MPX wait time as follows:

- When an MPX value is transferred to a port: Insert a wait of at least 1 μ s.
- When an MPX value is transferred with the SPI interface: Insert a wait of at least SPI transmission time \times 2.

34.6.2 Notes on Using Analog Input Pins

Do not perform an A/D conversion for the same analog pin with Delta-Sigma ADC, Cyclic ADC and SAR-ADC at the same time. Also, do not perform an A/D conversion with the SAR-ADC for an analog pin that is being used for a digital input. Doing so may degrade the A/D conversion accuracy.

When a digital input or output signal is multiplexed with an analog input signal, it can also be used as a digital general-purpose input or output pin. Changes in a digital input or output during an AD conversion may reduce the precision of conversion. Noise from the operation of digital pins near analog input pins may also reduce the precision of conversion. Notes on how to reduce the effects of digital input and output noise on the results of AD conversion are given below.

(1) Notes on Analog Input Pins

(a) **Place the capacitors of RC circuits as close to the LSI pin as is possible. This reduces the effects of digital inputs and outputs on the precision of conversion. Since the improvement in precision also depends on other conditions of the board, evaluate the situation on the actual board.**

(2) Notes on Digital Input and Output Pins near Analog Pins

- (a) **If digital input through the pin is not essential, disable the digital input.**
- (b) **If a digital signal is input to such a pin, the signal should not include overshoot or undershoot.**
- (c) **Design the board so that load capacitances connected to output pins to suppress discharge currents are small.**
- (d) **Lower the output driving ability of pins that may be affected.**

(3) Software Measures against Effects on the Results of Conversion

- (a) **Use an average of multiple results of AD conversion.**
- (b) **When using multiple consecutive results of ADC conversion, exclude outlying results.**

34.6.3 Module Standby Function

The ADCHn has a module standby function.

Module standby stops the clock supply to ADCH0, ASF0 and AVSEG by setting MSR_SAD.MS_SAD0.

Module standby stops the clock supply to ADCH2 by setting MSR_SAD.MS_SAD2.

Module standby stops the clock supply to ADCH1 by setting MSR_SAD.MS_SAD1.

Module standby stops the clock supply to ADCH3 by setting MSR_SAD.MS_SAD3.

Module standby stops the clock supply to IFC0. For the clock supply stop condition of IFC0, refer to **Table 34.101**.

Module standby stops the clock supply to IFC1. For the clock supply stop condition of IFC1, refer to **Table 34.102**.

Module standby stops the clock supply to ABFG and AIR. For the clock supply stop condition of ABFG and AIR, refer to **Table 34.103**.

Table 34.101 Clock Supply Stop Condition of IFC0 by Module Standby

States	MSR_SAD.MS_SAD0	MSR_SAD.MS_SAD2
Clock Supply Active	0	0 or 1
Clock Supply Active	1	0
Clock Supply Stopped	1	1

Table 34.102 Clock Supply Stop Condition of IFC1 by Module Standby

States	MSR_SAD.MS_SAD1	MSR_SAD.MS_SAD3
Clock Supply Active	0	0 or 1
Clock Supply Active	1	0
Clock Supply Stopped	1	1

Table 34.103 Clock Supply Stop Condition of ABFG and AIR by Module Standby

States	MSR_DAD. MS_DAD00 _10	MSR_DAD. MS_DAD20 _12	MSR_DAD. MS_DAD13 _11	MSR_CAD. MS_CAD	MSR_SAD. MS_SAD0	MSR_SAD. MS_SAD2	MSR_SAD. MS_SAD1	MSR_SAD. MS_SAD3
Clock Supply Active	0	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1
Clock Supply Active	1	0	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1
Clock Supply Active	1	1	0	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1
Clock Supply Active	1	1	1	0	0 or 1	0 or 1	0 or 1	0 or 1
Clock Supply Active	1	1	1	1	0	0 or 1	0 or 1	0 or 1
Clock Supply Active	1	1	1	1	1	0	0 or 1	0 or 1
Clock Supply Active	1	1	1	1	1	1	0	0 or 1
Clock Supply Active	1	1	1	1	1	1	1	0
Clock Supply Stopped	1	1	1	1	1	1	1	1

34.7 Integer/Floating-Point Conversion Module (IFC)

The IFC module can read data in the data register of ADCH via the IFC and convert it to floating-point format.

34.7.1 Overview

34.7.1.1 Function Overview

The IFC has the following features:

- Floating-point format: IEEE754 standard single-precision
- Supported registers: Data registers for all virtual channels of the ADCH
 - IFC0: ADCH0, ADCH2
 - IFC1: ADCH1, ADCH3

34.7.1.2 Block Diagram

Figure 34.43 is a configuration block diagram of the IFC0.

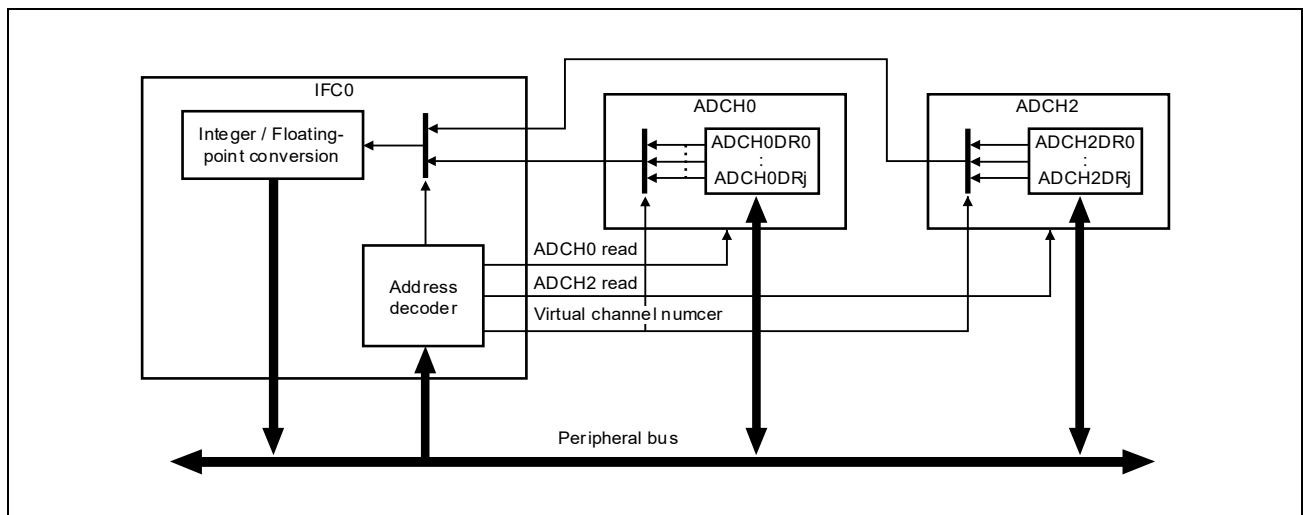


Figure 34.43 IFC0 Block Diagram

Figure 34.44 is a configuration block diagram of the IFC1.

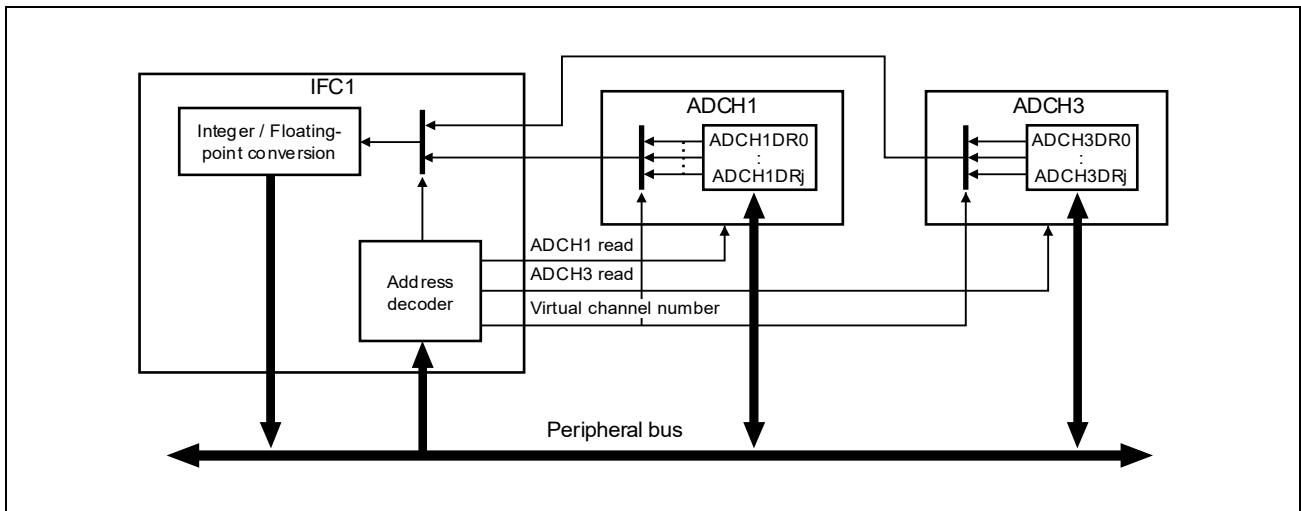


Figure 34.44 IFC1 Block Diagram

34.7.2 Registers

34.7.2.1 List of Registers

Table 34.104 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
IFC0	Floating-Point Data Register 0j	IFC0FDR0j	<IFC0_base> + j × 4 _H	32	—
IFC0	Floating-Point Data Register 2j	IFC0FDR2j	<IFC0_base> + 100 _H + j × 4 _H	32	—
IFC1	Floating-Point Data Register 1j	IFC1FDR1j	<IFC1_base> + j × 4 _H	32	—
IFC1	Floating-Point Data Register 3j	IFC1FDR3j	<IFC1_base> + 100 _H + j × 4 _H	32	—

34.7.2.2 IFC Specific Registers

(1) IFCuFDRj — Floating-Point Data Register nj

IFCuFDRj is a 32-bit read-only register that can read the ADCHnDRj of ADCHn, converting it to the floating-point format. Data in an ADC data register is converted from the signed fixed-point format to the floating-point format. Even data in the signed integer format (DFMT bit in ADCHnADCR2 of ADCHn is set to 1) is regarded as data in the signed fixed-point format, and converted to the floating-point format.

Value after reset: 0000 0000_H

• Floating-point format

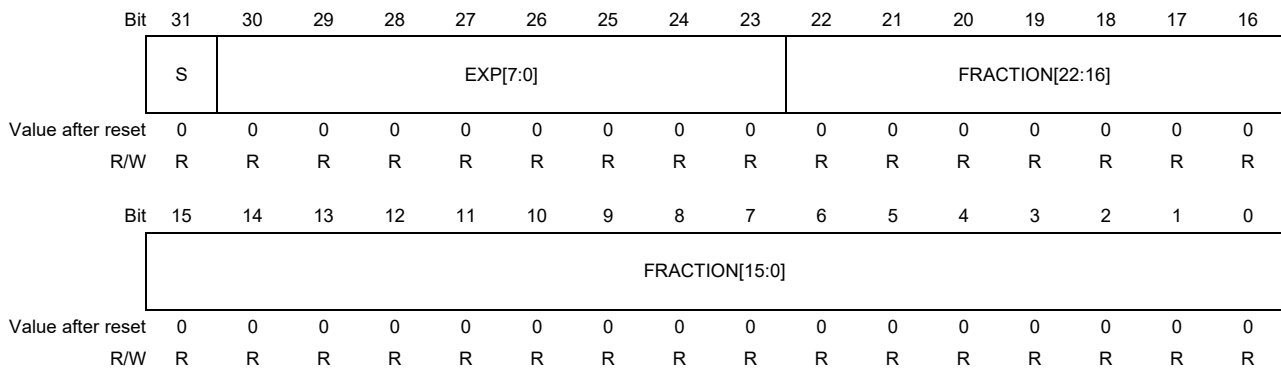
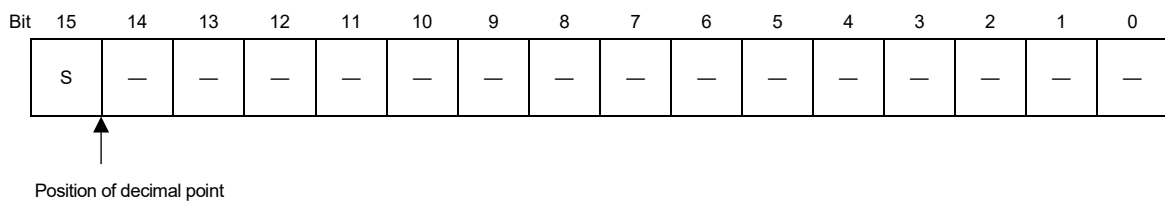


Table 34.105 IFCuFDRj Register Contents

Bit Position	Bit Name	Function
31	S	Sign bit
30 to 23	EXP[7:0]	Exponent
22 to 0	FRACTION[22:0]	Mantissa

Note: Expression: $-1^S \times 2^{(EXP-127)} \times (1 + (Fraction \times 2^{-23}))$

• Reference: ADCHnDRj signed fixed-point format



Floating-point data register 0j of IFC0 (IFC0FDR0j) correspond with virtual channel j of ADCH0 (ADCH0DRj).
 Floating-point data register 2j of IFC0 (IFC0FDR2j) correspond with virtual channel j of ADCH2 (ADCH2DRj).
 Floating-point data register 1j of IFC1 (IFC1FDR1j) correspond with virtual channel j of ADCH1 (ADCH1DRj).
 Floating-point data register 3j of IFC1 (IFC1FDR3j) correspond with virtual channel j of ADCH3 (ADCH3DRj).

34.8 ADC Summation Function (ASF)

The ASF is a function that accumulates the A/D converted value from the ADCH the specified number of times, and then stores the total value in a register.

34.8.1 Overview

34.8.1.1 Function Overview

The ASF has the following features:

- Compatible module: ADCH0
- Number of accumulation channels: 16
- Entry from ADCH:

Upon receiving an ASF entry request, TAG for DFE, and data for DFE from the ADCH, is transferred by the ASF to each accumulation channel (with the same number as the TAG value for DFE) of the ASF.

- Accumulation data register: 32 bits
- Accumulation end interrupt:

Each time accumulation for the number of cycles specified in each channel ends, an interrupt request (ASI00 to ASI15) to the CPU can be generated.

34.8.1.2 Block Diagram

Figure 34.45 illustrates the ASF0 block diagram.

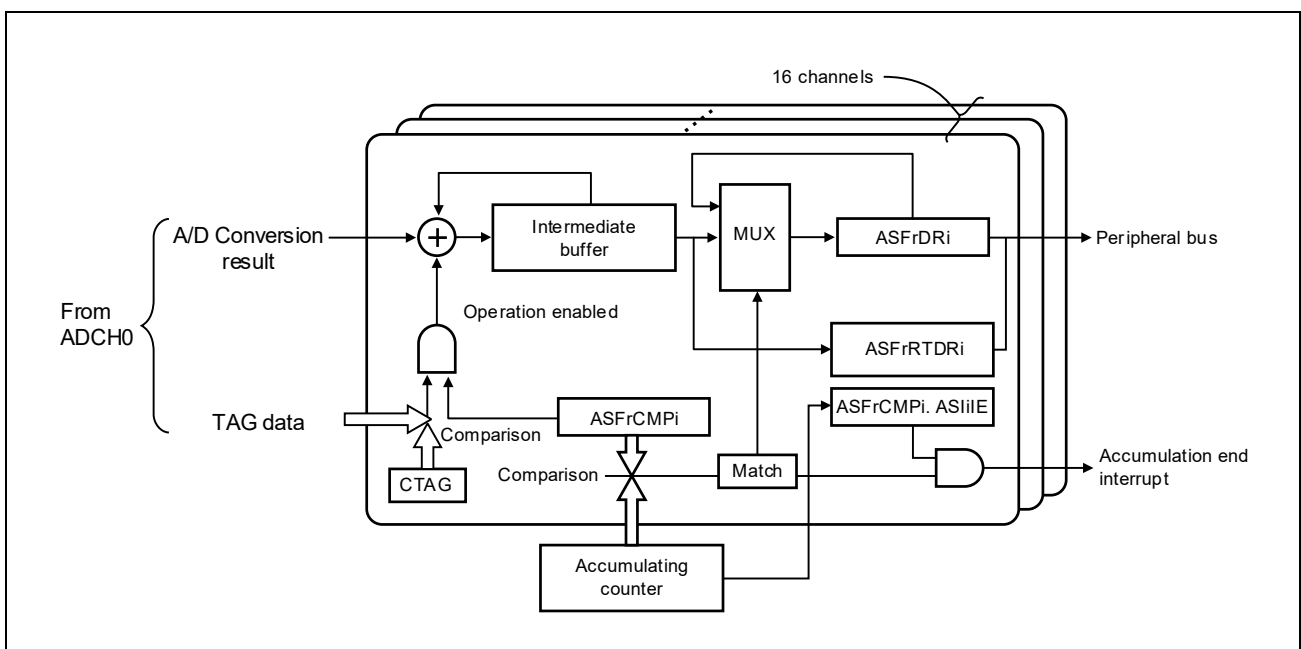


Figure 34.45 ASF0 Block Diagram

34.8.2 Registers

34.8.2.1 List of Registers

Table 34.106 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection
ASFr	Accumulation Data Register 0	ASFrDR00	<ASFr_base> + 00 _H	32	—
ASFr	Accumulation Data Register 1	ASFrDR01	<ASFr_base> + 04 _H	32	—
ASFr	Accumulation Data Register 2	ASFrDR02	<ASFr_base> + 08 _H	32	—
ASFr	Accumulation Data Register 3	ASFrDR03	<ASFr_base> + 0C _H	32	—
ASFr	Accumulation Data Register 4	ASFrDR04	<ASFr_base> + 10 _H	32	—
ASFr	Accumulation Data Register 5	ASFrDR05	<ASFr_base> + 14 _H	32	—
ASFr	Accumulation Data Register 6	ASFrDR06	<ASFr_base> + 18 _H	32	—
ASFr	Accumulation Data Register 7	ASFrDR07	<ASFr_base> + 1C _H	32	—
ASFr	Accumulation Data Register 8	ASFrDR08	<ASFr_base> + 20 _H	32	—
ASFr	Accumulation Data Register 9	ASFrDR09	<ASFr_base> + 24 _H	32	—
ASFr	Accumulation Data Register 10	ASFrDR10	<ASFr_base> + 28 _H	32	—
ASFr	Accumulation Data Register 11	ASFrDR11	<ASFr_base> + 2C _H	32	—
ASFr	Accumulation Data Register 12	ASFrDR12	<ASFr_base> + 30 _H	32	—
ASFr	Accumulation Data Register 13	ASFrDR13	<ASFr_base> + 34 _H	32	—
ASFr	Accumulation Data Register 14	ASFrDR14	<ASFr_base> + 38 _H	32	—
ASFr	Accumulation Data Register 15	ASFrDR15	<ASFr_base> + 3C _H	32	—
ASFr	Accumulation Compare Match Register 0	ASFrCMP00	<ASFr_base> + 40 _H	8, 16	—
ASFr	Accumulation Compare Match Register 1	ASFrCMP01	<ASFr_base> + 44 _H	8, 16	—
ASFr	Accumulation Compare Match Register 2	ASFrCMP02	<ASFr_base> + 48 _H	8, 16	—
ASFr	Accumulation Compare Match Register 3	ASFrCMP03	<ASFr_base> + 4C _H	8, 16	—
ASFr	Accumulation Compare Match Register 4	ASFrCMP04	<ASFr_base> + 50 _H	8, 16	—
ASFr	Accumulation Compare Match Register 5	ASFrCMP05	<ASFr_base> + 54 _H	8, 16	—
ASFr	Accumulation Compare Match Register 6	ASFrCMP06	<ASFr_base> + 58 _H	8, 16	—
ASFr	Accumulation Compare Match Register 7	ASFrCMP07	<ASFr_base> + 5C _H	8, 16	—
ASFr	Accumulation Compare Match Register 8	ASFrCMP08	<ASFr_base> + 60 _H	8, 16	—
ASFr	Accumulation Compare Match Register 9	ASFrCMP09	<ASFr_base> + 64 _H	8, 16	—
ASFr	Accumulation Compare Match Register 10	ASFrCMP10	<ASFr_base> + 68 _H	8, 16	—
ASFr	Accumulation Compare Match Register 11	ASFrCMP11	<ASFr_base> + 6C _H	8, 16	—
ASFr	Accumulation Compare Match Register 12	ASFrCMP12	<ASFr_base> + 70 _H	8, 16	—
ASFr	Accumulation Compare Match Register 13	ASFrCMP13	<ASFr_base> + 74 _H	8, 16	—
ASFr	Accumulation Compare Match Register 14	ASFrCMP14	<ASFr_base> + 78 _H	8, 16	—
ASFr	Accumulation Compare Match Register 15	ASFrCMP15	<ASFr_base> + 7C _H	8, 16	—
ASFr	Accumulation Counter Control Register 0	ASFrCTL0	<ASFr_base> + 80 _H	8, 16, 32	—
ASFr	Accumulation Counter Control Register 1	ASFrCTL1	<ASFr_base> + 84 _H	8	—
ASFr	Accumulation Count Reading Register	ASFrCNT	<ASFr_base> + 8C _H	16	—

Table 34.106 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 00	ASFrRTDR00	<ASFr_base> + 90 _H	32	—
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 01	ASFrRTDR01	<ASFr_base> + 94 _H	32	—
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 02	ASFrRTDR02	<ASFr_base> + 98 _H	32	—
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 03	ASFrRTDR03	<ASFr_base> + 9C _H	32	—
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 04	ASFrRTDR04	<ASFr_base> + A0 _H	32	—
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 05	ASFrRTDR05	<ASFr_base> + A4 _H	32	—
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 06	ASFrRTDR06	<ASFr_base> + A8 _H	32	—
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 07	ASFrRTDR07	<ASFr_base> + AC _H	32	—
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 08	ASFrRTDR08	<ASFr_base> + B0 _H	32	—
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 09	ASFrRTDR09	<ASFr_base> + B4 _H	32	—
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 10	ASFrRTDR10	<ASFr_base> + B8 _H	32	—
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 11	ASFrRTDR11	<ASFr_base> + BC _H	32	—
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 12	ASFrRTDR12	<ASFr_base> + C0 _H	32	—
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 13	ASFrRTDR13	<ASFr_base> + C4 _H	32	—
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 14	ASFrRTDR14	<ASFr_base> + C8 _H	32	—
ASFr	Accumulation Data/Accumulation Counter Intermediate Reading Register 15	ASFrRTDR15	<ASFr_base> + CC _H	32	—

34.8.2.2 ASF Specific Registers

(1) ASFrDRi — Accumulation Data Register i

ASFrDRi is a 32-bit read-only register that stores an accumulation value. When ASFrCNT matches ASFrCMPi, data in intermediate buffer i is updated to ASFrDRi. The ASFrDRi value is retained until the next compare match occurs.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ASFrDRi[20:5]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ASFrDRi[4:0]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 34.107 ASFrDRi Register Contents

Bit Position	Bit Name	Function
31 to 11	ASFrDRi[20:0]	Accumulation Data When ASFrCNT matches ASFrCMPi, data in intermediate buffer i is updated to ASFrDRi.
10 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

A 12-bit (bit14 to bit3) value is extracted from the data for DFE from the ADCHn for use in the input of accumulation as a 12-bit integer. Therefore, even if the valid number of digits of the data for DFE is 13 or 14 bits after addition is performed twice or four times in the ADCHn, the valid number of digits is regarded as 12 bits and accumulation processing is performed. The signed integer format (DFMT[1:0] = 01_B is set in the ADCHn) is regarded as signed fixed-point format and accumulation processing is performed. For this reason, when using the ASF function, conversion-class of ADCHn must be set to normal-A/D-conversion, or A/D-converter-self-diagnosis, or normal-A/D-conversion-with-the-MPX, and DFMT[1:0] must be set to a value different from 01_B.

An effective bit is also dealt with as 12bit in case of 10-bit-format (at the time of DFMT[1:0] = 11_B set by ADCHn), but lower 2bit of the accumulation-input-format, accumulation-value-format and ASFrDRi will be fixed to 0.

ASFrDRi is a format with left-aligned valid bits according to the accumulation count of 21-bit accumulation value.

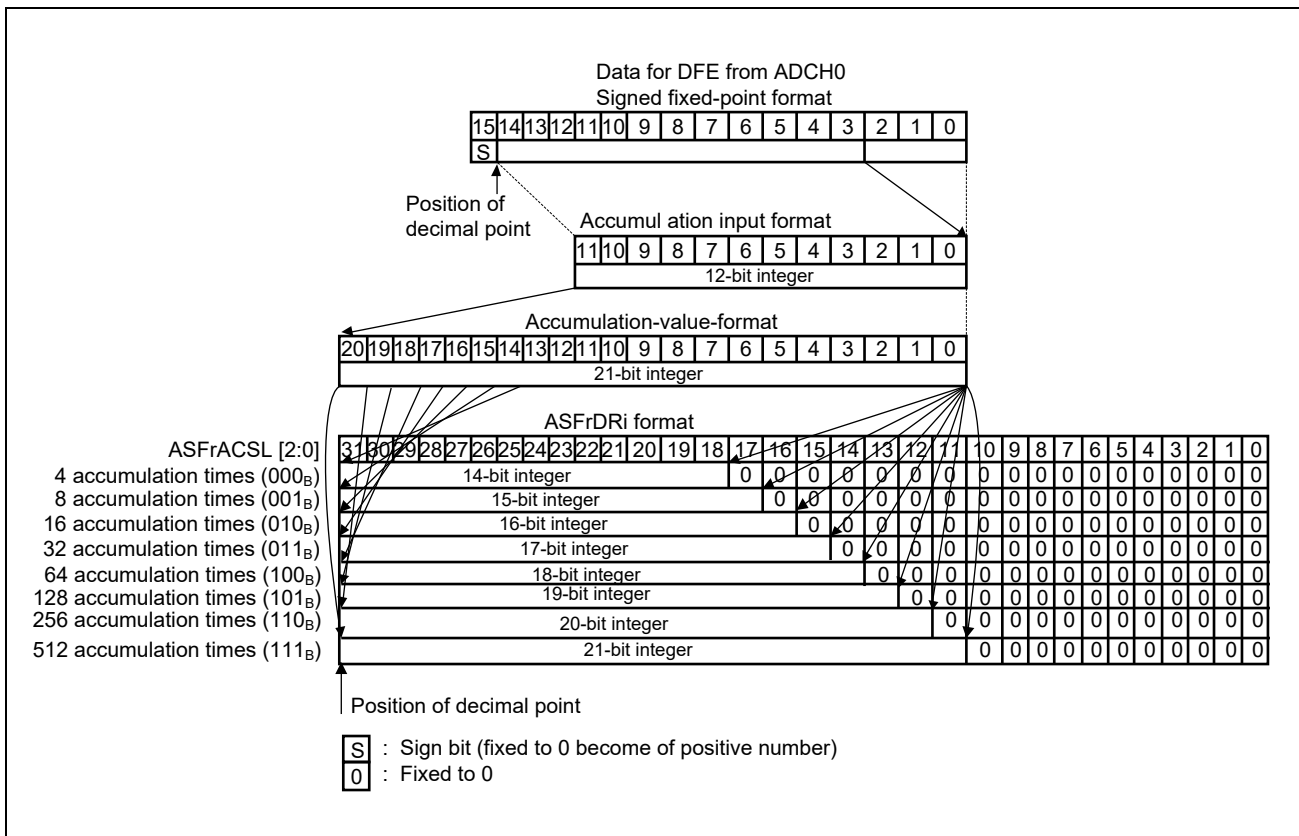


Figure 34.46 Accumulation Data Reading Register i (ASFrDRi) Bit Format

(2) ASFrCMPi — Accumulation Compare Match Register i

ASFrCMPi is a 16-bit readable/writable register that control the accumulation end interrupt output control, enables or disables accumulation processing for accumulation channel i and specifies the ASFrCNT compare match value for updating ASFrDRi. ASFrCMPi is initialized to 0000_H at reset.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ASlIE	—	—	—	—	—	—	—	ASFrCH Ei	—	—	ASFrCMPi[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Table 34.108 ASFrCMPi Register Contents

Bit Position	Bit Name	Function
15	ASlIE	Accumulation end interrupt enable 0: Accumulation end interrupt (ASl) is not output. 1: Accumulation end interrupt (ASl) is output.
14 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	ASFrCHEi	Accumulation Channel i Enable 0: Accumulation is disabled. •Intermediate buffer i is cleared to 0. •ASFrDRi is not updated. •Accumulation end interrupt (ASl) is not generated. 1: Accumulation is enabled. •Each time an ASF entry request of TAG for DFE (TAG for DFE = i) that matches accumulation channel i is accepted, the result of sum of data for DFE and intermediate buffer i is stored in intermediate buffer i (accumulation processing for accumulation channel i). •When ASFrCNT matches ASFrCMPi, data in intermediate buffer i is updated to ASFrDRi and intermediate buffer i is cleared to 0 (ASFrDRi update). •An accumulation end interrupt (ASl) is generated on compare match.
6 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	ASFrCMPi[4:0]	Accumulation Compare Match [When ASFrCTL0.ASFRTD=0] When ASFrCNT matches ASFrCMPi, data in intermediate buffer i is updated to ASFrDRi. [When ASFrCTL0.ASFRTD=1] Regardless of the set ASFrCMPi value, the accumulator i value is stored in ASFrDR i when ASFrCNTi[8:0] = 0. These bits must not be modified during operation (while ASFrST = 1).

CAUTIONS

1. Each time ASFrCNT[8:0] matches {0000_B,ASFrCMPI[4:0]}, the intermediate buffer i value is stored in ASFrDRi[20:0] and an ASli interrupt is generated.
Therefore, the first accumulation count in ASFrDRi[20:0] is not ensured after accumulation processing (where ASFrST is changed from 0 to 1) is started, or immediately after an accumulation channel (for which ASFrCHEi is changed from 0 to 1 during accumulation processing) is enabled (the accumulation count is variable).
For example of operation, see **Section 34.8.3.1, Example of Accumulation Processing Operation**.
 2. Set ASFrCMPI[4:0] to a value less than the accumulation count selected by ASFrACSL[2:0]. If a value equal to or greater than the accumulation count is set in ASFrCMPI[4:0], ASFrCNT[8:0] does not match {0000_B,ASFrCMPI[4:0]}, so the ASFrDRi[20:0] value is not updated.
-

(3) ASFrCTL0 — Accumulation Counter Control Register 0

ASFrCTL0 is a 32-bit readable/writable register that sets count-up conditions for ASFrCNT. ASFrCTL0 is initialized to 0000 0000_H at a reset.

This register must not be modified during operation (while ASFrST = 1).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASFrRTD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ASFrACSL[2:0]			—	—	—	—	ASFrCTAG[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.109 ASFrCTL0 Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	ASFrRTD	Real time display permission of accumulation information 0: Accumulation data / accumulation counter intermediate reading register i (ASFrRTDRi) is not updated. 1: Accumulation data / accumulation counter intermediate reading register i (ASFrRTDRi) is updated with the accumulation data value and accumulation counter value in real time.
15 to 11	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	ASFrACSL[2:0]	Accumulation Count Select 000 _B : 4 times 001 _B : 8 times 010 _B : 16 times 011 _B : 32 times 100 _B : 64 times 101 _B : 128 times 110 _B : 256 times 111 _B : 512 times
7 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	ASFrCTAG[3:0]	Accumulation Count-Up TAG When an ASF entry request of TAG for DFE that matches ASFrCTAG is accepted, ASFrCNT starts to count-up.

(4) ASFrCTL1 — Accumulation Counter Control Register 1

ASFrCTL1 is an 8-bit readable/writable register that enables (starts) or disables (stops) counting up by ASFrCNT. ASFrCTL1 is initialized to 00_H at a reset.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ASFrST
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 34.110 ASFrCTL1 Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ASFrST	Accumulation Counter Start 0: The accumulation counter and all intermediate buffers are cleared to 0. 1: Counting up and accumulation processing by the accumulation counter are enabled. When an ASF entry request of TAG for DFE that matches ASFrCTAG is accepted with ASFrST set to 1, ASFrCNT starts counting up.

(5) ASFrCNT — Accumulation Count Reading Register

ASFrCNT is an 16-bit read-only register that indicates the count value of the accumulation counter. ASFrCNT is initialized to 0000_H at a reset. When ASFrCTL0.ASFRTD = 0, the current summation count can be read. When ASFrCTL0.ASFRTD = 1, this register value is not guaranteed. This register is updated when the input value dftag[3:0] matches the set ASFrCTL0.ASFCTAG[3:0] value at the rising edge of ASF entry request (asreq).

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ASFrCNT[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 34.111 ASFrCNT Register Contents

Bit Position	Bit Name	Function
15 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	ASFrCNT[8:0]	Accumulation Counter Case of ASFrCTL0.ASFRTD=0 When ASFrST is set to 1, ASFrCNT counts up each time an ASF entry request of TAG for DFE that matches ASFrCTAG is accepted. When ASFrST is set to 0, ASFrCNT is cleared to 0. <ul style="list-style-type: none"> ●When ASFrACSL = 000_B (4 accumulation times) The counter counts up to 3 and then returns to 0. ●When ASFrACSL = 001_B (8 accumulation times) The counter counts up to 7 and then returns to 0. ●When ASFrACSL = 010_B (16 accumulation times) The counter counts up to 15 and then returns to 0. ●When ASFrACSL = 011_B (32 accumulation times) The counter counts up to 31 and then returns to 0. ●When ASFrACSL = 100_B (64 accumulation times) The counter counts up to 63 and then returns to 0. ●When ASFrACSL = 101_B (128 accumulation times) The counter counts up to 127 and then returns to 0. ●When ASFrACSL = 110_B (256 accumulation times) The counter counts up to 255 and then returns to 0. ●When ASFrACSL = 111_B (512 accumulation times) The counter counts up to 511 and then returns to 0. When ASFrCTL0.ASFRTD=1 The read value of this register cannot be guaranteed.

(6) ASFrRTDRi — Accumulation Data/Accumulation Counter Intermediate Reading Register i

ASFrRTDRi is a 32-bit read-only register that stores an accumulation data and accumulation counter value.

When ASFrCTL0.ASFrRTD = 0, the read value of this register cannot be guaranteed.

When ASFrCTL0.ASFrRTD = 1, current accumulation data and current accumulation cycles can be read from this register. This register is updated at the timing of an ASF entry request (asreq).

ASFrRTDRi of the accumulation channel (for example, when dftag=1_H, then it is ASFrRTDR01) indicated by TAG data (dftag [3:0]) is updated. For details, see **Section 34.8.3.1, Example of Accumulation Processing Operation**.

The data format of an accumulation-data/accumulation-counter-intermediate-reading-register-i (ASFrRTDRi) is according to **Figure 34.46** and **Section 34.8.2.2(5), ASFrCNT — Accumulation Count Reading Register**.

When an accumulation counter start is cleared to 0, this register is cleared to 0.

Value after reset: 0000 0000 _H																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ASFrRTDRi[20:5]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ASFrRTDRi[4:0]				—		—		ASFrCNTi[8:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 34.112 ASFrRTDRi Register Contents

Bit Position	Bit Name	Function
31 to 11	ASFrRTDRi [20:0]	Accumulation Data Intermediate Value When ASFrCTL0.ASFrRTD=0 The read value of this register cannot be guaranteed. When ASFrCTL0.ASFrRTD=1 Accumulation data value during the accumulation process.
10 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	ASFrCNTi[8:0]	Accumulation Counter Intermediate Value When ASFrCTL0.ASFrRTD=0 The read value of this register cannot be guaranteed. When ASFrCTL0.ASFrRTD=1 Accumulation counter value during the accumulation process.

CAUTION

ASF is equipped with only 1 accumulation counter.

Therefore, when DFTAG value for more than one ADCHnVCRj.DFTAG[3:0] of ADCHn value overlaps, a mismatch occurs between accumulation-data-intermediate-value and accumulation-counter-intermediate-value in specific accumulation channel.

ADCHnVCRj.DFTAG [3:0] has to be set as a unique value so that the accumulation channel for the requested entry doesn't overlap when more than one accumulation channel is requested for entry.

When an accumulation channel is terminated independently, a value of this register of terminated channel in period of "from terminated time to the first generating time of accumulation end interrupt after restarting" cannot be guaranteed.

34.8.3 Operation

The ADC summation function (ASF) accumulates the results of ADCH_n conversions from the ADCH_n, stores the accumulation value in a register, and generates an interrupt request each time accumulation for the specified number of cycles is completed.

The ASF allows accumulation for up to 16 channels. The number of accumulation cycles is selectable from 4, 8, 16, 32, 64, 128, 256 and 512. However, the number of accumulation cycles is common for all channels (it cannot be set individually for each channel). Furthermore, interrupt timing offset is enabled by register setting.

After ASFrST is set to 1 to enable accumulation processing, the ADCH_n conversion result is added to the accumulation channel that is the same as the TAG value for DFE each time an ASF entry request from the ADCH_n is set to a high level.

The accumulation counter (ASFrCNT[8:0]) counts up when TAG for DFE that matches ASFrCTAG[3:0] is input and an ASF entry request is set to a high level. When ASFrCTAG[3:0] matches the TAG for DFE and the accumulation count reaches number of accumulation cycles specified by ASFrACSL[2:0] - 1, the accumulation counter returns to 0 at the next ASF entry request.

When ASFrCTL0.ASFrRTD = 0, and {0000_B, ASFrCMPi[4:0]} matches with ASFrCNT[8:0], the accumulation data (ASFrDRi[31:0]) is updated and an accumulation end interrupt ASFi is generated.

When ASFrCTL0.ASFrRTD = 1, accumulation data (ASFrDRi[31:0]) is updated when ASFrCNTi[8:0] returns to 0 irrespective of the value of ASFrCMPi[4:0], and accumulation end interrupt (ASFi) is generated at the same time.

When ASFrCTL0.ASFrRTD = 1, accumulation-data-intermediate-value and accumulation-count-intermediate-values can be read from ASFrRTDRi.

ASFrRTDRi of the accumulation channel (for example, when dftag = 1_H, then it is ASFrRTDR01) indicated by TAG data (dftag[3:0]) is updated at the time that ASF entry request (asreq) is requested.

[Operating Example]

- (1) The following registers must be set when state of ASFrCTL1.ASFrST = 0.
ASFrCTL0.ASFrACSL[2:0] = 000_B, ASFrCTL0.ASFrCTAG[3:0] = 1_H, ASFrCMP00.ASFrCHE0=1,
ASFrCMP01.ASFrCHE1 = 1, ASFrCMP00.ASI00IE=1, ASFrCMP01.ASI01IE = 1, ASFrCTL0.ASFrRTD = 0/1,
ASFrCMP00.ASFrCMP00[4:0] = 00_H, ASFrCMP01.ASFrCMP01[4:0] = 01_H.(There are no regulations for the order in which these registers should set.)
- (2) Set ASFrCTL1.ASFrST to 1 to enable (start) the accumulation processing operation.
- (3)
 1. When ASFrCTL0.ASFrRTD = 0, after any accumulation end interrupt ASFi (except for the first one) is generated, ASFrDR00.ASFrDR00[20:0] is read and accumulated value of A/D conversion result is acquired.
See note below for instructions on how to handle the first accumulation end interrupt.
 2. When ASFrCTL0.ASFrRTD = 1, accumulation-data-intermediate-value and accumulation-count-intermediate-value can be read from accumulation-data/accumulation-counter-intermediate-read-register (ASFrRTDRi).
- (4) After the required accumulation value of the ADCH_n conversion results is obtained, set ASFrCTL1.ASFrST to 0 to stop the accumulation processing.

CAUTION

Ignore the first accumulation end interrupt (ASli) (or discard the ASFrDRi[20:0] read value).

Because the first accumulation count after accumulation start is undefined.

34.8.3.1 Example of Accumulation Processing Operation

(1) Operation start

When 1 is written to ASFrST of accumulation counter control register 1 (ASFrCTL1), accumulation processing is started.

• When ASFrRTD = 0:

Figure 34.47 shows an example of accumulation processing operation for two channels when the number of accumulation cycles is set to 4 (ASFrACSL[2:0] = 000_B, ASFrCTAG[3:0] = 1_H, ASFrCMP00[4:0] = 00_H, ASFrCMP01[4:0] = 02_H).

In this example, data (dfdata[15:0]) of dftag[3:0] = 0_H is always 10, data (dfdata[15:0]) of dftag[3:0] = 1_H is always 20 for convenience.

- (1) The counts up operation of ASFrCNT is started (is permitted) by ASFrST=1.
- (2) When the Accumulation Counter (ASFrCNT[8:0]) matches the Accumulation Compare Match ({0000_B, ASFrCMP00[4:0]}), the Accumulation Data (ASFrDR00[31:0]) is updated. The accumulation end interrupt (ASI00) is generated at the same time (the first time). Refer to 'CAUTION' of Section 34.8.3, Operation.
- (3) When ASFrCTAG[3:0] matches dftag[3:0] and the ASF entry request (asreq) is 1, the Accumulation Counter (ASFrCNT[8:0]) perform counts up.
- (4) When ASFrCTAG[3:0] matches dftag[3:0] and the ASF entry request (asreq) is 1 and defined Accumulation Cycles (ASFrACSL[2:0] - 1) is 3, the Accumulation Counter (ASFrCNT[8:0]) returns to 0.
- (5) When the Accumulation Counter (ASFrCNT[8:0]) matches the Accumulation Compare Match ({0000_B, ASFrCMP00[4:0]}), the Accumulation Data (ASFrDR00[31:0]) is updated. The accumulation end interrupt (ASI00) is generated at the same time (the second time).

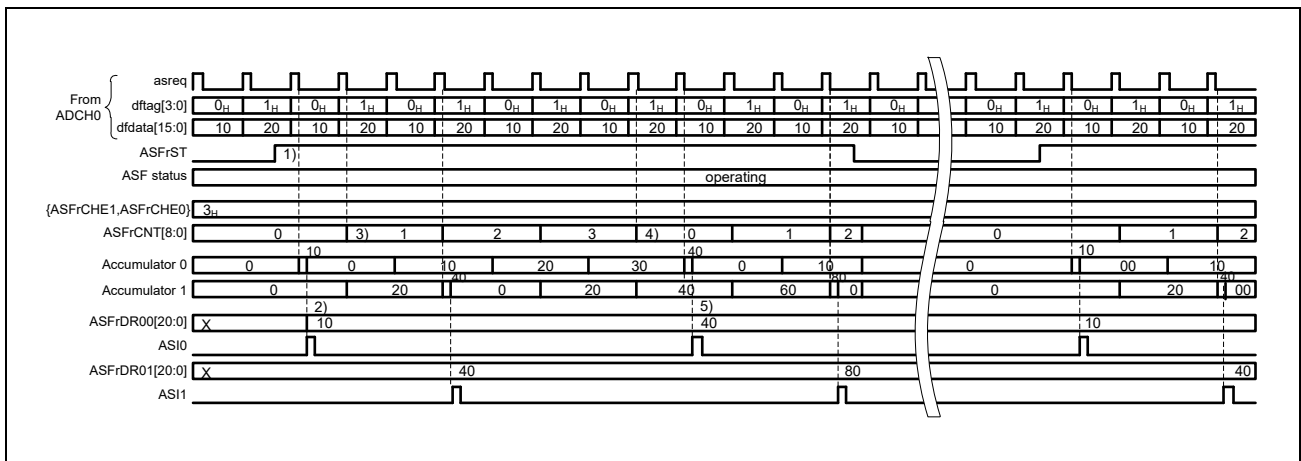


Figure 34.47 Timing of Accumulation Processing Start Operation Example (When ASFrRTD = 0)

• When ASFrRTD=1:

Figure 34.48 shows an example of accumulation processing operation for two channels when the number of accumulation cycles is set to 4 (ASFrACSL[2:0] = 000_B, ASFrCTAG[3:0] = 1_H, ASFrCMP00[4:0] = 00_H, ASFrCMP01[4:0] = 01_H).

In this example, data (dfdata[15:0]) of dftag[3:0] = 0_H is always 10, data (dfdata[15:0]) of dftag[3:0] = 1_H is always 20 for convenience.

- (1) The counts up operation of ASFrCNT is started (is permitted) by ASFrST=1.
- (2) When ASFrCTAG[3:0] matches dftag[3:0] and the ASF entry request (asreq) is 1, ASF transfers from non-operating state to operating state.
- (3) When dftag[3:0] matches 0_H and the ASF entry request (asreq) is 1, the intermediate buffer (ASFrInRTDR00[20:0]) is updated and the intermediate accumulation counter (ASFrRTCNT00[8:0]) perform counts up (+ 1).
- (4) When ASFrCTAG[3:0] matches dftag[3:0] and the ASF entry request (asreq) is 1, the Accumulation Counter (ASFrCNT[8:0]) perform counts up.
- (5) When the intermediate accumulation counter (ASFrRTCNT00[8:0]) returns to 0, the Accumulation Data (ASFrDR00[31:0]) is updated. The accumulation end interrupt (ASI0) is generated at the same time.
- (6) When ASFrCTAG[3:0] matches dftag[3:0] and the ASF entry request (asreq) is 1 and defined Accumulation Cycles (ASFrACSL[2:0] - 1) is 3, the Accumulation Counter (ASFrCNT[8:0]) returns to 0.

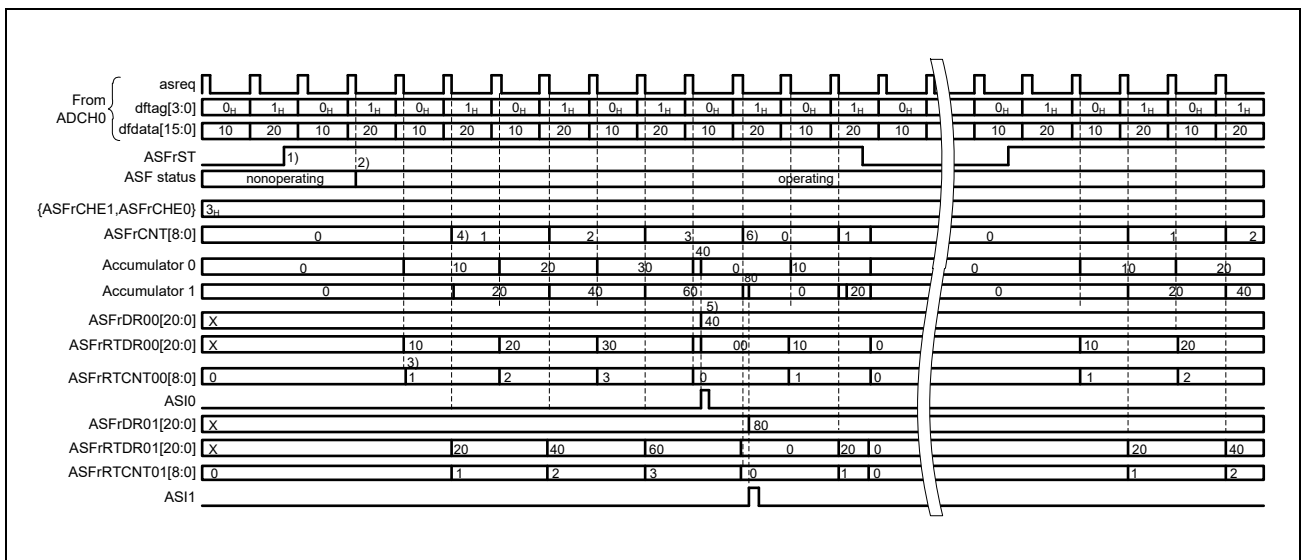


Figure 34.48 Timing of Accumulation Processing Start Operation Example (When ASFrRTD = 1)

(2) Operation Stop and Restart

Figure 34.49 shows an example of accumulation processing stop and restart operation.

When ASFrST is set to 0, the accumulation counter ASFrCNT[8:0] and intermediate buffer i are cleared to 0.

When ASFrST is set to 1, accumulation processing is started again.

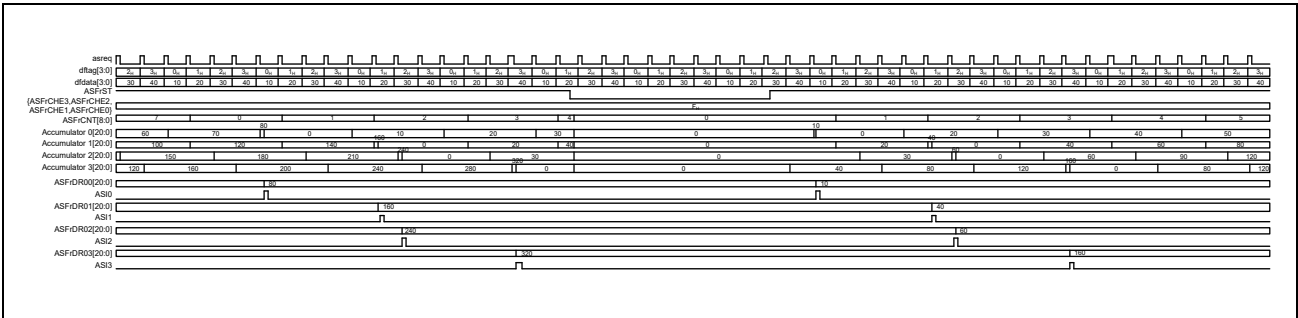


Figure 34.49 Timing of Accumulation Processing Stop and Restart Operation Example

(3) Channel Stop

Figure 34.50 shows an example of accumulation processing stop and restart operation for an accumulation channel.

When ASFrCHEi is set to 0, intermediate buffer i is cleared to 0.

When ASFrCHEi is set to 1, accumulation processing is started again.

When ASFrCHEi is set to 0, ignore the first accumulation end interrupt (ASi) (or discard the ASFrDRi[20:0] read value) because the first accumulation count after accumulation start is undefined.

When ASFrRTD=1, value of accumulation-data/accumulation-counter-intermediate-reading-register (ASFrRTDRi) cannot be guaranteed until the first accumulation end interrupt (ASi) generated.

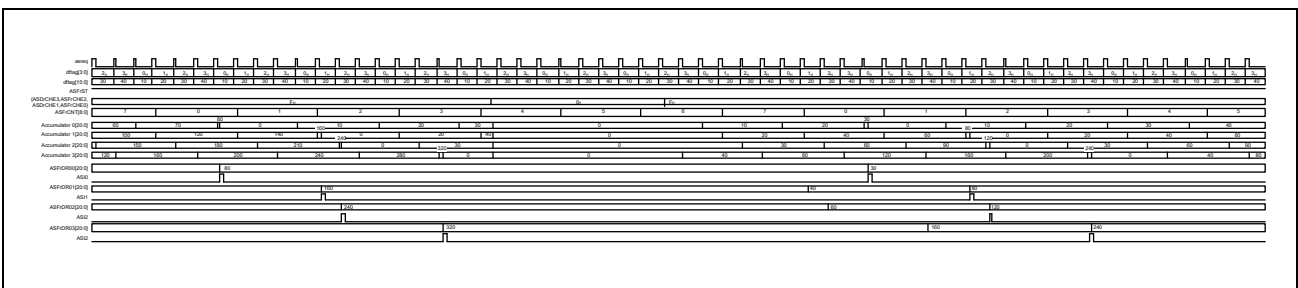


Figure 34.50 Timing of Accumulation Processing Stop Operation Example using ASFrCHEi

(4) Accumulation End Interrupt Request Function

The ASF can output a accumulation end interrupt request (ASLi). An accumulation end interrupt request is output in 1 register access clock period.

When ASFrCMPi.ASiIE is set, a accumulation end interrupt request (ASLi) is output.

For details, see **Figure 34.47** and **Figure 34.48**.

If the accumulation channel i enable bit (ASFrCHEi) or the accumulation counter start bit (ASFrST) is cleared to 0 immediately before the summation ends, the processing ends but a summation end interrupt may be output.

If the accumulation end interrupt enable bit (ASiIE) is cleared to 0 immediately before the accumulation ends, an accumulation end interrupt may be output.

34.9 ADC VMON Secondary Error Generator (AVSEG)

34.9.1 Overview

34.9.1.1 Function Overview

AVSEG is equivalent to the function which notify an upper error pulse (secondary HDET) and a lower error pulse (Secondary LDET) of each power supply (VCC, EVCC and VDD) to ECM in the secondary power supply voltage monitor.

- Upper Error Pulse Control and Lower Error Pulse Control

This function generates an error signal from ADCH0 to ECM if an upper or lower bound is exceeded.

- Noise Filter

The upper and lower error pulses are generated by signals passed through a filter to reduce noise.

CAUTION

The input of both upper and lower pulses into the same module at the same time is prohibited.

34.9.1.2 Block Diagram

Figure 34.51 illustrates the AVSEG block diagram.

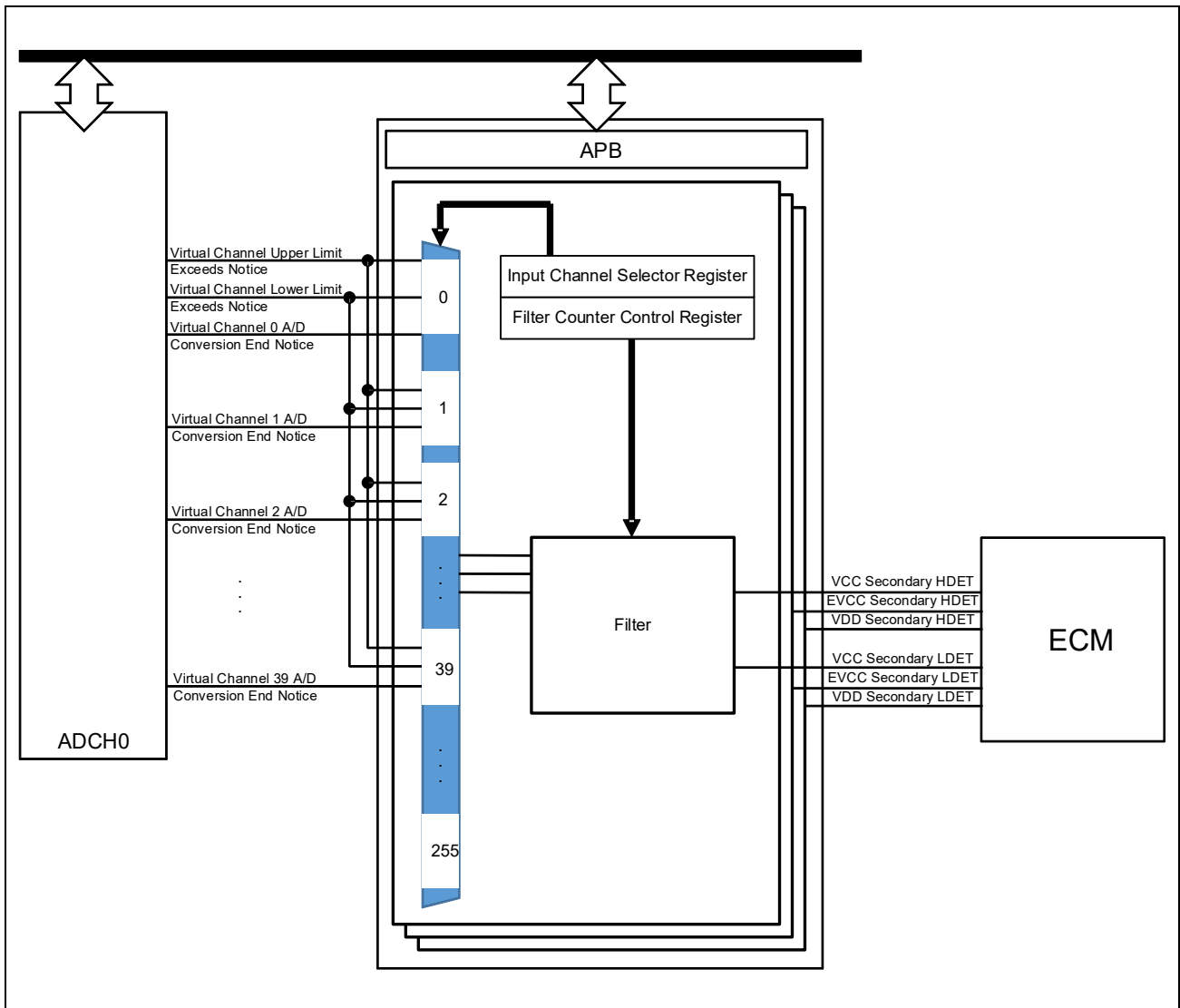


Figure 34.51 AVSEG Block Diagram

34.9.2 Registers

34.9.2.1 List of Registers

Table 34.113 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
AVSEG	Input Channel Selector Register VCC	AVSEGVCCCHSCR	<AVSEG_base> + 00 _H	32	—
AVSEG	Filter Counter Control Register VCC	AVSEGVCCCNTCR	<AVSEG_base> + 04 _H	32	—
AVSEG	Input Channel Selector Register EVCC	AVSEGEVCCCHSCR	<AVSEG_base> + 10 _H	32	—
AVSEG	Filter Counter Control Register EVCC	AVSEGEVCCCNTCR	<AVSEG_base> + 14 _H	32	—
AVSEG	Input Channel Selector Register VDD	AVSEGVDDCHSCR	<AVSEG_base> + 20 _H	32	—
AVSEG	Filter Counter Control Register VDD	AVSEGVDDCNTCR	<AVSEG_base> + 24 _H	32	—

34.9.2.2 AVSEG Specific Registers

(1) AVSEGVCCCHSCR — Input Channel Selector Register VCC

This register selects the input channel.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	AVSEGVCCCHS							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.114 AVSEGVCCCHSCR Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	AVSEGVCCCHS	Input Channel Selector These bits select the channel used for generating error of VCC in the secondary power supply voltage monitor. 00 _H : Selects ADCH0 virtual channel 0 : 27 _H : Selects ADCH0 virtual channel 39 28 _H to FF _H : Reserved

(2) AVSEGVCCNTCR — Filter Counter Control Register VCC

This register controls the filter counter. The noise filtering interval is controlled by using an error counter and a recovery counter. The internal down counter is reset with the written values when this register is modified.

Value after reset: 0000 0101_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	AVSEG VCCEN B	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	AVSEGVCCNRM CNT				—	—	—	—	AVSEGVCCERRCNT			
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.115 AVSEGVCCNTCR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	—	Reserved These bits are always read as 0. The write value should always be 0.
28	AVSEGVCCENB	Filter Enable This bit enable AVSEG for VCC 0: Disabled 1: Enables the filter. When set to disabled all AVSEG for VCC functions are disabled.
27 to 12	—	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	AVSEGVCCNRM CNT	Recovery Counter Settings These bits control the number of counts until the signal is considered recovered. Once the signal is considered out of bounds, it will not be recovered until the set number of consecutive non-error pulses are detected. The counter value is reset when one of the following values is written to AVSEGVCCNRM CNT. 1 _H : Recovery after the signal is within the boundaries for 1 count. : F _H : Recovery after the signal is within the boundaries for 15 consecutive counts.
7 to 4	—	Reserved These bits are always read as 0. The write value should always be 0.

Table 34.115 AVSEGVCCNTR Register Contents (2/2)

Bit Position	Bit Name	Function
3 to 0	AVSEGVCCERRCNT	<p>Error Counter Control</p> <p>These bits control the number of counts until the signal is considered out of bounds. The signal is considered out of bounds once the set number of consecutive error pulses are detected.</p> <p>The counter value is reset when one of the following values is written to AVSEGVCCERRCNT.</p> <p>1_H: Out of bounds when the signal violates the boundaries for 1 count.</p> <p>:</p> <p>F_H: Out of bounds when the signal violates the boundaries for 15 consecutive counts.</p>

CAUTION

Setting AVSEGVCCNTRMCNT and AVSEGVCCERRCNT to 0_H is prohibited.

(3) AVSEGEVCCCHSCR — Input Channel Selector Register EVCC

This register selects the input channel.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	AVSEGEVCCCHS							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.116 AVSEGEVCCCHSCR Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	AVSEGEVCCCHS	Input Channel Selector These bits select the channel used for generating error of EVCC in the secondary power supply voltage monitor 00 _H : Selects ADCH0 virtual channel 0 : 27 _H : Selects ADCH0 virtual channel 39 28 _H to FF _H : Reserved

(4) AVSEGEVCCNTR — Filter Counter Control Register EVCC

This register controls the filter counter. The noise filtering interval is controlled by using an error counter and a recovery counter. The internal down counter is reset with the written values when this register is modified.

Value after reset: 0000 0101_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	AVSEG EVCCENB	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	AVSEGEVCCNRM CNT				—	—	—	—	AVSEGEVCCERRCNT			
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.117 AVSEGEVCCNTR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	—	Reserved These bits are always read as 0. The write value should always be 0.
28	AVSEGEVCCENB	Filter Enable This bit enable AVSEG for EVCC 0: Disabled 1: Enables the filter. When set to disabled all AVSEG for EVCC functions are disabled.
27 to 12	—	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	AVSEGEVCCNRM CNT	Recovery Counter Settings These bits control the number of counts until the signal is considered recovered. Once the signal is considered out of bounds, it will not be recovered until the set number of consecutive non-error pulses are detected. The counter value is reset when one of the following values is written to AVSEGEVCCNRM CNT. 1 _H : Recovery after the signal is within the boundaries for 1 count. : F _H : Recovery after the signal is within the boundaries for 15 consecutive counts.
7 to 4	—	Reserved These bits are always read as 0. The write value should always be 0.

Table 34.117 AVSEGEVCCNTR Register Contents (2/2)

Bit Position	Bit Name	Function
3 to 0	AVSEGEVCCERRCNT	<p>Error Counter Control</p> <p>These bits control the number of counts until the signal is considered out of bounds. The signal is considered out of bounds once the set number of consecutive error pulses are detected.</p> <p>The counter value is reset when one of the following values is written to AVSEGEVCCERRCNT.</p> <p>1_H: Out of bounds when the signal violates the boundaries for 1 count.</p> <p>:</p> <p>F_H: Out of bounds when the signal violates the boundaries for 15 consecutive counts.</p>

CAUTION

Setting AVSEGEVCCNTRMNT and AVSEGEVCCERRCNT to 0_H is prohibited.

(5) AVSEGVDDCHSCR — Input Channel Selector Register VDD

This register selects the input channel.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	AVSEGVDDCHS							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.118 AVSEGVDDCHSCR Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	AVSEGVDDCHS	Input Channel Selector These bits select the channel used for generating error of VDD in the secondary power supply voltage monitor 00 _H : Selects ADCH0 virtual channel 0 : 27 _H : Selects ADCH0 virtual channel 39 28 _H to FF _H : Reserved

(6) AVSEGVDDCNTCR — Filter Counter Control Register VDD

This register controls the filter counter. The noise filtering interval is controlled by using an error counter and a recovery counter. The internal down counter is reset with the written values when this register is modified.

Value after reset: 0000 0101_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	AVSEG VDDEN B	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	AVSEGVDNRCNT				—	—	—	—	AVSEGVDERRCNT			
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.119 AVSEGVDDCNTCR Register Contents

Bit Position	Bit Name	Function
31 to 29	—	Reserved These bits are always read as 0. The write value should always be 0.
28	AVSEGVDENB	Filter Enable This bit enable AVSEG for VDD 0: Disabled 1: Enables the filter. When set to disabled all AVSEG for VDD functions are disabled.
27 to 12	—	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	AVSEGVDNRCNT	Recovery Counter Settings These bits control the number of counts until the signal is considered recovered. Once the signal is considered out of bounds, it will not be recovered until the set number of consecutive non-error pulses are detected. The counter value is reset when one of the following values is written to AVSEGVDNRCNT. 1 _H : Recovery after the signal is within the boundaries for 1 count. : F _H : Recovery after the signal is within the boundaries for 15 consecutive counts.
7 to 4	—	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	AVSEGVDERRCNT	Error Counter Control These bits control the number of counts until the signal is considered out of bounds. The signal is considered out of bounds once the set number of consecutive error pulses are detected. The counter value is reset when one of the following values is written to AVSEGVDERRCNT. 1 _H : Out of bounds when the signal violates the boundaries for 1 count. : F _H : Out of bounds when the signal violates the boundaries for 15 consecutive counts.

CAUTION

Setting AVSEGVDNRCNT and AVSEGVDERRCNT to 0_H is prohibited.

34.9.3 Operation

34.9.3.1 Noise Count Method

The AVSEG noise count methods are shown below. Noise is filtered and reduced by counting the number of times the ADC boundary values are exceeded using the input error pulses.

CAUTION

The input of both upper and lower pulses into the same module at the same time is prohibited.

Example for the upper error pulse AVSEG for VCC:

Method 1:

AVSEGVCCNTR.AVSEGVCCERMNT = 1_H to 3_H and AVSEGVCCNTR.AVSEGVCCNRMNT = 1_H.

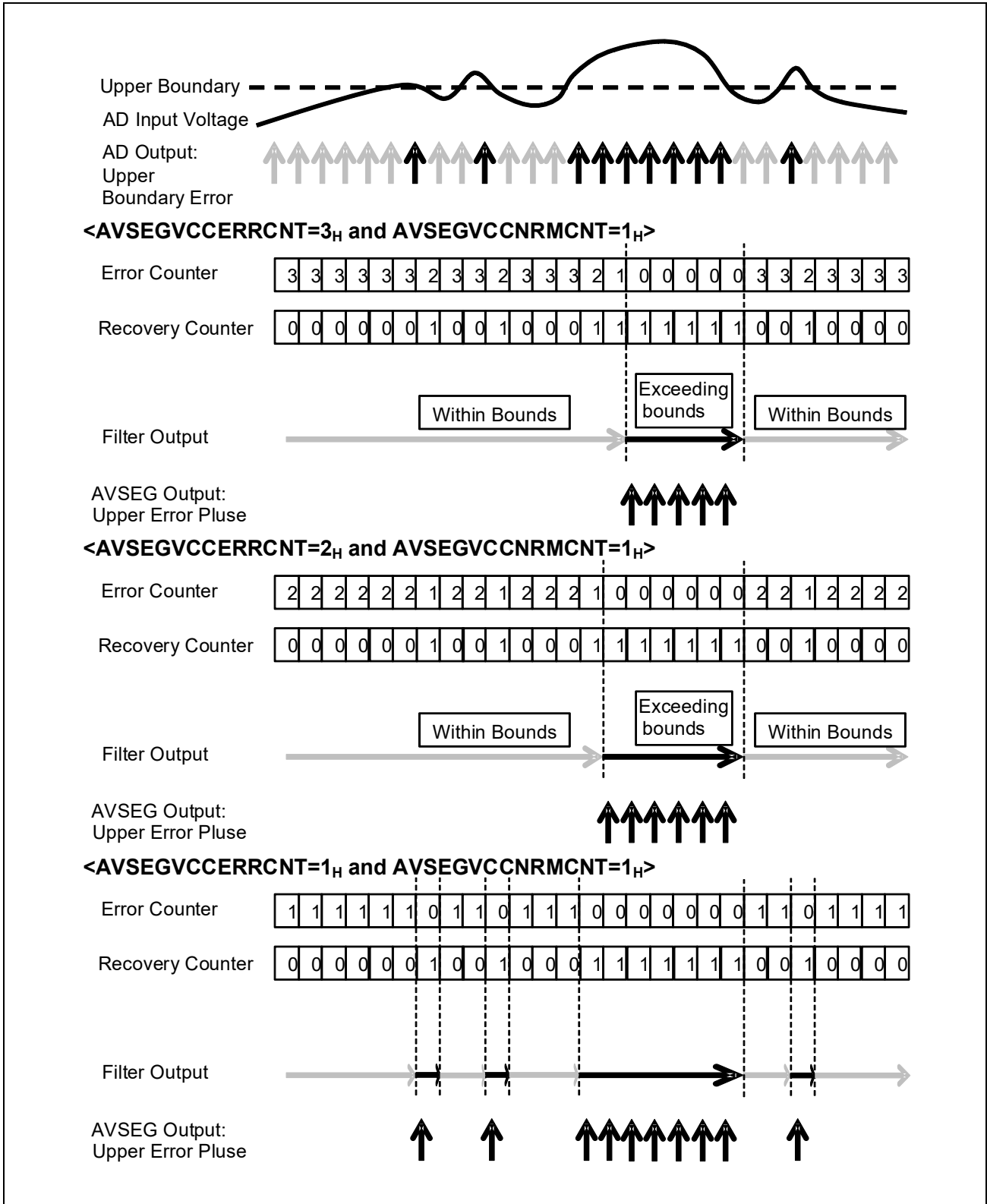


Figure 34.52 Noise Count (1)

Method 2:

AVSEGVCCNTCR.AVSEGVCCERM CNT = 1_H to 3_H and AVSEGVCCNTCR.AVSEGVCCNRMCNT = 2_H.

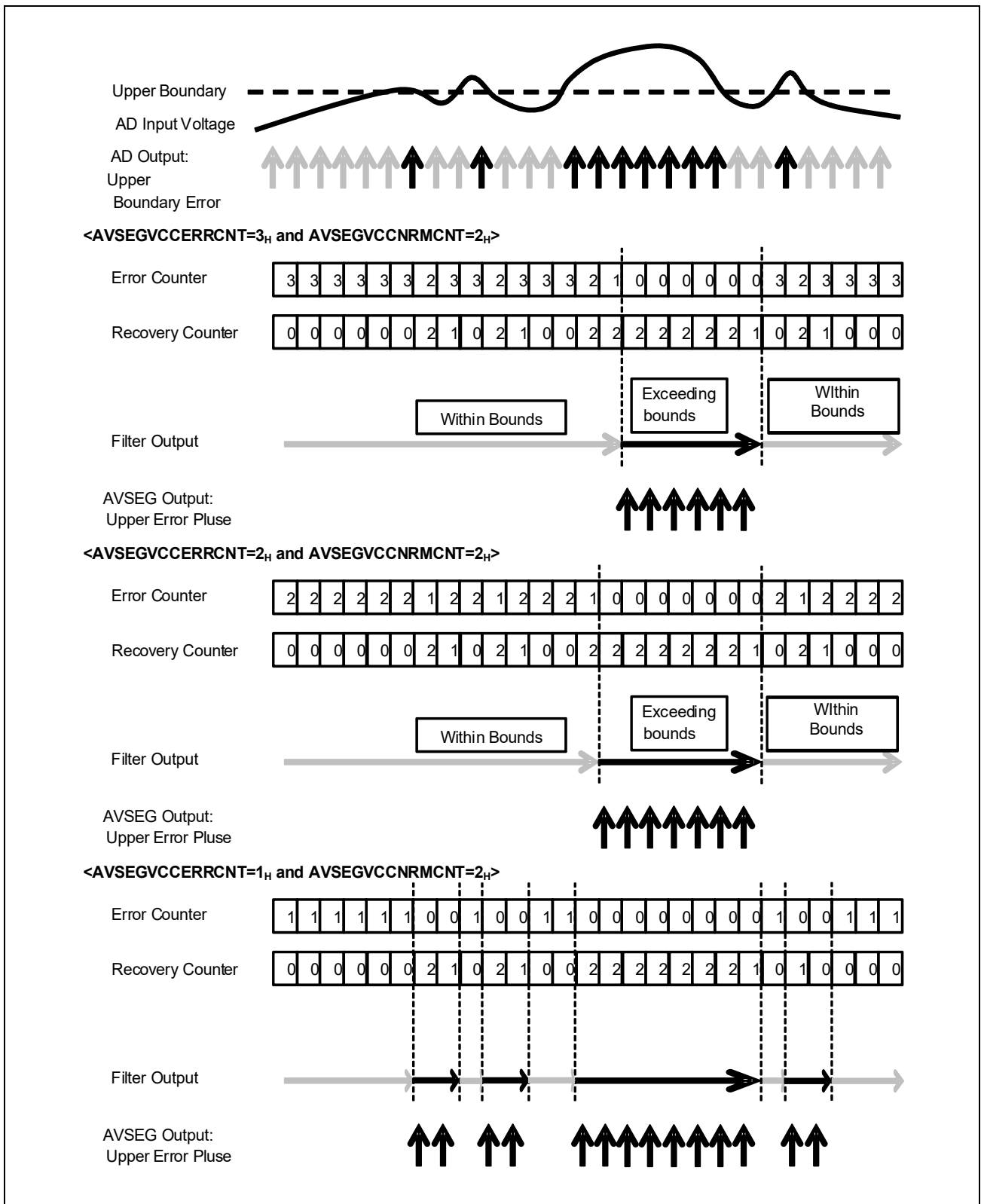


Figure 34.53 Noise Count (2)

Method 3:

AVSEGVCCNTR.AVSEGVCCERRCNT = 1_H to 3_H and AVSEGVCCNTR.AVSEGVCCNRMCMNT = 3_H.

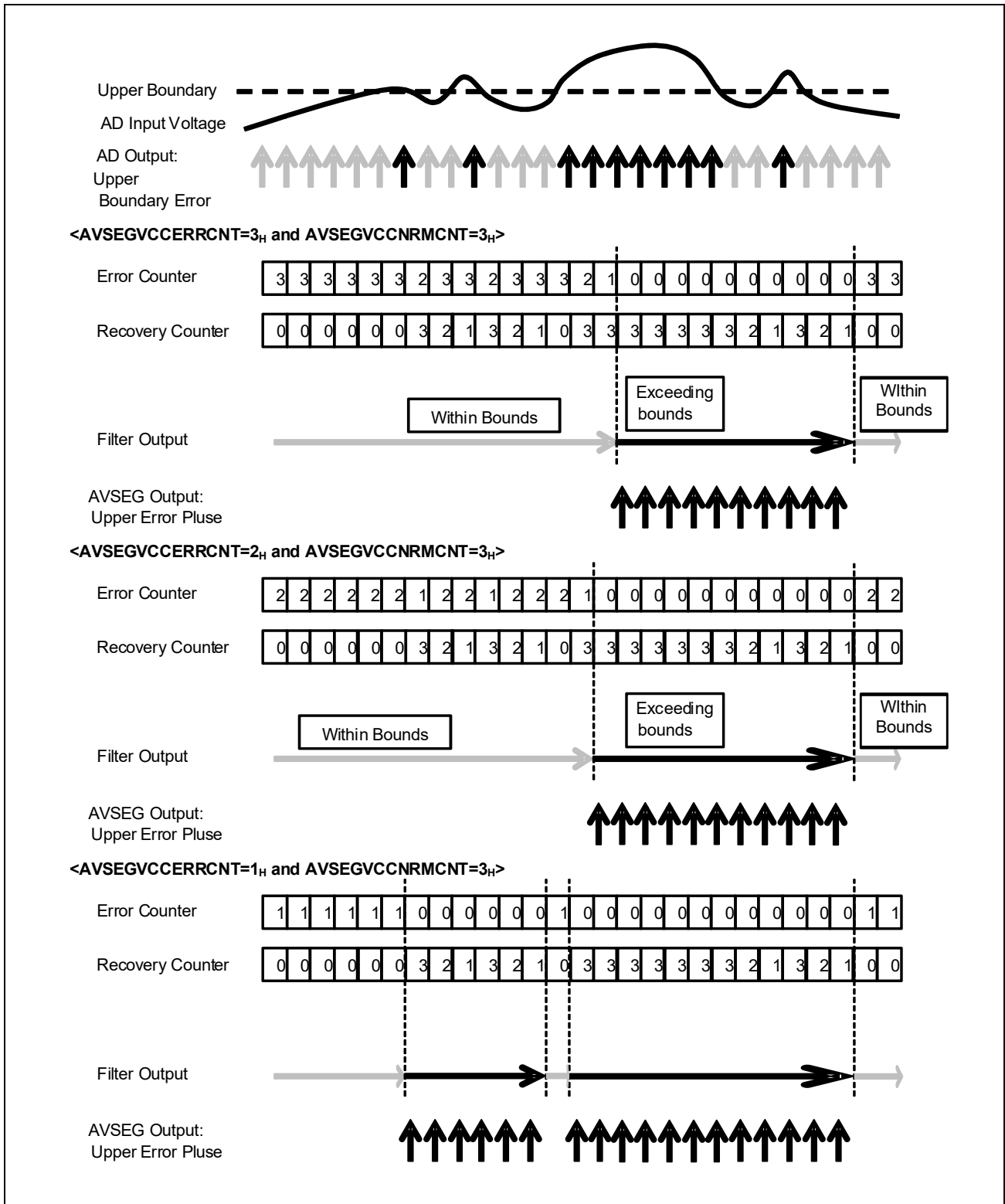


Figure 34.54 Noise Count (3)

34.9.3.2 Upper/Lower Error Pulse Output

The AVSEG upper/lower error pulse output signals are shown below. All output signals are generated by signals that have passed through the filter function to reduce noise.

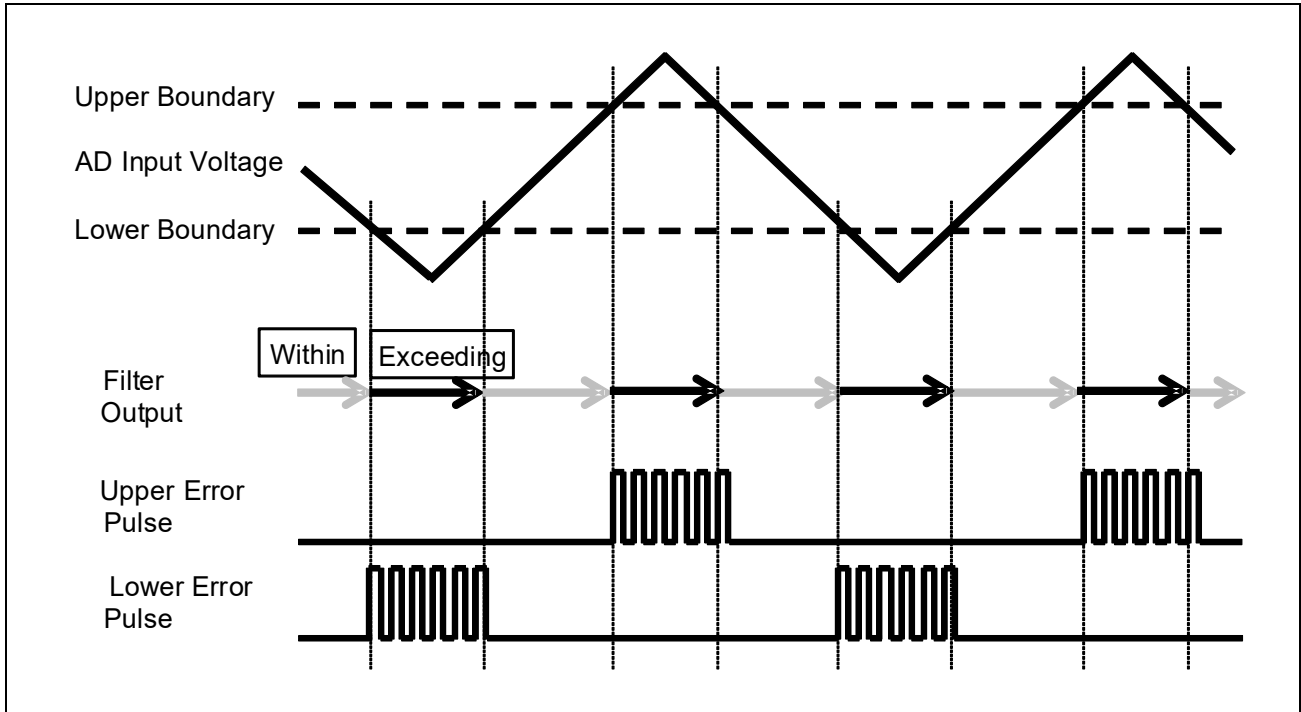


Figure 34.55 Upper/Lower Error Pulse Output

34.9.4 Use Method

(1) Startup Procedure Example

(a) ADC Settings (For details, refer to Section 34, Analog to Digital Converter (ADCH))

Set the virtual channel registers and virtual channel upper/lower limit table registers so that comparisons are made between the limit registers and the input voltage during conversion.

(b) ECM Settings

Refer to **Section 39, Error Control Module (ECM)**.

(c) AVSEG Settings

Set the Input Channel Selector Register of respective voltage (AVSEGVCCCHSCR, AVSEGEVCCCHSCR and AVSEGVDDCHSCR).

Set the Filter Counter Control Register of respective voltage (AVSEGVCCCNTCR, AVSEGEVCCCNTCR and AVSEGVDDCNTCR).

(d) Begin AD Conversion

Begin AD conversion after setting up AVSEG. If the AVSEG Filter Counter Control Register is modified, the error counter and recovery counter will be set with the written values, and all outputs change will be masked.

(2) Stop Procedure Example

(a) End A/D Conversion

Stop the ADCH0.

(b) AVSEG Setting

Set the filter counter enable bit, AVSEGVCCCNTCR. AVSEGVCCENB, AVSEGEVCCCNTCR. AVSEGEVCCENB and AVSEGVDDCNTCR. AVSEGVDDENB to 0.

This will prevent any and all pulse signals from being output from AVSEG.

NOTE

The internal filter counters will be reset to the written values during the AVSEGVCCCNTCR, AVSEGEVCCCNTCR and AVSEGVDDCNTCR write procedure.

(3) Usage Notes

(a) The input of both upper and lower error pulses into the same module at the same time is prohibited.

(b) Setting AVSEGVCCNRM CNT, AVSEGVCCERRCNT, AVSEGEVCCNRM CNT, AVSEGEVCCERRCNT, AVSEGVDDNRM CNT and AVSEGVDDERRCNT to 0_H is prohibited.

34.10 ADC Boundary Flag Generator (ABFG)

34.10.1 Overview

34.10.1.1 Function Overview

ABFG has the following functions.

- **Boundary Flag Control**
This function generates a boundary flag by using virtual channel conversion end pulse signals with their respective upper-limit/lower-limit error pulse signals provided by the SAR-ADC, Delta-Sigma ADC and Cyclic ADC converters.
- **Boundary Pulse Control**
This function generates an interrupt request and/or DMAC request that indicates the current state of the boundary flag.
- **Noise Filter**
The boundary flag, and boundary pulse are generated by signals passed through a filter to reduce noise.

CAUTION

The input of both upper and lower pulses into the same module at the same time is prohibited.

34.10.1.2 Block Diagram

Figure 34.56 illustrates the ABFG block diagram.

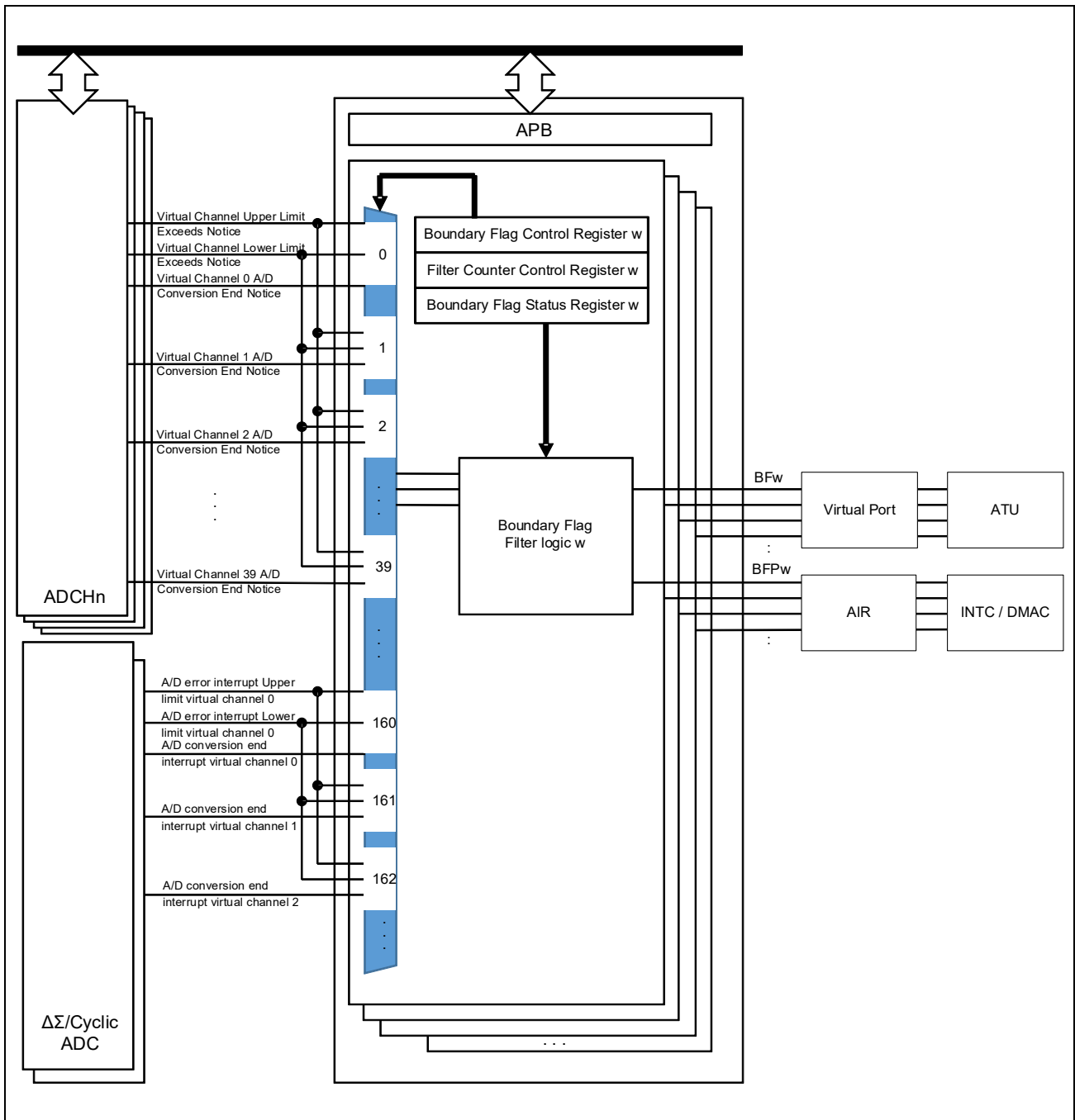


Figure 34.56 ABFG Block Diagram

Figure 34.57 illustrates the ABFG block diagram.

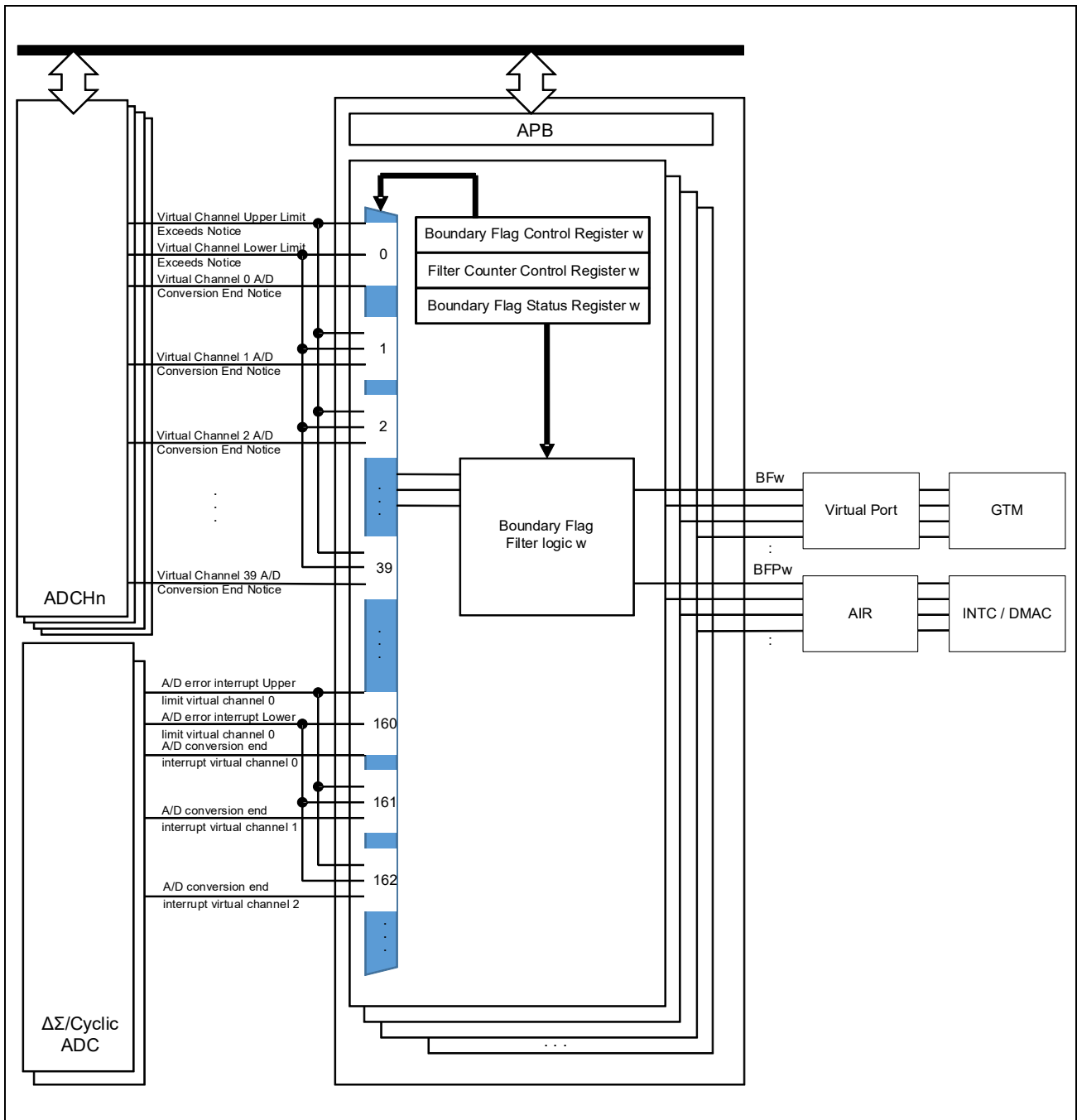


Figure 34.57 ABFG Block Diagram

34.10.2 Registers

34.10.2.1 List of Registers

Table 34.120 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
ABFG	Boundary Flag Control Register w	ABFGBFGCRw	<ABFG_base> + 10 _H × w + 00 _H	32	—
ABFG	Filter Counter Control Register w	ABFGCNTCRw	<ABFG_base> + 10 _H × w + 04 _H	32	—
ABFG	Boundary Flag Status Register w	ABFGBFSRw	<ABFG_base> + 10 _H × w + 08 _H	32	—

34.10.2.2 ABFG Specific Registers

(1) ABFGwBFGCRw —Boundary Flag Control Register w

This register sets selection of the input channel, toggle condition of the boundary flag and output condition of the boundary flag pulse.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ABFGwBTGC	—	—	—	—	—	—	—	ABFGwBPGC	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ABFGwCHS							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.121 ABFGwBFGCRw Register Contents

Bit Position	Bit Name	Function
31 to 26	—	Reserved These bits are always read as 0. The write value should always be 0.
25, 24	ABFGwBTGC	Boundary Flag Toggle Control These bits control the boundary flag toggle method. 00 _B : Flag is set to High when violating the upper boundary, and is set to Low when violating the lower boundary. 01 _B : Flag is set to Low when violating the upper boundary, and is set to High when violating the lower boundary. 10 _B : Flag is set to Low when within the upper and lower boundaries, and is set to High when violating either boundary. 11 _B : Flag is set to High when within the upper and lower boundaries, and is set to Low when violating either boundary.
23 to 18	—	Reserved These bits are always read as 0. The write value should always be 0.
17 to 16	ABFGwBPGC	Boundary Flag Pulse Generation Control These bits control the method used in generating the boundary flag pulse. 00 _B : No pulse is generated. 01 _B : A boundary flag pulse is generated on the rising edge of the boundary flag. 10 _B : A boundary flag pulse is generated on the falling edge of the boundary flag. 11 _B : A boundary flag pulse is generated on both rising and falling edges of the boundary flag.
15 to 8	—	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	ABFGwCHS	Input Channel Selector These bits select the channel used in generating the boundary flag. Refer to Table 34.122 to Table 34.129 .

Table 34.122 Input Channel Select (1)

ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel
0	ADCH0.VC0	10	ADCH0.VC10	20	ADCH0.VC20	30	ADCH0.VC30
1	ADCH0.VC1	11	ADCH0.VC11	21	ADCH0.VC21	31	ADCH0.VC31
2	ADCH0.VC2	12	ADCH0.VC12	22	ADCH0.VC22	32	ADCH0.VC32
3	ADCH0.VC3	13	ADCH0.VC13	23	ADCH0.VC23	33	ADCH0.VC33
4	ADCH0.VC4	14	ADCH0.VC14	24	ADCH0.VC24	34	ADCH0.VC34
5	ADCH0.VC5	15	ADCH0.VC15	25	ADCH0.VC25	35	ADCH0.VC35
6	ADCH0.VC6	16	ADCH0.VC16	26	ADCH0.VC26	36	ADCH0.VC36
7	ADCH0.VC7	17	ADCH0.VC17	27	ADCH0.VC27	37	ADCH0.VC37
8	ADCH0.VC8	18	ADCH0.VC18	28	ADCH0.VC28	38	ADCH0.VC38
9	ADCH0.VC9	19	ADCH0.VC19	29	ADCH0.VC29	39	ADCH0.VC39

Table 34.123 Input Channel Select (2)

ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel
40	ADCH2.VC0	50	ADCH2.VC10	60	ADCH2.VC20	70	ADCH2.VC30
41	ADCH2.VC1	51	ADCH2.VC11	61	ADCH2.VC21	71	ADCH2.VC31
42	ADCH2.VC2	52	ADCH2.VC12	62	ADCH2.VC22	72	ADCH2.VC32
43	ADCH2.VC3	53	ADCH2.VC13	63	ADCH2.VC23	73	ADCH2.VC33
44	ADCH2.VC4	54	ADCH2.VC14	64	ADCH2.VC24	74	ADCH2.VC34
45	ADCH2.VC5	55	ADCH2.VC15	65	ADCH2.VC25	75	ADCH2.VC35
46	ADCH2.VC6	56	ADCH2.VC16	66	ADCH2.VC26	76	ADCH2.VC36
47	ADCH2.VC7	57	ADCH2.VC17	67	ADCH2.VC27	77	ADCH2.VC37
48	ADCH2.VC8	58	ADCH2.VC18	68	ADCH2.VC28	78	ADCH2.VC38
49	ADCH2.VC9	59	ADCH2.VC19	69	ADCH2.VC29	79	ADCH2.VC39

Table 34.124 Input Channel Select (3)

ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel
80	ADCH1.VC0	90	ADCH1.VC10	100	ADCH1.VC20	110	ADCH1.VC30
81	ADCH1.VC1	91	ADCH1.VC11	101	ADCH1.VC21	111	ADCH1.VC31
82	ADCH1.VC2	92	ADCH1.VC12	102	ADCH1.VC22	112	ADCH1.VC32
83	ADCH1.VC3	93	ADCH1.VC13	103	ADCH1.VC23	113	ADCH1.VC33
84	ADCH1.VC4	94	ADCH1.VC14	104	ADCH1.VC24	114	ADCH1.VC34
85	ADCH1.VC5	95	ADCH1.VC15	105	ADCH1.VC25	115	ADCH1.VC35
86	ADCH1.VC6	96	ADCH1.VC16	106	ADCH1.VC26	116	ADCH1.VC36
87	ADCH1.VC7	97	ADCH1.VC17	107	ADCH1.VC27	117	ADCH1.VC37

Table 34.125 Input Channel Select (4)

ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel
88	ADCH1.VC8	98	ADCH1.VC18	108	ADCH1.VC28	118	ADCH1.VC38
89	ADCH1.VC9	99	ADCH1.VC19	109	ADCH1.VC29	119	ADCH1.VC39
120	ADCH3.VC0	130	ADCH3.VC10	140	ADCH3.VC20	150	ADCH3.VC30
121	ADCH3.VC1	131	ADCH3.VC11	141	ADCH3.VC21	151	ADCH3.VC31
122	ADCH3.VC2	132	ADCH3.VC12	142	ADCH3.VC22	152	ADCH3.VC32
123	ADCH3.VC3	133	ADCH3.VC13	143	ADCH3.VC23	153	ADCH3.VC33
124	ADCH3.VC4	134	ADCH3.VC14	144	ADCH3.VC24	154	ADCH3.VC34
125	ADCH3.VC5	135	ADCH3.VC15	145	ADCH3.VC25	155	ADCH3.VC35

Table 34.126 Input Channel Select (5)

ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel
126	ADCH3.VC6	136	ADCH3.VC16	146	ADCH3.VC26	156	ADCH3.VC36
127	ADCH3.VC7	137	ADCH3.VC17	147	ADCH3.VC27	157	ADCH3.VC37
128	ADCH3.VC8	138	ADCH3.VC18	148	ADCH3.VC28	158	ADCH3.VC38
129	ADCH3.VC9	139	ADCH3.VC19	149	ADCH3.VC29	159	ADCH3.VC39
160	DSADC00.VC0	170	DSADC10.VC2	180	DSADC13.VC2	190	—
161	DSADC00.VC1	171	DSADC10.VC3	181	DSADC13.VC3	191	—
162	DSADC00.VC2	172	DSADC20.VC0	182	DSADC11.VC0	192	—
163	DSADC00.VC3	173	DSADC20.VC1	183	DSADC11.VC1	193	—

Table 34.127 Input Channel Select (6)

ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel
164	DSADC00.VC4	174	DSADC12.VC0	184	DSADC11.VC2	194	—
165	DSADC00.VC5	175	DSADC12.VC1	185	DSADC11.VC3	195	—
166	DSADC00.VC6	176	DSADC12.VC2	186	—	196	—
167	DSADC00.VC7	177	DSADC12.VC3	187	—	197	—
168	DSADC10.VC0	178	DSADC13.VC0	188	—	198	CADC00.VC0
169	DSADC10.VC1	179	DSADC13.VC1	189	—	199	CADC00.VC1

Table 34.128 Input Channel Select (7)

ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel
200	CADC00.VC2	210	—	220	—	230	—
201	CADC00.VC3	211	—	221	—	231	—
202	CADC00.VC4	212	—	222	—	232	—
203	CADC00.VC5	213	—	223	—	233	—
204	CADC00.VC6	214	—	224	—	234	—
205	CADC00.VC7	215	—	225	—	235	—
206	—	216	—	226	—	236	—
207	—	217	—	227	—	237	—
208	—	218	—	228	—	238	—
209	—	219	—	229	—	239	—

Table 34.129 Input Channel Select (8)

ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel	ABFGwC HS	Select virtual channel
240	—	244	—	248	—	252	—
241	—	245	—	249	—	253	—
242	—	246	—	250	—	254	—
243	—	247	—	251	—	255	—

(2) ABFGNTRw — Filter Counter Control Register w

This register controls the filter counter. The noise filtering interval is controlled by using an error counter and a recovery counter. The internal down counter is reset with the written values when this register is modified.

Value after reset: 0000 0101_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	ABFGwENB	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	ABFGwNRCNT				—	—	—	—	ABFGwERRCNT			
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.130 ABFGNTRw Register Contents

Bit Position	Bit Name	Function
31 to 29	—	Reserved These bits are always read as 0. The write value should always be 0.
28	ABFGwENB	Filter Enable This bit enable boundary flag w functions. 0: Disabled 1: Enables the filter When set to disabled, all boundary flag w functions are disabled.
27 to 12	—	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	ABFGwNRCNT	Recovery Counter Settings These bits control the number of counts until the signal is considered recovered. Once the signal is considered out of bounds, it will not be recovered until the set number of consecutive non-error pulses are detected. The counter value is reset when one of the following values is written to ABFGwNRCNT. 1 _H : Recovery after the signal is within the boundaries for 1 count. : F _H : Recovery after the signal is within the boundaries for 15 consecutive counts.
7 to 4	—	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	ABFGwERRCNT	Error Counter Control These bits control the number of counts until the signal is considered out of bounds. The signal is considered out of bounds once the set number of consecutive error pulses are detected. The counter value is reset when one of the following values is written to ABFGwERRCNT. 1 _H : Out of bounds when the signal violates the boundaries for 1 count. : F _H : Out of bounds when the signal violates the boundaries for 15 consecutive counts.

CAUTION

Setting ABFGwNRCNT and ABFGwERRCNT to 0_H is prohibited.

(3) ABFGBFSRw — Boundary Flag Status Register w

This register shows the status of the boundary flag.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ABFGw BFS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 34.131 ABFGBFSRw Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved These bits are always read as 0. The write value should always be 0.
0	ABFGwBFS	Boundary Flag Status This bit shows the status of the boundary flag. It is possible to set the boundary flag directly by writing to this register.

CAUTION

Writing to this register while ABFGCNTCRw.ABFGwENB is 1 is prohibited.

34.10.3 Operation

34.10.3.1 Noise Count Method

The boundary flag noise count methods are shown below. Noise is filtered and reduced by counting the number of times the ADC boundary values are exceeded using the input error pulses.

CAUTION

The input of both upper and lower pulses into the same module at the same time is prohibited.

Example for the upper boundary error.

Method 1:

$ABFGNTRw.ABFGwERRCNT = 1_H$ to 3_H and $ABFGNTRw.ABFGwNRM CNT = 1_H$.

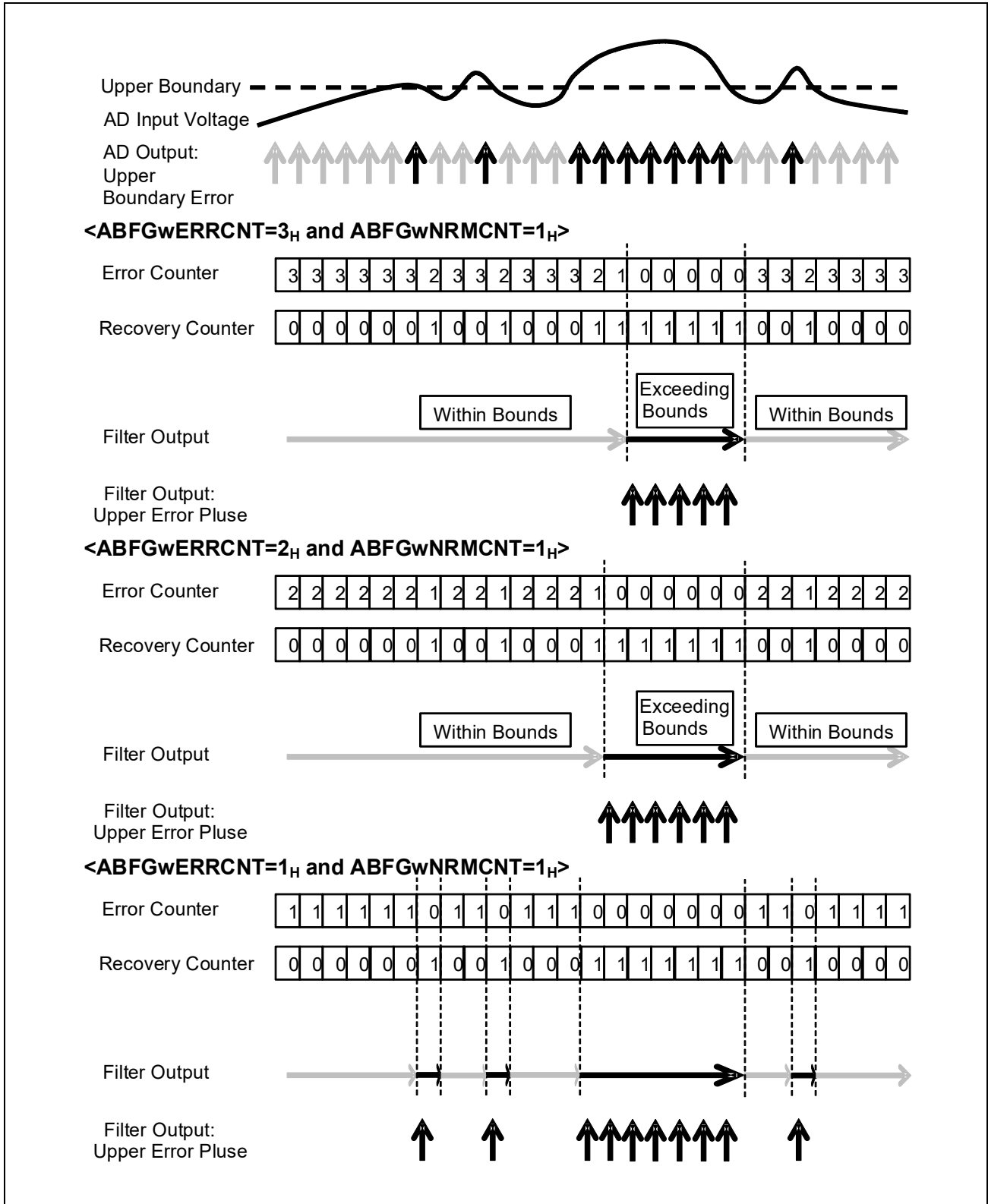


Figure 34.58 Noise Count (1)

Method 2:

$$ABFGCNTCRw.ABFGwERRCNT = 1_H \text{ to } 3_H \text{ and } ABFGCNTCRw.ABFGwNRCMNT = 2_H.$$

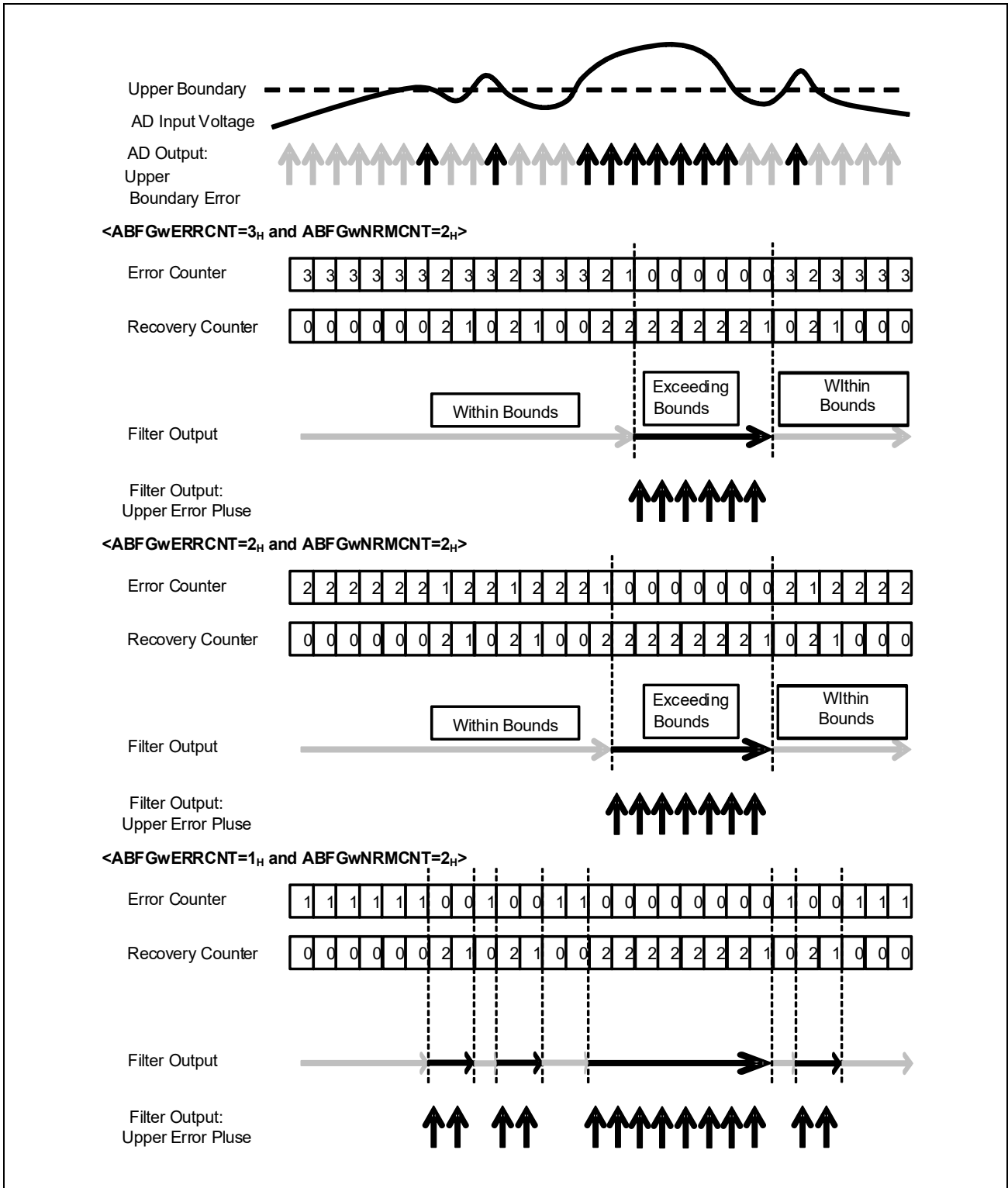


Figure 34.59 Noise Count (2)

Method 3:

$$ABFGNTRw.ABFGwERRCNT = 1_H \text{ to } 3_H \text{ and } ABFGNTRw.ABFGwNRCNT = 3_H.$$

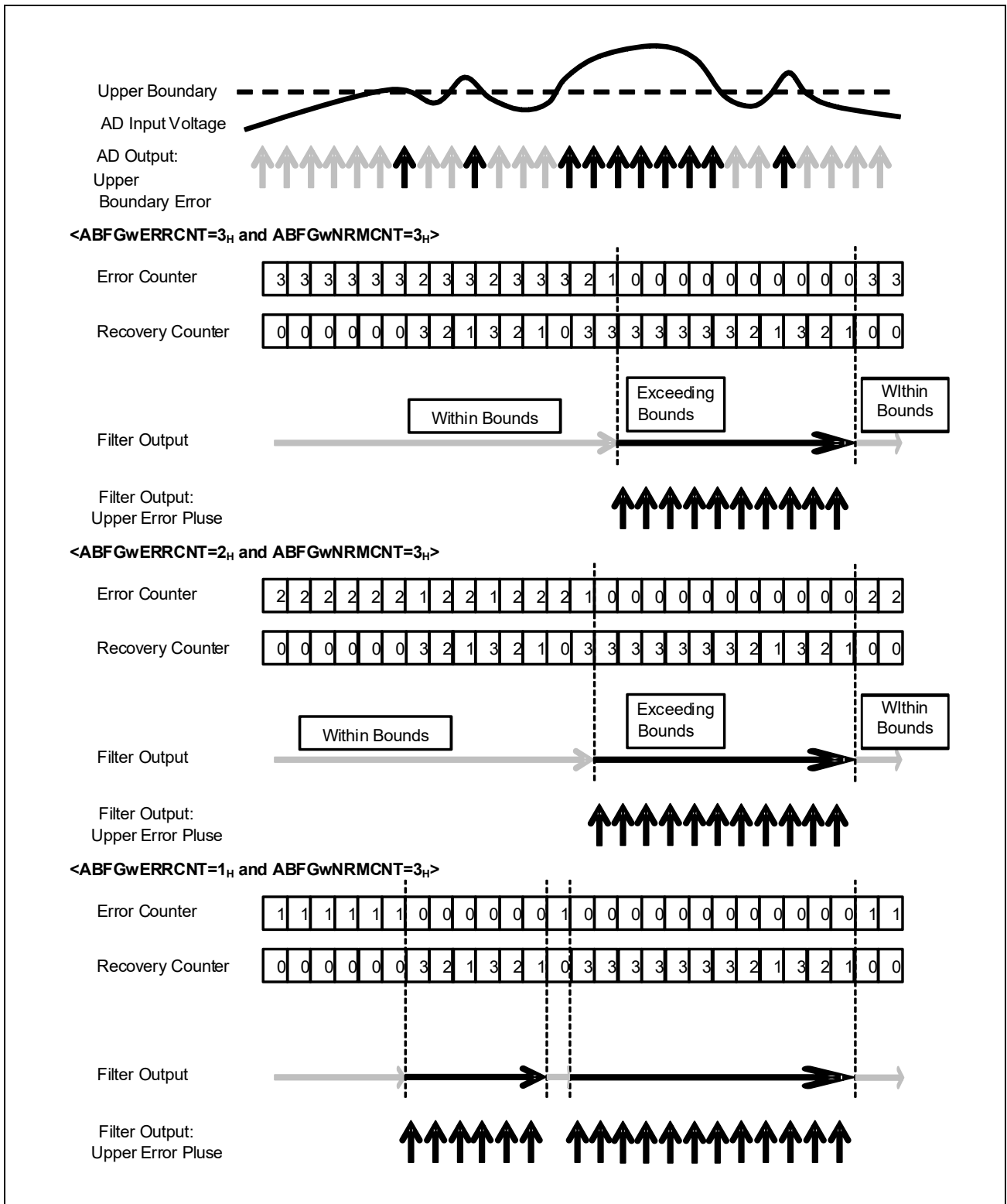


Figure 34.60 Noise Count (3)

34.10.3.2 Boundary Flag Output, Boundary Flag Pulse Output

The ABFG boundary flag output, boundary flag pulse output signals are shown below. All output signals are generated by signals that have passed through the filter function to reduce noise. The boundary flag can be generated using 4 different methods. The boundary flag pulse is generated by detecting the rising and/or falling edges of the boundary flag with 4 modes of output.

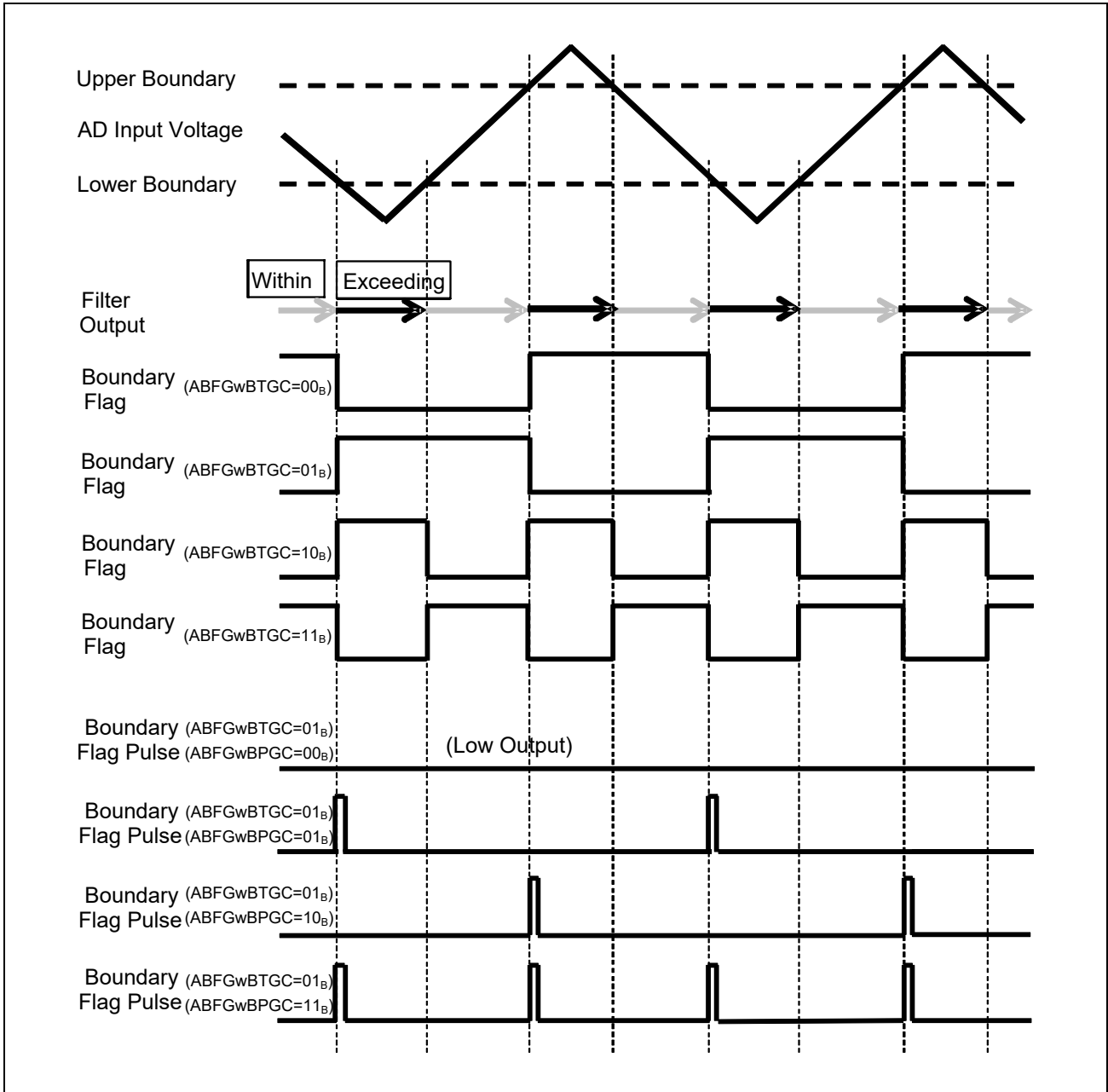


Figure 34.61 Boundary Flag Output, Boundary Flag Pulse Output

34.10.4 Use Method

(1) Starting Procedure Example

- (a) **ADC Settings (For details, refer to Section 34, Analog to Digital Converter (ADCH), Section 35, Delta-Sigma Analog to Digital Converter (DSADC) and Section 36, Cyclic Analog to Digital Converter (CADC))**

Set the virtual channel registers and virtual channel upper/lower limit table registers so that comparisons are made between the limit registers and the input voltage during conversion.

(b) ATU, GTM, INTC, DMAC Settings

Refer to **Section 30, ATU-V, Section 31, Generic Timer Module (GTM), Section 6, Interrupts, Section 7, sDMA Controller (sDMAC), Section 8, DTS Controller.**

(c) ABFG Settings

Set the boundary flag control register with the input channel, pulse generation and toggle methods (ABFGBFGCRw).

Set the boundary flag status (ABFGBFSRw) (optional).

Set the filter counter value (ABFGNCTCRw).

(d) Begin AD Conversion

Begin AD conversion after setting up ABFG. If the ABFG Filter Counter Control Register is modified, the error counter and recovery counter will be set with the written values, and all outputs change will be masked.

(2) Stop Procedure Example

(a) End A/D Conversion

Stop the ADCH, DSADC and CADC.

For details, refer to **Section 34, Analog to Digital Converter (ADCH), Section 35, Delta-Sigma Analog to Digital Converter (DSADC) and Section 36, Cyclic Analog to Digital Converter (CADC).**

(b) ABFG Setting

Set the filter counter enable bit, ABFGNCTCRw.ABFGwENB to 0.

This will prevent any and all pulse signals from being output from ABFG, and the boundary flag signal will hold its level until ABFGBFSRw register is changed or ABFG is turned back on.

NOTE

The internal filter counters will be reset to the written values during the ABFGNCTCRw write procedure.

(3) Usage Notes

- (a) The input of both upper and lower error pulses into the same module at the same time is prohibited.**
- (b) Setting ABFGwNRCNT and ABFGwERRCNT to 0_H is prohibited.**
- (c) Writing to ABFGBFSRw while ABFGCNTCRw.ENB is 1 is prohibited.**

34.11 ADC Interrupt Router (AIR)

The AIR is a function that selects signals from all A/D related resources, and outputs interrupt requests and DMA requests.

34.11.1 Overview

34.11.1.1 Function Overview

- AIR selects interrupt requests from resource 0 (ADCH, DSADC and CADC) and resource 1 (ABFG).
- AIR selects DMA requests from resource 0 (ADCH, DSADC and CADC) and resource 1 (ABFG).

34.11.1.2 Block Diagram

Figure 34.62 illustrates the AIR block diagram.

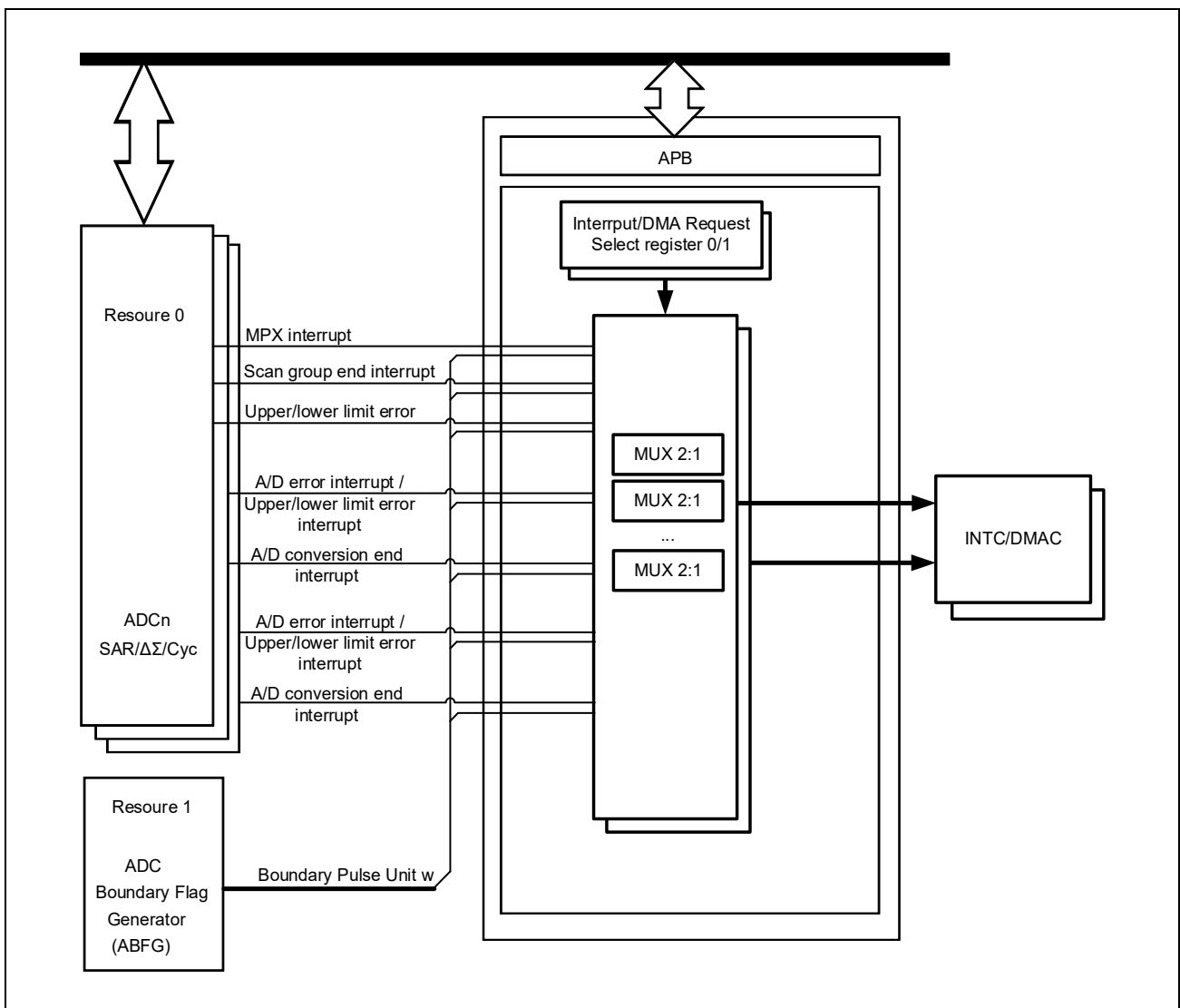


Figure 34.62 AIR Block Diagram

34.11.2 Registers

34.11.2.1 List of Registers

Table 34.132 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
AIR	Interrupt Request Select Register 0	AIR_ISELR0	<AIR_base> + 00 _H	8, 16, 32	—
AIR	Interrupt Request Select Register 1	AIR_ISELR1	<AIR_base> + 04 _H	8, 16, 32	—
AIR	DMA Request Select Register 0	AIR_DSELR0	<AIR_base> + 08 _H	8, 16, 32	—
AIR	DMA Request Select Register 1	AIR_DSELR1	<AIR_base> + 0C _H	8, 16, 32	—

34.11.2.2 AIR Specific Registers

(1) AIR_ISELR0 — Interrupt Request Select Register 0

This register selects interrupt request resources.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ISEL[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ISEL[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.133 AIR_ISELR0 Register Contents

Bit Position	Bit Name	Function
31 to 0	ISEL[31:0]	Interrupt Request Select These bits select from resource 0 and resource 1. For resource 0 and resource 1, refer to Table 34.135 to Table 34.139 .

(2) AIR_ISELR1 — Interrupt Request Select Register 1

This register selects interrupt request resources.

Value after reset: FF80 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ISEL[63:48]															
Value after reset	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ISEL[47:32]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.134 AIR_ISELR1 Register Contents

Bit Position	Bit Name	Function
31 to 0	ISEL[63:32]	Interrupt Request Select These bits select from resource 0 and resource 1. For resource 0 and resource 1, refer to Table 34.135 to Table 34.139 .

Table 34.135 Interrupt Request Select (1)

Interrupt Signal	Outline			
	Register Bit Name	Resource 0	Resource 1	Initial resource number
INTAIRINTREQ0	ISEL[0]	ADMPXI0	—	0
INTAIRINTREQ1	ISEL[1]	ADMPXI2	—	0
INTAIRINTREQ2	ISEL[2]	ADMPXI1	BFP18	0
INTAIRINTREQ3	ISEL[3]	ADMPXI3	BFP19	0
INTAIRINTREQ4	ISEL[4]	ADI00	—	0
INTAIRINTREQ5	ISEL[5]	ADI01	—	0
INTAIRINTREQ6	ISEL[6]	ADI02	—	0
INTAIRINTREQ7	ISEL[7]	ADI03	—	0
INTAIRINTREQ8	ISEL[8]	ADI04	—	0
INTAIRINTREQ9	ISEL[9]	ADI20	—	0
INTAIRINTREQ10	ISEL[10]	ADI21	—	0
INTAIRINTREQ11	ISEL[11]	ADI22	—	0
INTAIRINTREQ12	ISEL[12]	ADI23	—	0
INTAIRINTREQ13	ISEL[13]	ADI24	—	0
INTAIRINTREQ14	ISEL[14]	ADI10	BFP20	0
INTAIRINTREQ15	ISEL[15]	ADI11	BFP21	0

Table 34.136 Interrupt Request Select (2)

Interrupt Signal	Outline			
	Register Bit Name	Resource 0	Resource 1	Initial resource number
INTAIRINTREQ16	ISEL[16]	ADI12	BFP22	0
INTAIRINTREQ17	ISEL[17]	ADI13	BFP23	0
INTAIRINTREQ18	ISEL[18]	ADI14	BFP24	0
INTAIRINTREQ19	ISEL[19]	ADI30	BFP25	0
INTAIRINTREQ20	ISEL[20]	ADI31	BFP26	0
INTAIRINTREQ21	ISEL[21]	ADI32	BFP27	0
INTAIRINTREQ22	ISEL[22]	ADI33	BFP28	0
INTAIRINTREQ23	ISEL[23]	ADI34	BFP29	0
INTAIRINTREQ24	ISEL[24]	ADE0 (merged INT_UL0)	—	0
INTAIRINTREQ25	ISEL[25]	ADE2 (merged INT_UL2)	—	0

Table 34.137 Interrupt Request Select (3)

Interrupt Signal	Outline			
	Register Bit Name	Resource 0	Resource 1	Initial resource number
INTAIRINTREQ26	ISEL[26]	ADE1 (merged INT_UL1)	BFP13	0
INTAIRINTREQ27	ISEL[27]	ADE3 (merged INT_UL3)	BFP14	0
INTAIRINTREQ28	ISEL[28]	DSADE00	—	0
INTAIRINTREQ29	ISEL[29]	DSADE10	—	0
INTAIRINTREQ30	ISEL[30]	DSADE20	—	0
INTAIRINTREQ31	ISEL[31]	DSADE12	BFP15	0
INTAIRINTREQ32	ISEL[32]	DSADE13	BFP16	0
INTAIRINTREQ33	ISEL[33]	DSADE11	BFP17	0
INTAIRINTREQ34	ISEL[34]	—	BFP9	0
INTAIRINTREQ35	ISEL[35]	—	BFP10	0
INTAIRINTREQ36	ISEL[36]	—	BFP11	0
INTAIRINTREQ37	ISEL[37]	—	BFP12	0
INTAIRINTREQ38	ISEL[38]	CADE00	—	0
INTAIRINTREQ39	ISEL[39]	ASIO	BFP30	0

Table 34.138 Interrupt Request Select (4)

Interrupt Signal	Outline			
	Register Bit Name	Resource 0	Resource 1	Initial resource number
INTAIRINTREQ40	ISEL[40]	ASI1	BFP31	0
INTAIRINTREQ41	ISEL[41]	ASI2	BFP32	0
INTAIRINTREQ42	ISEL[42]	ASI3	BFP33	0
INTAIRINTREQ43	ISEL[43]	ASI4	BFP34	0
INTAIRINTREQ44	ISEL[44]	ASI5	BFP35	0
INTAIRINTREQ45	ISEL[45]	ASI6	BFP36	0

Table 34.139 Interrupt Request Select (5)

Interrupt Signal	Outline			
	Register Bit Name	Resource 0	Resource 1	Initial resource number
INTAIRINTREQ46	ISEL[46]	ASI7	BFP37	0
INTAIRINTREQ47	ISEL[47]	ASI8	BFP38	0
INTAIRINTREQ48	ISEL[48]	ASI9	BFP39	0
INTAIRINTREQ49	ISEL[49]	ASI10	—	0
INTAIRINTREQ50	ISEL[50]	ASI11	—	0
INTAIRINTREQ51	ISEL[51]	ASI12	—	0
INTAIRINTREQ52	ISEL[52]	ASI13	—	0
INTAIRINTREQ53	ISEL[53]	ASI14	—	0
INTAIRINTREQ54	ISEL[54]	ASI15	—	0
INTAIRINTREQ55	ISEL[55]	—	BFP0	1
INTAIRINTREQ56	ISEL[56]	—	BFP1	1
INTAIRINTREQ57	ISEL[57]	—	BFP2	1
INTAIRINTREQ58	ISEL[58]	—	BFP3	1
INTAIRINTREQ59	ISEL[59]	—	BFP4	1
INTAIRINTREQ60	ISEL[60]	—	BFP5	1
INTAIRINTREQ61	ISEL[61]	—	BFP6	1
INTAIRINTREQ62	ISEL[62]	—	BFP7	1
INTAIRINTREQ63	ISEL[63]	—	BFP8	1

(3) AIR_DSELR0 — DMA Request Select Register 0

This register selects DMA request resources.

Value after reset: 0F00 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DSEL[31:16]															
Value after reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSEL[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.140 AIR_DSELR0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DSEL[31:0]	DMA Request Select These bits select from resource 0 and resource 1. For resource 0 and resource 1, refer to Table 34.142 to Table 34.146 .

(4) AIR_DSELR1 — DMA Request Select Register 1

This register selects DMA request resources.

Value after reset: FF80 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DSEL[63:48]															
Value after reset	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSEL[47:32]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.141 AIR_DSELR1 Register Contents

Bit Position	Bit Name	Function
31 to 0	DSEL[63:32]	DMA Request Select These bits select from resource 0 and resource 1. For resource 0 and resource 1, refer to Table 34.142 to Table 34.146 .

Table 34.142 DMA Request Select (1)

DMA Signal	Outline			
	Register Bit Name	Resource 0	Resource 1	Initial resource number
INTAIRDMAREQ0	DSEL[0]	ADMPXI0	—	0
INTAIRDMAREQ1	DSEL[1]	ADMPXI2	—	0
INTAIRDMAREQ2	DSEL[2]	ADMPXI1	BFP18	0
INTAIRDMAREQ3	DSEL[3]	ADMPXI3	BFP19	0
INTAIRDMAREQ4	DSEL[4]	ADI00	—	0
INTAIRDMAREQ5	DSEL[5]	ADI01	—	0
INTAIRDMAREQ6	DSEL[6]	ADI02	—	0
INTAIRDMAREQ7	DSEL[7]	ADI03	—	0
INTAIRDMAREQ8	DSEL[8]	ADI04	—	0
INTAIRDMAREQ9	DSEL[9]	ADI20	—	0
INTAIRDMAREQ10	DSEL[10]	ADI21	—	0
INTAIRDMAREQ11	DSEL[11]	ADI22	—	0
INTAIRDMAREQ12	DSEL[12]	ADI23	—	0
INTAIRDMAREQ13	DSEL[13]	ADI24	—	0
INTAIRDMAREQ14	DSEL[14]	ADI10	BFP20	0
INTAIRDMAREQ15	DSEL[15]	ADI11	BFP21	0
INTAIRDMAREQ16	DSEL[16]	ADI12	BFP22	0
INTAIRDMAREQ17	DSEL[17]	ADI13	BFP23	0
INTAIRDMAREQ18	DSEL[18]	ADI14	BFP24	0
INTAIRDMAREQ19	DSEL[19]	ADI30	BFP25	0

Table 34.143 DMA Request Select (2)

DMA Signal	Outline			
	Register Bit Name	Resource 0	Resource 1	Initial resource number
INTAIRDMAREQ20	DSEL[20]	ADI31	BFP26	0
INTAIRDMAREQ21	DSEL[21]	ADI32	BFP27	0
INTAIRDMAREQ22	DSEL[22]	ADI33	BFP28	0
INTAIRDMAREQ23	DSEL[23]	ADI34	BFP29	0
INTAIRDMAREQ24	DSEL[24]	—	BFP9	1
INTAIRDMAREQ25	DSEL[25]	—	BFP10	1
INTAIRDMAREQ26	DSEL[26]	—	BFP11	1
INTAIRDMAREQ27	DSEL[27]	—	BFP12	1
INTAIRDMAREQ28	DSEL[28]	DSADI00	—	0
INTAIRDMAREQ29	DSEL[29]	DSADI10	—	0
INTAIRDMAREQ30	DSEL[30]	DSADI20	—	0
INTAIRDMAREQ31	DSEL[31]	DSADI12	BFP17	0
INTAIRDMAREQ32	DSEL[32]	DSADI13	—	0

Table 34.144 DMA Request Select (3)

DMA Signal	Outline			
	Register Bit Name	Resource 0	Resource 1	Initial resource number
INTAIRDMAREQ33	DSEL[33]	DSADI11	—	0
INTAIRDMAREQ34	DSEL[34]	—	BFP13	0
INTAIRDMAREQ35	DSEL[35]	—	BFP14	0
INTAIRDMAREQ36	DSEL[36]	—	BFP15	0
INTAIRDMAREQ37	DSEL[37]	—	BFP16	0
INTAIRDMAREQ38	DSEL[38]	CADI00	—	0
INTAIRDMAREQ39	DSEL[39]	ASI0	BFP30	0
INTAIRDMAREQ40	DSEL[40]	ASI1	BFP31	0
INTAIRDMAREQ41	DSEL[41]	ASI2	BFP32	0
INTAIRDMAREQ42	DSEL[42]	ASI3	BFP33	0

Table 34.145 DMA Request Select (4)

DMA Signal	Outline			
	Register Bit Name	Resource 0	Resource 1	Initial resource number
INTAIRDMAREQ43	DSEL[43]	ASI4	BFP34	0
INTAIRDMAREQ44	DSEL[44]	ASI5	BFP35	0
INTAIRDMAREQ45	DSEL[45]	ASI6	BFP36	0
INTAIRDMAREQ46	DSEL[46]	ASI7	BFP37	0
INTAIRDMAREQ47	DSEL[47]	ASI8	BFP38	0
INTAIRDMAREQ48	DSEL[48]	ASI9	BFP39	0
INTAIRDMAREQ49	DSEL[49]	ASI10	—	0
INTAIRDMAREQ50	DSEL[50]	ASI11	—	0

Table 34.146 DMA Request Select (5)

DMA Signal	Outline			
	Register Bit Name	Resource 0	Resource 1	Initial resource number
INTAIRDMAREQ51	DSEL[51]	ASI12	—	0
INTAIRDMAREQ52	DSEL[52]	ASI13	—	0
INTAIRDMAREQ53	DSEL[53]	ASI14	—	0
INTAIRDMAREQ54	DSEL[54]	ASI15	—	0
INTAIRDMAREQ55	DSEL[55]	—	BFP0	1
INTAIRDMAREQ56	DSEL[56]	—	BFP1	1
INTAIRDMAREQ57	DSEL[57]	—	BFP2	1
INTAIRDMAREQ58	DSEL[58]	—	BFP3	1
INTAIRDMAREQ59	DSEL[59]	—	BFP4	1
INTAIRDMAREQ60	DSEL[60]	—	BFP5	1
INTAIRDMAREQ61	DSEL[61]	—	BFP6	1
INTAIRDMAREQ62	DSEL[62]	—	BFP7	1
INTAIRDMAREQ63	DSEL[63]	—	BFP8	1

Section 35 Delta-Sigma Analog to Digital Converter (DSADC)

This section contains a description of the Delta Sigma A/D Converter (DSADC).

The first part of this section describes all of the products, such as the number of units, and register base addresses. The remainder of the section describes the functions and registers of the DSADC.

35.1 Features

35.1.1 Number of Units

This microcontroller has following number of DSADC units.

Table 35.1 Number of Units (For RH850/E2x-FCC1, E2M)

Product Name	RH850/E2x-FCC1, E2M	
	292 Pins	373 Pins
Number of DSADC Common Units	1	1
Number of DSADC Channel Units	6	6
Name of DSADC Units	DSADC DSADCn (n = 00, 10, 11, 12, 13, 20)	

A DSADCn unit has the same number of physical channels as the number of A/D input pins. The numbers of channels on individual products are listed below.

Table 35.2 Unit Configurations and Physical Channels (For RH850/E2x-FCC1, E2M)

Unit Name (Number of Channels)	RH850/E2x-FCC1, E2M	
	292 Pins	373 Pins
DSADCn		
DSADC00	8	8
DSADC10	4	4
DSADC20	2	2
DSADC12	4	4
DSADC13	4	4
DSADC11	4	4

Table 35.3 Unit Configurations and Virtual Channels (For RH850/E2x-FCC1, E2M)

Unit Name (Number of Channels)	RH850/E2x-FCC1, E2M	
	292 Pins	373 Pins
DSADCn		
DSADC00	8	8
DSADC10	4	4
DSADC20	2	2
DSADC12	4	4
DSADC13	4	4
DSADC11	4	4

Table 35.4 Indices

Index	Description
n	Throughout this section, the individual DSADC units are identified by the index "n" (n = 00, 10, 20, 12, 13, 11); For example, DSADCnCCR indicates the channel control register.
x	Throughout this section, the individual analog input attributes at differential input are identified by the index "x" (x = P, N); for example, DSANn0x.
j	Number of the virtual channel

35.1.2 Register Base Addresses

The DSADC and DSADCn base addresses are listed in the following table.

The DSADC and DSADCn register addresses are given as offsets from the base address.

Table 35.5 Register Base Addresses

Base Address Name	Base Address	Bus Group
<DSADC_base>	FFF3 0000 _H	Peripheral Group 7
<DSADC00_base>	FFF3 1000 _H	Peripheral Group 7
<DSADC10_base>	FFF3 2000 _H	Peripheral Group 7
<DSADC20_base>	FFF3 3000 _H	Peripheral Group 7
<DSADC12_base>	FFF3 4000 _H	Peripheral Group 7
<DSADC13_base>	FFF5 0000 _H	Peripheral Group 8
<DSADC11_base>	FFF5 1000 _H	Peripheral Group 8

35.1.3 Clock Supply

The DSADC and DSADCn clock supplies are shown in the following table.

Table 35.6 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
DSADC	ADCLK	CLK_HSB
	Register access clock	CLK_LSB
DSADCn	ADCLK	CLK_HSB
	Register access clock	CLK_LSB

35.1.4 Interrupt Requests

The DSADCn interrupt requests are listed in the following table.

Table 35.7 Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMAC Trigger Number	DTS Trigger Number	Other Trigger Signals
DSADCn						
*1	DSADEn	A/D error interrupt Upper limit/lower limit error interrupt	*1	—	—	—
*1	DSADIn	A/D conversion end interrupt	—	*1	*1	—

Note 1. DSADEn and DSADIn are chosen by the AIR (ADC Interrupt Router), and those are forwarded to the INTC and DMAC.
For Interrupt Number and DMA Trigger Number, see **Section 34, Analog to Digital Converter (ADCH)**.

35.1.5 Reset Source

The DSADC and DSADCn reset sources are listed in the following table. The DSADC and DSADCn are initialized by the following reset sources.

Table 35.8 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
DSADC	All registers	✓	✓	✓	✓	—	—
DSADCn	All registers	✓	✓	✓	✓	—	—

35.1.6 External Input/Output Signals

The External Input/Output signals of the DSADC_n are listed below.

Table 35.9 DSADC External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
DSADC _n Common		
ADSVCC	Power supply pin for the analog part	ADSVCC
ADSVSS	Ground pin for the analog part	ADSVSS
ADSVREFH	Upper reference voltage pin for the analog part	ADSVREFH
ADSVREFL	Lower reference voltage pin for the analog part	ADSVREFL
ADSVCL	Delta sigma ADC external capacitor pin	ADSVCL

Table 35.10 DSADC_n (n = 00) External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
DSANn0P	Analog input pin (for single-ended/differential P)	DSANn0P
DSANn0N	Analog input pin (for single-ended/differential N)	DSANn0N
DSANn1P	Analog input pin (for single-ended/differential P)	DSANn1P
DSANn1N	Analog input pin (for single-ended/differential N)	DSANn1N
DSANn2P	Analog input pin (for single-ended/differential P)	DSANn2P
DSANn2N	Analog input pin (for single-ended/differential N)	DSANn2N
DSANn3P	Analog input pin (for single-ended/differential P)	DSANn3P
DSANn3N	Analog input pin (for single-ended/differential N)	DSANn3N
DSADTRG _n	External trigger pin	DSADTRG _n
DSADEND _n	A/D conversion timing monitor pin	DSADEND _n

Table 35.11 DSADC_n (n = 10, 11, 12, 13) External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
DSANn0P	Analog input pin (for single-ended/differential P)	DSANn0P
DSANn0N	Analog input pin (for single-ended/differential N)	DSANn0N
DSANn1P	Analog input pin (for single-ended/differential P)	DSANn1P
DSANn1N	Analog input pin (for single-ended/differential N)	DSANn1N
DSADTRG _n	External trigger pin	DSADTRG _n
DSADEND _n	A/D conversion timing monitor pin	DSADEND _n

Table 35.12 DSADC_n (n = 20) External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
DSANn0P	Analog input pin (for single-ended/differential P)	DSANn0P
DSANn0N	Analog input pin (for single-ended/differential N)	DSANn0N
DSADTRG _n	External trigger pin	DSADTRG _n
DSADEND _n	A/D conversion timing monitor pin	DSADEND _n

35.2 Overview

35.2.1 Function Overview

The following describes the features (provides a functional overview) of the DSADC.

- A/D conversion method: Delta sigma modulation method
- Configuration of analog input pins
 - 6 modules, 26 analog input pins in total
 - DSADC00: 8 single-ended inputs (4 differential inputs)
 - DSADC1y (y = 0 to 3): 4 single-ended inputs (2 differential inputs)
 - DSADC20: 2 single-ended inputs (1 differential input)

NOTES

1. One differential input can be used as two single-ended inputs.
2. All analog inputs are multiplexed with ADCH analog inputs.

- Input type: Single-ended input and differential input
 - Single-ended input or differential input can be selected for each DSADCn.
 - In single-ended mode:
 - The A/D conversion value format of unsigned 16-bit resolution is generated when ADSVREFL is chosen as the common mode voltage.
 - Range of the A/D conversion value: 0000_H to FFFF_H
 - The A/D conversion value format of signed 15-bit resolution (15-bit value preceded by the sign bit) is generated when ADSVREFH/2 is chosen as the common mode voltage.
 - Range of the A/D conversion value: C000_H to 3FFF_H (When gain is ×1), 8000_H to 7FFF_H (When gain is ×2, ×4, or ×8)
 - In differential mode:
 - The A/D conversion value format of signed 15-bit resolution (15-bit value preceded by the sign bit) is generated.
 - Range of the A/D conversion value: 8000_H to 7FFF_H
- Input gain function (PGA): ×1, ×2, ×4, or ×8 can be selected by each virtual channel using PGA (programmable gain amplifier) in each DSADCn.
- Sampling rate (Fs): 100 ksps, 200 ksps, 400ksps, 800ksps and 1600 ksps (Fos (Over sampling rate) is 8Msps)
- Pass band: 30 kHz, 60 kHz, 90 kHz, 100 kHz and 200 kHz
- Virtual channels
 - See **Section 35.1.1, Number of Units.**
- Calibration function
 - It is possible to achieve highly accurate A/D conversions by using the calibration function in spite of the use environment.
- Data register
 - A data register corresponding to each DSADCn is contained in the data supplementary information register j (DSADCnDIRj per virtual channel) or the unit data supplementary information register (DSADCnUDIR).
 - The latest data can be read from DSADCnUDIR, even if not reading virtual channel pointer.
- A/D conversion start trigger
 - Each DSADCn can start an A/D conversion by software, each timer trigger, or an external trigger (DSADTRGn).

- A/D conversion end trigger
Each DSADC_n can terminate A/D conversion by software, by each timer trigger, or by an external trigger (DSADTRG_n).
- Entry to the digital filter engine and the Generic Timer Module
A/D converted values can be entered directly into the Digital Filter Engine (DFE), or Generic Timer Module (GTM). Each DSADC_n can select from DFE0 and DFE1 as an entry target.
Whether to enable or disable DFE entry and the TAG to determine the channels for entry can be set for each virtual channel.
For details of the DFE, see **Section 37, Digital Filter Engine (DFE)**.
For details of the GTM, see **Section 31, Generic Timer Module (GTM)**.
- Timestamp function
The lapsed time from the result output is stored in the timestamp register at the rise time of the read-gate (DSADRGT_n) from the timer (ATU, GTM) as timestamp information. This function allows measurement of the time difference between the DSADRGT_n and input sampling time.
For details of the ATU, see **Section 30, ATU-V**.
For details of the GTM, see **Section 31, Generic Timer Module (GTM)**.
- DMA transfer based on A/D conversion end timing
Each time an A/D conversion ends, each DSADC_n outputs a DMA request on completion of the A/D conversion (DSADIn), which can generate DMA request sources (DSADIn) to the DMAC or activate the DMAC.
The output of the DSADIn is capable of beginning limited by the read-gate from a timer (ATU, GTM).
- Boundary flag output function
It is possible to output a boundary flag generating signal (DSADUEn, DSADLEn and DSADC_nVCI_j) to the ABFG (ADC Boundary Flag Generator).
For details of the ABFG, see **Section 34, Analog to Digital Converter (ADCH)**.
- AD error interrupt request and AD parity error notification
Each DSADC can generate an AD error interrupt request (DSADEn) to the INTC and an AD parity error notification (DSADPE_n) to the ECM (Error Control Module).
- Settable analog conversion voltage range
An analog conversion voltage range can be set using the ADSVREFH pin and the ADSVREFL pin. When performing an A/D conversion by using a single-ended input, either ADSVREFL or ADSVREFH/2 can be selected as the common voltage.
- A/D conversion monitor output
A/D conversion timing can be output to the A/D conversion monitor output pin (DSADEND_n).
- A wide range of safety functions
A variety of safety functions are provided, including AD self-diagnosis, pin-level self-diagnosis, wiring-break detection, self-diagnosis of wiring-break detection, normality check for analog selection, upper-limit/lower-limit check for data registers, data register parity check for data registers, overwrite check for data registers, and read and clear function for data registers.

35.2.2 Block Diagram

The block diagram of the DSADC is shown in **Figure 35.1**.

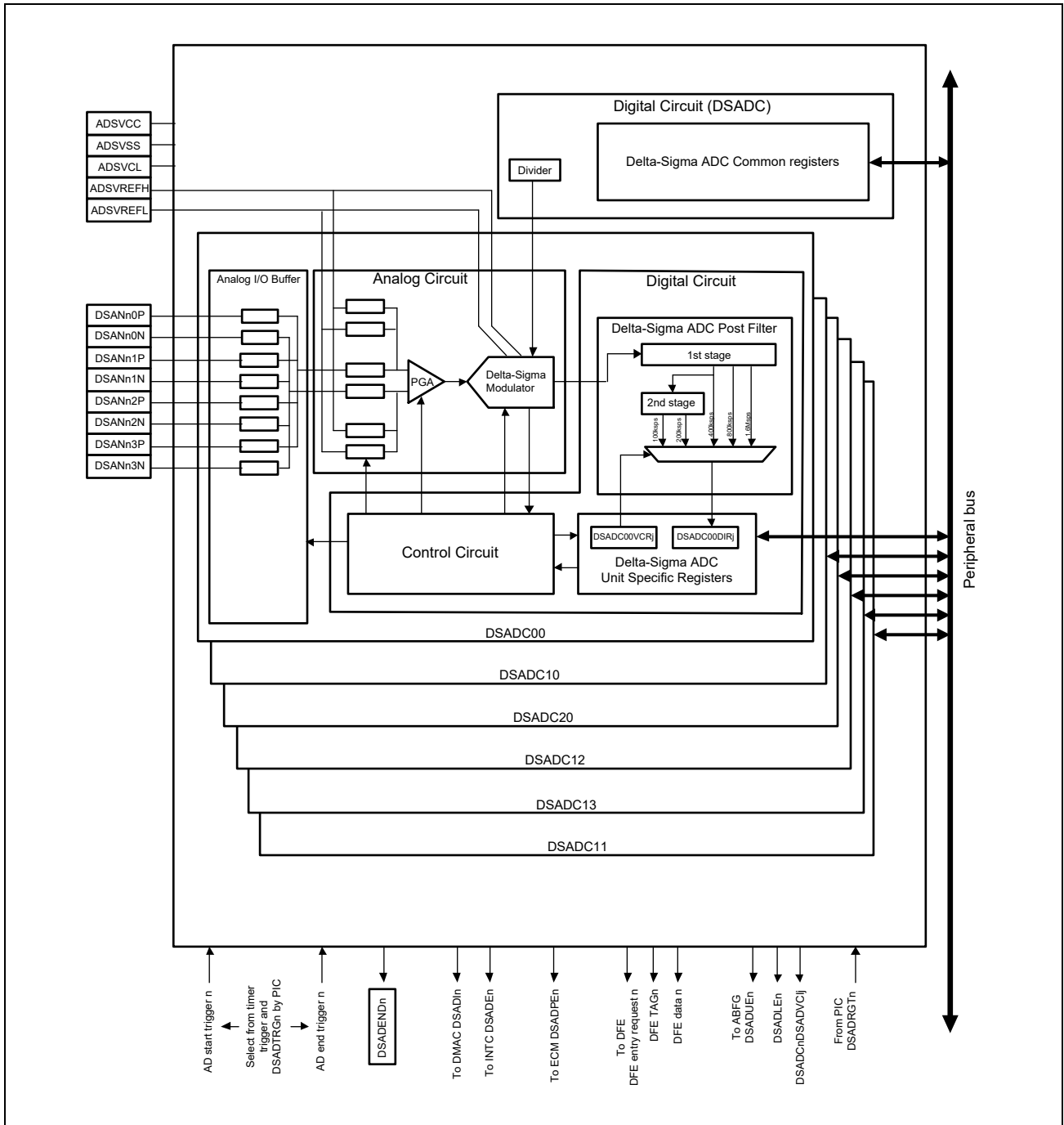


Figure 35.1 Block Diagram of DSADC

35.3 Registers

35.3.1 List of Registers

The DSADC registers are listed in following table.

For details about <DSADC_base> and <DSADCn_base>, see **Section 35.1.2, Register Base Address**.

Table 35.13 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection
DSADC	AD Synchronization Start Control Register	DSADCSYNSTCR	<DSADC_base> + 00 _H	8	—
DSADC	AD Global Control Register	DSADCADGCR	<DSADC_base> + 04 _H	8	—
DSADC	Pin Level Self-Diagnosis Control Register	DSADCTDCR	<DSADC_base> + 0C _H	8	—
DSADCn	Virtual Channel Register 0*1	DSADCnVCR0	<DSADCn_base> + 00 _H	8, 16, 32	—
DSADCn	Virtual Channel Register 1*1	DSADCnVCR1	<DSADCn_base> + 04 _H	8, 16, 32	—
DSADCn	Virtual Channel Register 2*1	DSADCnVCR2	<DSADCn_base> + 08 _H	8, 16, 32	—
DSADCn	Virtual Channel Register 3*1	DSADCnVCR3	<DSADCn_base> + 0C _H	8, 16, 32	—
DSADCn	Virtual Channel Register 4*1	DSADCnVCR4	<DSADCn_base> + 10 _H	8, 16, 32	—
DSADCn	Virtual Channel Register 5*1	DSADCnVCR5	<DSADCn_base> + 14 _H	8, 16, 32	—
DSADCn	Virtual Channel Register 6*1	DSADCnVCR6	<DSADCn_base> + 18 _H	8, 16, 32	—
DSADCn	Virtual Channel Register 7*1	DSADCnVCR7	<DSADCn_base> + 1C _H	8, 16, 32	—
DSADCn	Data Supplementary Information Register 0*1	DSADCnDIR0	<DSADCn_base> + 20 _H	16*3, 32	—
DSADCn	Data Supplementary Information Register 1*1	DSADCnDIR1	<DSADCn_base> + 24 _H	16*3, 32	—
DSADCn	Data Supplementary Information Register 2*1	DSADCnDIR2	<DSADCn_base> + 28 _H	16*3, 32	—
DSADCn	Data Supplementary Information Register 3*1	DSADCnDIR3	<DSADCn_base> + 2C _H	16*3, 32	—
DSADCn	Data Supplementary Information Register 4*1	DSADCnDIR4	<DSADCn_base> + 30 _H	16*3, 32	—
DSADCn	Data Supplementary Information Register 5*1	DSADCnDIR5	<DSADCn_base> + 34 _H	16*3, 32	—
DSADCn	Data Supplementary Information Register 6*1	DSADCnDIR6	<DSADCn_base> + 38 _H	16*3, 32	—
DSADCn	Data Supplementary Information Register 7*1	DSADCnDIR7	<DSADCn_base> + 3C _H	16*3, 32	—
DSADCn	AD Start Control Register	DSADCnADSTCR	<DSADCn_base> + 40 _H	8	—
DSADCn	AD Stop Control Register	DSADCnADENDCR	<DSADCn_base> + 44 _H	8	—
DSADCn	Calibration Start Control Register	DSADCnCLBSTCR	<DSADCn_base> + 48 _H	8	—
DSADCn	Calibration Stop Control Register	DSADCnCLBEDCR	<DSADCn_base> + 4C _H	8	—
DSADCn	AD Conversion Trigger Control Register	DSADCnADTCCR	<DSADCn_base> + 50 _H	8	—
DSADCn	Unit Control Register	DSADCnUCR	<DSADCn_base> + 54 _H	8, 16, 32	—
DSADCn	Virtual Channel Pointer Control Register	DSADCnVCPTRR	<DSADCn_base> + 58 _H	8	—
DSADCn	AD Conversion Status Register	DSADCnADSR	<DSADCn_base> + 60 _H	8	—
DSADCn	Unit Data Pointer Register	DSADCnUDPTRR	<DSADCn_base> + 64 _H	8	—
DSADCn	Unit Data Supplementary Information Register	DSADCnUDIR	<DSADCn_base> + 68 _H	32	—
DSADCn	Timestamp Register	DSADCnTSVAL	<DSADCn_base> + 6C _H	32	—

Table 35.13 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection
DSADCn	Safety Control Register	DSADCnSFTCR	<DSADCn_base> + 70 _H	8	—
DSADCn	Error Clear Register	DSADCnECR	<DSADCn_base> + 74 _H	8	—
DSADCn	Error Register	DSADCnER	<DSADCn_base> + 78 _H	32	—
DSADCn	Pin Level Self-Diagnosis Level Register	DSADCnTDLVR	<DSADCn_base> + 7C _H	8	—
DSADCn	Upper-Limit/Lower-Limit Table Register 0*2	DSADCnULTBR0	<DSADCn_base> + 80 _H	16, 32	—
DSADCn	Upper-Limit/Lower-Limit Table Register 1*2	DSADCnULTBR1	<DSADCn_base> + 84 _H	16, 32	—
DSADCn	Upper-Limit/Lower-Limit Table Register 2*2	DSADCnULTBR2	<DSADCn_base> + 88 _H	16, 32	—
DSADCn	Upper-Limit/Lower-Limit Table Register 3*2	DSADCnULTBR3	<DSADCn_base> + 8C _H	16, 32	—

- Note 1. For n = 00: "Virtual Channel Register" and "Data Supplementary Information Register" numbers are from 0 to 7.
 For n = 10/12/13/11: "Virtual Channel Register" and "Data Supplementary Information Register" numbers are from 0 to 3.
 For n = 20: "Virtual Channel Register" and "Data Supplementary Information Register" numbers are from 0 to 1.
- Note 2. For n = 00: "Upper-Limit/Lower-Limit Table Register" numbers are from 0 to 3.
 For n = 10/12/13/11: "Upper-Limit/Lower-Limit Table Register" numbers are from 0 to 1.
 For n = 20: "Upper-Limit/Lower-Limit Table Register" number is 0.
- Note 3. When reading DSADCnDIRj in 16-bit units, see **Section 35.5.2, DSADCnDIRj — Data Supplementary Information Register j**.

35.4 Delta-Sigma ADC Common Registers

This section describes the registers shared by all Delta-Sigma ADCs (DSADCn).

35.4.1 DSADCSYNSTCR — AD Synchronization Start Control Register

DSADCSYNSTCR is an 8-bit write-only register to control the simultaneous start of each Delta-Sigma ADC. This register is always read as 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADSTART
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 35.14 DSADCSYNSTCR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ADSTART	A/D Conversion Start of Each Delta-Sigma ADC A/D conversion start condition of DSADCn with ADSTART: When ADSTART is set to 1 with ADSTTE = 1. All DSADCn in which ADSTARTE is set to 1 are activated simultaneously. However, DSADCB and DSADCA operate with a shift of half oversampling rate cycles. When 1 is written to this bit with A/D conversion operating, if the DSADCnUCR.VCEP[2:0] value is 0 _H , the ongoing A/D conversion stops and A/D conversion of the virtual channel 0 starts. When 1 is written to this bit with A/D conversion operating, if the DSADCnUCR.VCEP[2:0] value is not 0 _H , the ongoing A/D conversion stops and A/D conversion of the next virtual channel starts. When 1 is written to this bit in DSADCnADTCR.ADSTTE = 0, it is ignored and A/D conversion does not start. Writing 0 to this bit is ignored. Note: DSADCA is DSADC00, DSADC20 and DSADC13. DSADCB is DSADC10, DSADC12 and DSADC11.

NOTE

We recommend that ADSTART be written with STTRGE and ENDTRGE for all Delta-Sigma ADCs set to 0 and not used with external triggers.

35.4.2 DSADCADGCR — AD Global Control Register

DSADCADGCR is an 8-bit read/write register to fully control the AD.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	ODDE	ODE	—	—	—	UNSN
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R/W

Table 35.15 DSADCADGCR Register Contents

Bit Position	Bit Name	Function
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	ODDE	Wiring-break Detection Function Self-Diagnosis Enable 0: Self-diagnosis of the wiring-break detection function is disabled. 1: Self-diagnosis of the wiring-break detection function is enabled. When ODDE is set to 1, wiring-break detection function self-diagnosis and pin level self-diagnosis are enabled. When doing pin level self-diagnosis, set ODDE to 1 before setting TDE to 1. If TDE is set to 1 with ODDE = 0, the self-diagnosis pin level voltage is output to the chip pin, which might affect the external circuit of the connected device.
4	ODE	Wiring-break Detection Enable 0: Wiring-break detection is disabled. 1: Wiring-break detection is enabled. When ODE is set to 1, wiring-break detection is enabled for all analog pins. Setting this bit in 1 simultaneously with pin level self-diagnosis (TDE=1) is prohibited. When setting DSADCADGCR.ODE as 1, if DSADCnVCRj.CNVCLS[1:0] value is set to anything but 0 _H , A/D conversion is prohibited.
3 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	UNSN	Unsigned Conversion Result Output Enable When this bit is set to 1, the A/D conversion result (when ADSVREFL is selected for the single-ended common voltage) can be output as an unsigned value. 0: The A/D conversion result is output as a signed value. 1: When single-ended common voltage = ADSVREFL, the A/D conversion result is output as an unsigned value.*1 Otherwise, it is output as a signed value. For differences in data format due to UNSN setting, see the description on the Section 35.5.2, DSADCnDIRj — Data Supplementary Information Register j format.

Note 1. When the calculation result after correction is a negative value, an A/D conversion result of 0 is output.

NOTE

To prevent a malfunction, set DSADCADGCR when ADACT and CLBACT of all Delta-Sigma ADCs = 0 (before starting an A/D conversion) and STTRGE of all Delta-Sigma ADCs = 0.

35.4.3 DSADCTDCR — Pin Level Self-Diagnosis Control Register

DSADCTDCR is an 8-bit read/write register to control pin level self-diagnosis.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TDE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 35.16 DSADCTDCR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	TDE	<p>Pin Level Self-Diagnosis Enable</p> <p>0: Pin level self-diagnosis is disabled. 1: Pin level self-diagnosis is enabled.</p> <p>When TDE is set to 1, all analog pins are disconnected from the analog I/O buffer. When TDE is set to 0, all analog pins are connected to the analog I/O buffer.</p> <p>When TDE = 1, analog pin DSANn0x to DSANn3x is fixed to the level specified by DSADCnTDLVR.AN0xLV to DSADCnTDLVR.AN3xLV.</p> <p>By performing an A/D conversion in this state and checking the A/D converted value, the path between the analog pin and the Delta-Sigma ADC can be diagnosed.</p> <p>Do not set this bit as 1 simultaneously with ODE = 1 (wiring-break detection enable) and CNVCLS = 3_H (A/D self-diagnosis).</p> <p>When doing pin level self-diagnosis, set ODDE to 1 before setting TDE to 1.</p> <p>If TDE is set to 1 with ODDE = 0, the self-diagnosis pin level voltage is output to the chip pin, which might affect the external circuit of the connected device.</p>

NOTE

To prevent a malfunction, set DSADCTDCR when ADOACT and CLBACT of all delta-sigma ADCs = 0 (before starting A/D conversion) and STTRGE of all delta-sigma ADCs = 0.

35.5 Delta-Sigma ADC Unit-Specific Registers

This section describes registers provided for each Delta-Sigma ADC unit.

35.5.1 DSADCnVCRj — Virtual Channel Register j

DSADCnVCRj is a 32-bit read/write register to control virtual channels.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	GAIN[1:0]		VCULME	VCLLME	VCULLMTBS[1:0]		ORT	TPVSL[2:0]			—	—	DSDFTYP[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADIE	ULEIE	—	DFENT	DFTAG[3:0]			—	—	CNVCLS[1:0]		GCTRL[3:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 35.17 DSADCnVCRj Register Contents (1/3)

Bit Position	Bit Name	Function
31, 30	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
29, 28	GAIN[1:0]	Gain These bits specify the input gain. 0 _H : × 1 1 _H : × 2 2 _H : × 4 (When high resolution mode is set, this setting is prohibited.) 3 _H : × 8 (When high resolution mode is set, this setting is prohibited.)
27	VCULME	Virtual Channel Upper-Limit Excess Notification Enable 0: An A/D conversion greater than the result upper-limit of the virtual channel is not notified. 1: An A/D conversion greater than the result upper-limit of the virtual channel is notified.
26	VCLLME	Virtual Channel Lower-Limit Excess Notification Enable 0: An A/D conversion less than the result lower-limit of virtual channel is not notified. 1: An A/D conversion less than the result lower-limit of virtual channel is notified.
25, 24	VCULLMTBS [1:0]	Virtual Channel Upper-Limit/Lower-Limit Table Register Select These bits select the upper-limit/lower-limit table register to be compared. 0 _H : Upper-limit/lower-limit values are checked by DSADCnULTBR0. [Initial value] 1 _H : Upper-limit/lower-limit values are checked by DSADCnULTBR1. 2 _H : Upper-limit/lower-limit values are checked by DSADCnULTBR2. 3 _H : Upper-limit/lower-limit values are checked by DSADCnULTBR3. When the A/D converted value is stored in the DR, upper-limit/lower-limit values are checked by using the upper-limit value/lower-limit value table.

Table 35.17 DSADnVCRj Register Contents (2/3)

Bit Position	Bit Name	Function
23	ORT	<p>Delta-Sigma ADC Post-Filter 2nd Stage Output Rate Select</p> <p>This bit is invalid when TPVSL[2:0] = 0_H (2nd stage not used).</p> <p>0: The 2nd stage output rate is set to 1/2.</p> <p>1: The 2nd stage output rate is set to 1/4.</p> <p>The output rate (entire filter type) is determined in combination with the DSDFTYP set value. Specifications of the settable filter type are provided in Table 35.36.</p> <p>Setting to any combination of ORT, TPVSL, and DSDFTYP other than the settings shown in Table 35.36 is prohibited.</p>
22 to 20	TPVSL[2:0]	<p>Use of the Delta-Sigma ADC Post-Filter 2nd Stage and the TAP Coefficient Select</p> <p>These bits are valid when DSDFTYP = 0_H. See Table 35.18 Coefficients.</p> <p>0_H: The 2nd stage of the post-filter in the delta-sigma ADC is not used. (Fs = 400ksps)</p> <p>1_H: The 2nd stage of the post-filter in the delta-sigma ADC is used, and coefficient 1 is applied.</p> <p>2_H: The 2nd stage of the post-filter in the delta-sigma ADC is used, and coefficient 2 is applied.</p> <p>3_H: The 2nd stage of the post-filter in the delta-sigma ADC is used, and coefficient 3 is applied.</p> <p>4_H: The 2nd stage of the post-filter in the delta-sigma ADC is used, and coefficient 4 is applied.</p> <p>5 to 7_H: Setting prohibited</p> <p>Specifications of settable filter type are provided in Table 35.36.</p> <p>Setting to any combination of ORT, TPVSL, and DSDFTYP other than the settings shown in Table 35.36 is prohibited.</p>
19, 18	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
17, 16	DSDFTYP[1:0]	<p>Delta-Sigma ADC Post-Filter Type</p> <p>These bits specify the type of the post-filter in the delta-sigma ADC.</p> <p>0_H: The filter type depends on the ORT and TPVSL settings. (See Table 35.36.) Set ORT and TPVSL.</p> <p>1_H: Fs = 1.6 Msps (2nd stage not used, small group delay)</p> <p>2_H: Fs = 800 ksps (2nd stage not used, large group delay)</p> <p>3_H: Setting prohibited</p> <p>Specifications of settable filter type are provided in Table 35.36.</p> <p>Setting to any combination of ORT, TPVSL, and DSDFTYP other than the settings shown in Table 35.36 is prohibited.</p>
15	ADIE	<p>Conversion End Interrupt Enable</p> <p>0: DSADIn is not output by writing to DSADnDIRj.</p> <p>1: DSADIn is output by writing to DSADnDIRj.</p>
14	ULEIE	<p>Upper-Limit/Lower-Limit Error Interrupt Enable</p> <p>0: DSADEn is not output at an upper-limit or lower-limit error of DSADnDIRj.</p> <p>1: DSADEn is output at an upper-limit or lower-limit error of DSADnDIRj.</p>
13	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
12	DFENT	<p>DFE Entry</p> <p>0: Entry is not performed.</p> <p>1: Entry is performed.</p> <p>This bit selects whether to perform entry to the DFE0 or DFE1 (Digital Filter Engine).</p>

Table 35.17 DSADnVCRj Register Contents (3/3)

Bit Position	Bit Name	Function
11 to 8	DFTAG[3:0]	DFE-TAG These bits are used to perform entry to a channel of the DFE for which the TAG identical to the setting of DFTAG[3:0] is set. If the TAG matches the setting of DFTAG[3:0] for multiple channels, entry to multiple channels is performed.
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5, 4	CNVCLS[1:0]	Conversion Type 0 _H : Single-ended input, common voltage = ADVSREFL 1 _H : Single-ended input, common voltage = ADVSREFH/2 2 _H : Differential input 3 _H : Self-diagnosis
3 to 0	GCTRL[3:0]	General Control These bits set the A/D conversion for each conversion mode. Single-ended input (CNVCLS = 0 _H , 1 _H) Analog input channels are selected by the GCTRL[3:0] bits. 0 _H : DSANn0P 1 _H : DSANn0N 2 _H : DSANn1P 3 _H : DSANn1N 4 _H : DSANn2P 5 _H : DSANn2N 6 _H : DSANn3P 7 _H : DSANn3N Setting of unsupported channels is prohibited. Differential input (CNVCLS = 2 _H) Analog input channels are selected by the GCTRL[3:0] bits. 0 _H : DSANn0P/DSANn0N 1 _H : Setting prohibited 2 _H : DSANn1P/DSANn1N 3 _H : Setting prohibited 4 _H : DSANn2P/DSANn2N 5 _H : Setting prohibited 6 _H : DSANn3P/DSANn3N 7 _H : Setting prohibited Setting of unsupported channels is prohibited. Self-diagnosis (CNVCLS[1:0] = 3 _H) Self-diagnosis level is selected by the GCTRL[3:0] bits. 7 _H : ADVSREFH × 1 4 _H : ADVSREFH × 1/2 0 _H : ADVSREFH × 0 C _H : - ADVSREFH × 1/2 8 _H : - ADVSREFH × 1 Other than the above: Setting prohibited NOTE: Setting of self-diagnosis level does not correspond to gain ×8. Set gain ×4 as self-diagnosis in the high impedance mode when using gain of more than ×4 in the high impedance mode. And always set gain ×1 or ×2 in the high resolution mode.

NOTE

To prevent a malfunction, set DSADCnVCRj when ADACT = 0 and CLBACT = 0 (before starting A/D conversion) and STTRGE = 0. When rewriting this register after A/D conversion stop, change the setting after it is confirmed that ADACT was cleared by reading ADSR.

Table 35.18 Coefficients of Post-Filter 2nd Stage in Delta-Sigma ADC

No.	Coefficient 1 (TPVSL = 1 _H) BW = 30 kHz @ 400 kSps	Coefficient 2 (TPVSL = 2 _H) BW = 60 kHz @ 400 kSps	Coefficient 3 (TPVSL = 3 _H) BW = 30 kHz @ 400 kSps	Coefficient 4 (TPVSL = 4 _H) BW = 60 kHz @ 400 kSps
0	681	1649	177	161
1	1940	5765	460	121
2	3911	10863	227	-528
3	6089	12706	-198	-414
4	7674	8333	-958	721
5	7890	101	-1395	1414
6	6397	-5597	-938	-499
7	3581	-4689	810	-3168
8	417	349	3570	-1490
9	-1978	3549	6444	6205
10	-2889	2155	8279	13888
11	-2320	-1082	8279	13888
12	-913	-2167	6444	6205
13	460	-577	3570	-1490
14	1176	1082	810	-3168
15	1105	964	-938	-499
16	556	-170	-1395	1414
17	-26	-694	-958	721
18	-335	-298	-198	-414
19	-332	169	227	-528
20	-210	228	460	121
21	58	-53	177	161

35.5.2 DSADCnDIRj — Data Supplementary Information Register j

DSADCnDIRj is a 32-bit read-only register to store A/D converted values and their supplementary information. This register is cleared to 0000 0000_H by reading this register when RDCLRE is set to 1. Additionally, DSADCnUDIR is also cleared to 0000 0000_H by reading this register which is mirrored to DSADCnUDIR when DSADCnRDCLRE is set to 1. The WFLG bit is cleared by reading this register regardless of the RDCLRE value.

When reading the DSADCnDIRj in 16-bit units, disable Read-and-Clear-Enable (DSADCnSFTCR.RDCLRE = 0) and disable the overwriting function (*1).

Note 1. The expected value comparison of the WFLG bit of DSADCnDIRj and Overwrite-Error of the Error-Register is not put into effect, and the Overwrite-Error-Interrupt-Enable (OWEIE) of the Safety-Control-Register (DSADCnSFTCR) is 0 (prohibited).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	IDEF	WFLG	PRTY	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 35.19 DSADCnDIRj Register Contents

Bit Position	Bit Name	Function
31 to 27	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
26	IDEF	ID Error 0: An error is present. 1: No error is present.
25	WFLG	Write Flag Setting condition Storing an A/D converted value in DSADCnDIRj Clearing condition Reading DSADCnDIRj
24	PRTY	Parity This bit indicates the even parity bit of DR and IDEF.
23 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	DR[15:0]	These bits are used to store the A/D converted value. For the data format, see “Bits 15 to 0 DR Data Register Data Format”.

The setting timing of IDEF, WFLG and PRTY is simultaneous with the timing at which the A/D conversion result is stored up in a data register (DR).

When update (setting) and reading (clearance) of IDEF, WFLG and PRTY are in contention, update (setting) is given priority.

Bits 15 to 0 DR Data Register Data Format

The A/D converted value is stored in the DR. The data format of the DR is shown below. The format of the data to be transferred to the DFE (Digital Filter Engine) or the GTM (Generic Timer Module) is in the signed fixed-point format regardless of the UNSND setting.

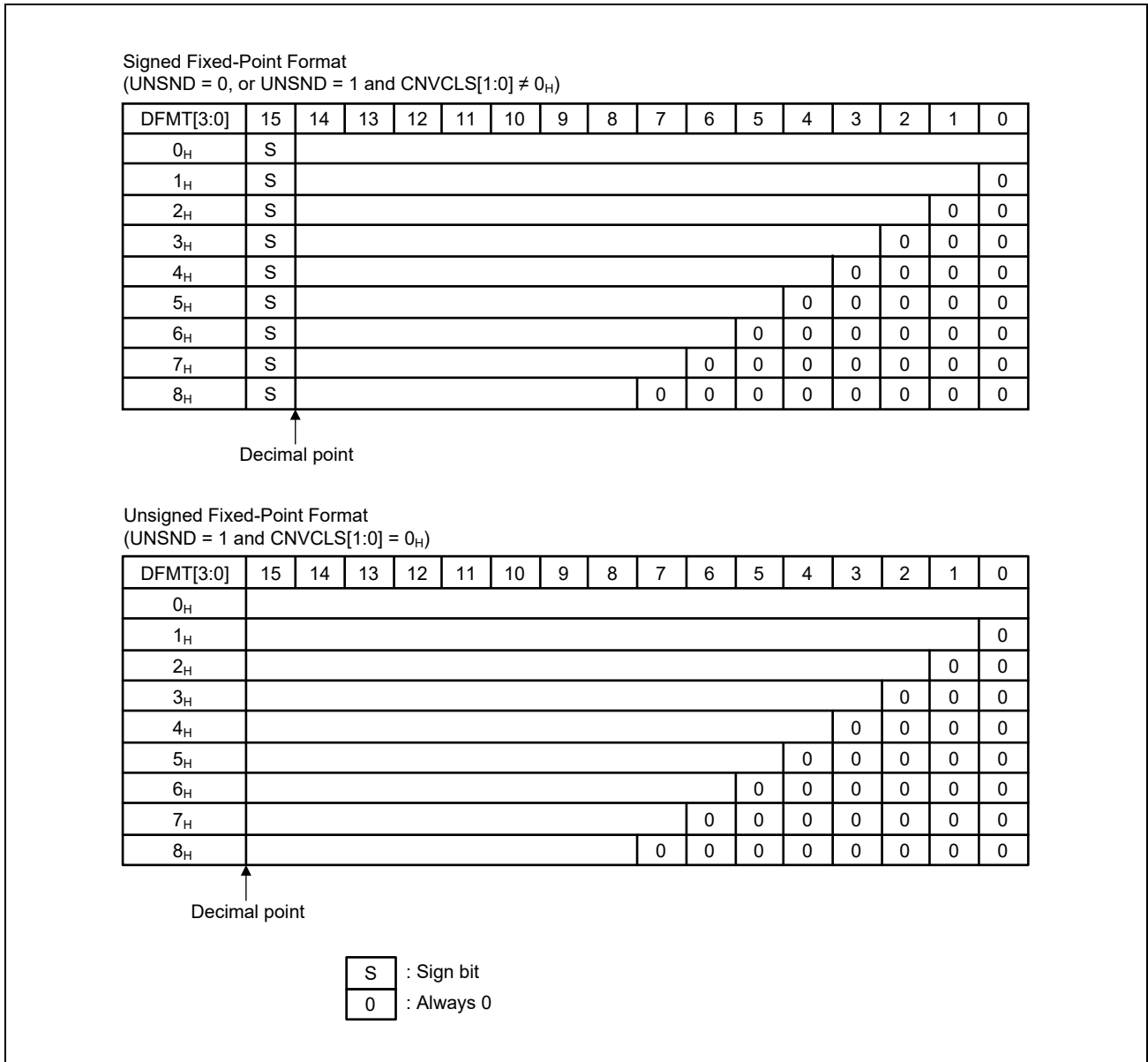


Figure 35.2 DR (Data Register) Data Format 1

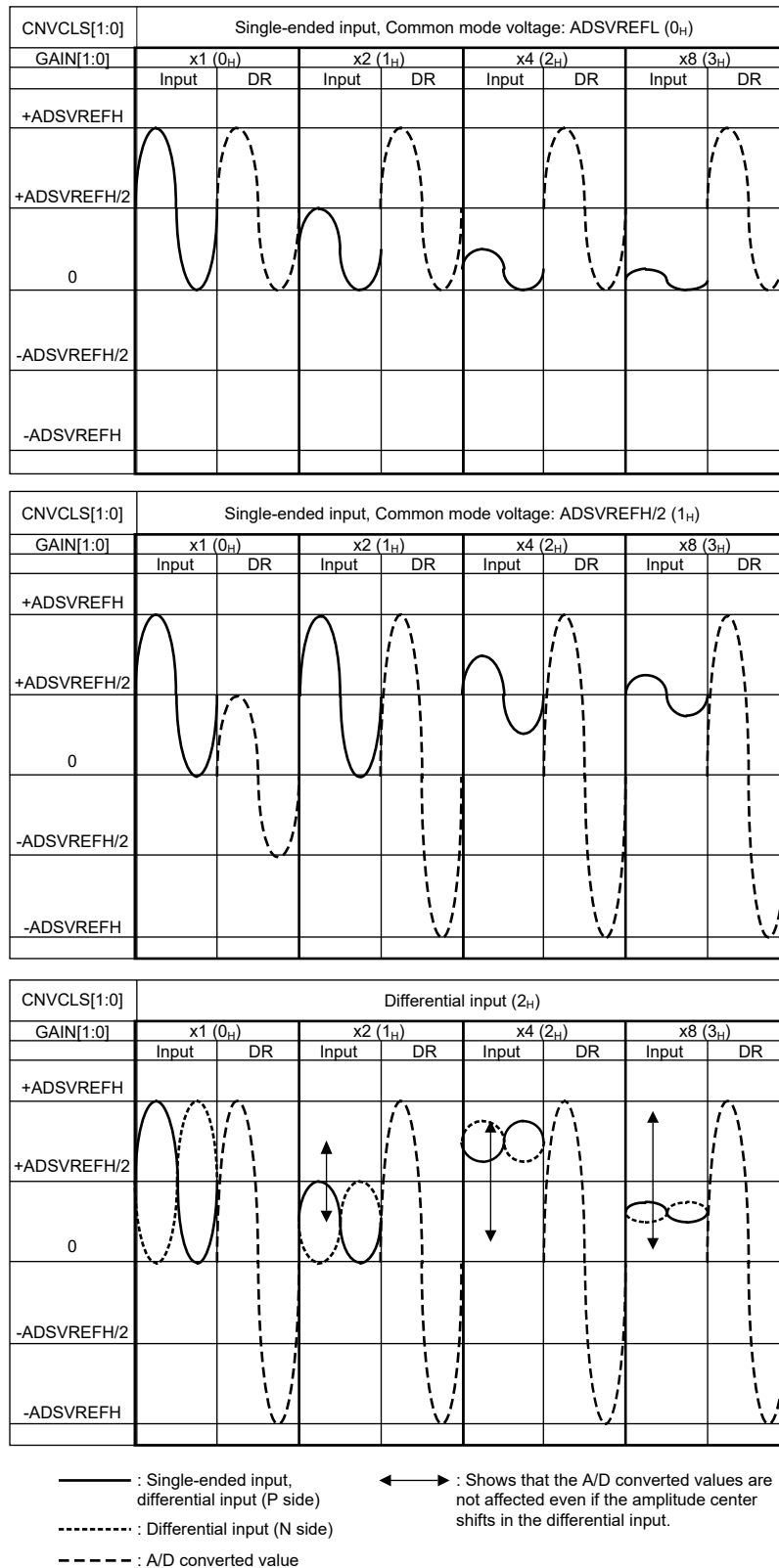


Figure 35.3 DR (Data Register) Data Format 2

35.5.3 DSADCnADSTCR — AD Start Control Register

DSADCnADSTCR is an 8-bit write-only register to control the Delta-Sigma ADC start. This register is always read as 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADST
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 35.20 DSADCnADSTCR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ADST	A/D Conversion Start A/D conversion start condition with ADST Writing 1 to ADST When 1 is written to this bit with A/D conversion operating, the ongoing A/D conversion stops and A/D conversion of the next virtual channel starts. When 1 is written to this bit with A/D conversion operating, if the DSADCnUCR.VCEP[2:0] value is 0 _H , the ongoing A/D conversion stops and A/D conversion of the virtual channel 0 starts. Writing 0 to this bit is ignored.

NOTE

We recommend that a value be written to the ADST bit when STTRGE = 0 and ENDTRGE = 0. When the external trigger and write to ADST occur at the same time, the count value of the channel sometimes does not match the number of inputs of the trigger.

To prevent a malfunction, the ADST bit must be set after confirmation of calibration completion following the A/D conversion flow in **Figure 35.4**.

35.5.4 DSADCnADENDCR — AD Stop Control Register

DSADCnADENDCR is an 8-bit write-only register to control the Delta-Sigma ADC stop. This register is always read as 0.

Value after reset: 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADEND
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 35.21 DSADCnADENDCR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ADEND	A/D Conversion End A/D conversion stop condition with ADEND Writing 1 to this bit stops A/D conversion when A/D conversion is operating. When A/D conversion is stopped with VPRSTE = 0, the virtual channel is incremented. When A/D conversion is stopped with VPRSTE = 1, the virtual channel is cleared to 0. Writing 1 to this bit is ignored when A/D conversion is stopped. Writing 0 to this bit is ignored.

NOTE: When changing the other register setting after an A/D conversion is stopped, please be sure to confirm that ADACT is 0.

NOTE

We recommend that a value be written to the ADEND bit when STTRGE = 0 and ENDTRGE = 0.

35.5.5 DSADCnCLBSTCR — Calibration Start Control Register

DSADCnCLBSTCR is an 8-bit write-only register to control calibration start. This register is always read as 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLBST
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 35.22 DSADCnCLBSTCR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CLBST	Calibration Start Calibration start condition with CLBST Writing 1 to this bit starts calibration when calibration and A/D conversion are stopped. Writing 1 to this bit is ignored when calibration or A/D conversion is operating.

35.5.6 DSADCnCLBEDCR — Calibration Stop Control Register

DSADCnCLBEDCR is an 8-bit write-only register to control calibration stop. This register is always read as 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLBEND
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 35.23 DSADCnCLBEDCR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CLBEND	Calibration Stop Calibration stop condition with CLBEND Writing 1 to this bit stops calibration when calibration is operating. Writing 1 to this bit is ignored when calibration is stopped. CAUTION: If calibration is suspended by CLBEND before calibration is completed, the correction value before calibration is performed is retained. Writing 0 to this bit is ignored.

NOTE

To prevent a malfunction, A/D conversion must be restarted from the initialization process following the A/D conversion flow in **Figure 35.4** after writing 1 to the CLBEND bit in during calibration.

Setting of ADST or CLBST must not be done before at least 6 register access cycles after confirmation of calibration stop.

35.5.7 DSADCnADTCR — AD Conversion Trigger Control Register

DSADCnADTCR is an 8-bit read/write register to control the Delta-Sigma ADC.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ADSTTE	—	—	—	—	ENDTRGE	STTRGE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R	R/W	R/W

Table 35.24 DSADCnADTCR Register Contents

Bit Position	Bit Name	Function
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	ADSTTE	A/D Synchronization Start Enable 0: ADSTART is disabled. 1: ADSTART is enabled.
5 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ENDTRGE	A/D Conversion End Trigger Enable 0: A/D conversion end trigger n is disabled. 1: A/D conversion end trigger n is enabled. The trigger selection function is used to set the trigger source selection for A/D conversion end trigger n. For details, see Section 33, Peripheral Interconnection (PIC) .
0	STTRGE	A/D Conversion Start Trigger Enable 0: A/D conversion start trigger n is disabled. 1: A/D conversion start trigger n is enabled. The trigger selection function is used to set the trigger source selection for A/D conversion start trigger n. For details, see Section 33, Peripheral Interconnection (PIC) .

NOTE

Enable the module standby function when ENDTRGE = 0 and STTRGE = 0. For details, see **Section 35.7.2, Module Standby Function**.

35.5.8 DSADCnUCR — Unit Control Register

DSADCnUCR is a 32-bit read/write register to control the Delta-Sigma AD units.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	VP RSTE	RDMA	—	RESO0	—	DFES	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R	R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DFMT[3:0]				—	—	—	—	—	VCEP[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 35.25 DSADCnUCR Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
29	VPRSTE	Virtual Channel Pointer Reset Enable 0: The virtual channel pointer is incremented at the end of the A/D conversion by the A/D conversion end trigger n or ADEND. 1: The virtual channel pointer is set as 0 at the end of the A/D conversion by the A/D conversion end trigger n or ADEND.
28	RDMA	Read Gate DMA Mode This bit selects a DMA transfer request output condition. 0: A DMA transfer request is output for all A/D conversion results. 1: A DMA transfer request is output for the A/D conversion result for which DSADRGTn is being asserted.
27	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
26	RESO0	High Resolution Mode 0: High resolution mode 1: High impedance mode Be sure to perform calibration after high resolution mode is changed.
25	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
24	DFES	DFE Channel Select This bit sets the DFE channel to be assigned to DSADCn. Entry to the DFE is performed for each VCR register. 0: DFE0 is selected. 1: DFE1 is selected.
23 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 35.25 DSADCnUCR Register Contents (2/2)

Bit Position	Bit Name	Function
11 to 8	DFMT[3:0]	<p>Data Format</p> <p>These bits specify the lower bits to be zero-masked (rounded in the $-\infty$ direction).</p> <p>0_H: Not masked 1_H: Lower 1 bit masked 2_H: Lower 2 bits masked 3_H: Lower 3 bits masked 4_H: Lower 4 bits masked 5_H: Lower 5 bits masked 6_H: Lower 6 bits masked 7_H: Lower 7 bits masked 8_H: Lower 8 bits masked Others: Setting prohibited</p> <p>The setting of these bits depends on the format of the data for DR, DFE, and GTM. For details about the data format, see Section 35.5.2, DSADCnDIRj — Data Supplementary Information Register j.</p>
7 to 3	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
2 to 0	VCEP[2:0]	<p>End Virtual Channel Pointer</p> <p>These bits set the end virtual channel number.</p>

NOTE

To prevent a malfunction, set DSADCnUCRn when A_{DACT} = 0 and CL_{BACT} = 0 (before starting A/D conversion) and ST_{TRGE} = 0.

35.5.9 DSADCnVCPTRR — Virtual Channel Pointer Control Register

DSADCnVCPTRR is an 8-bit read/write register to monitor the number of the virtual channel for which A/D conversion is in progress.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	VCPTR[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 35.26 DSADCnVCPTRR Register Contents

Bit Position	Bit Name	Function
7 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	VCPTR[2:0]	A/D Conversion Ongoing Virtual Channel Number These bits output the number of virtual channels for which A/D conversion is in progress. If a value is written to these bits while A/D conversion is not ongoing, the virtual channel number is updated and A/D conversion starts for the channel written at the next conversion start. Writing to these bits during A/D conversion is prohibited. When any value is written to these bits with A/D conversion operating, the A/D conversion of the next start channel cannot be guaranteed. The number of the channel for which the next A/D conversion is processed is retained while conversion is stopped or calibration is processed. A setting exceeding the end pointer is prohibited. When changing a channel by the external trigger or ADST, it is necessary to wait up to 20 register access cycles until the change is reflected to VCPTR.

NOTE

To prevent a malfunction, set DSADCnVCPTRR when ADACT = 0 and CLBACT = 0 (before starting A/D conversion) and ADSTTE = 0 and STTRGE = 0 and ENDTRGE = 0. When writing during external trigger use, be sure to stop A/D conversion. When writing to VCPTR, set ADSTTE, STTRGE and ENDTRGE as 0, and confirm ADACT = 0, or write ADEND = 1 to ensure that A/D conversion stops.

35.5.10 DSADCnADSR — AD Conversion Status Register

DSADCnADSR is an 8-bit read-only register that indicates the delta-sigma ADC status.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CLBACT	ADACT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 35.27 DSADCnADSR Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CLBACT	Calibration Status 0: Calibration is not in progress. 1: Calibration is in progress.
0	ADACT	A/D Conversion Status 0: A/D conversion is not in progress. 1: A/D conversion is in progress. The ADACT bit is set to 1 when an A/D conversion start source is generated, and it is set to 0 when an A/D conversion end source is generated. This bit is continuously set to 1 unless an A/D conversion end source is generated. (This bit is not cleared to 0 by storing the converted data in the data register.) If an A/D conversion start source is generated when ADACT = 1 (where the period of ADACT=0 just after the ADST set is included, or the period of ADACT=1 just after the ADEND set is not included), the ongoing A/D conversion is stopped and A/D conversion of the next virtual channel is started. An A/D conversion end source generated when ADACT = 0 (where the period of ADACT=0 just after the ADST set is not included, or the period of ADACT=1 just after the ADEND set is included) is ignored. About the timing of setting or clearing of ADACT, see Figure 35.5 .

NOTE

Enable module standby function when ADACT = 0 and CLBACT = 0. For details, see **Section 35.7.2, Module Standby Function**.

35.5.11 DSADCnUDPTRR — Unit Data Pointer Register

DSADCnUDPTRR is an 8-bit read-only register that indicates the virtual channel number of DSADCnUDIR.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	UDPTR[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 35.28 DSADCnUDPTRR Register Contents

Bit Position	Bit Name	Function
7 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	UDPTR	Unit Data Pointer These bits indicate the virtual channel number in which the latest A/D conversion result is stored.

35.5.12 DSADCnUDIR — Unit Data Supplementary Information Register

DSADCnUDIR is a 32-bit read-only register to output all A/D conversion results of DSADCn units regardless of the selection of the virtual channel. The bit besides CHNUM[2:0] and UPRTY of this register is a mirror register of DSADCnDIRj specified by UDPTR[3:0]. When DSADCnUDIR is read, the WFLG of the actual DSADCnDIRj is cleared at the same time, and DR[15:0], PRTY and IDEF in DSADCnDIRj as well as CHNUM[2:0] and UPRTY in this register are also cleared when RDCLRE is set to 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CHNUM[2:0]			—	—	UIDEF	UWFLG	UPRTY	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 35.29 DSADCnUDIR Register Contents

Bit Position	Bit Name	Function
31 to 29	CHNUM[2:0]	Virtual Channel Number Virtual channel number of UDR[15:0]
28, 27	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
26	UIDEF	ID Error 0: An error is present. 1: No error is present.
25	UWFLG	Write Flag Setting condition Stores an A/D converted value in DSADCnUDIR. Clearing condition Reads DSADCnUDIR or DSADCnDIRj of the channel indicated by UDPTR.
24	UPRTY	Parity This bit indicates the even parity bit of UDR, IDEF and CHNUM.
23 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	UDR[15:0]	All A/D converted values of DSADCn are stored in these bits. The data format is the same as DSADCnDIRj.

35.5.13 DSADCnTSVAL — Timestamp Register

DSADCnTSVAL is a 32-bit read-only register to output timestamp data. When DSADRGTn from ATU or GTM is rising edge, the timestamp value and A/D conversion result can be captured in this register.

Value after reset: 007F 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	TSVAL[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 35.30 DSADCnTSVAL Register Contents

Bit Position	Bit Name	Function
31 to 23	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	TSVAL[6:0]	Timestamp Data at the DSADRGTn Rising edge The lapsed time (oversampling cycles) after the previous conversion ended can be monitored. A value of 7F _H is output during A/D conversion stop or the first A/D conversion.
15 to 0	TSDR[15:0]	Timestamp Data Output The A/D conversion result when reading the timestamp can be monitored. It is the signed fixed point format. It does not correspond to DFMT.

35.5.14 DSADCnSFTCR — Safety Control Register

DSADCnSFTCR is an 8-bit read/write register for safety control.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RDCLRE	—	—	—	—	OWEIE	PEIE	IDEIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R/W	R/W	R/W

Table 35.31 DSADCnSFTCR Register Contents

Bit Position	Bit Name	Function
7	RDCLRE	Read and Clear Enable 0: DSADCnDIRj (DSADCnUDIR) is not cleared by reading DSADCnDIRj (DSADCnUDIR). 1: DSADCnDIRj (DSADCnUDIR) is cleared by reading DSADCnDIRj (DSADCnUDIR). The WFLG of DSADCnDIRj (DSADCnUDIR) is cleared by reading DSADCnDIRj (DSADCnUDIR) regardless of the RDCLRE value. When DSADCnUDIR is read after RDCLRE has been set to 1, DSADCnDIRj which corresponds with DSADCnUDPTR is also cleared at the same time. When DSADCnDIRj which corresponds with DSADCnUDPTR is read, DSADCnUDIR is also cleared at the same time.
6 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	OWEIE	Overwrite Error Interrupt Enable 0: Disabled 1: Enabled When using DMA transfer with read-gate, if read-gate is in the inactive state, a DMA request is not output, but, on the other hand, an overwrite error occurs. Therefore, the overwrite Error Interrupt and DMA transfer by the DSADIn with read-gate cannot be used at the same time.
1	PEIE	Parity Error Interrupt Enable 0: Disabled 1: Enabled
0	IDEIE	ID Error Interrupt Enable 0: Disabled 1: Enabled

NOTE

To prevent a malfunction, set DSADCnSFTCR when ADACT = 0 (before starting A/D conversion) and STTRGE = 0.

35.5.15 DSADCnECR — Error Clear Register

DSADCnECR is an 8-bit write-only register to control error clear. This register is always read as 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	ULEC	LLEC	OWEC	PEC	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	W	W	W	W	R

Table 35.32 DSADCnECR Register Contents

Bit Position	Bit Name	Function
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	ULEC	Upper-Limit Error Clear 0: Does not clear the upper-limit error. 1: Clears the upper-limit error.
3	LLEC	Lower-Limit Error Clear 0: Does not clear the lower-limit error. 1: Clears the lower-limit error.
2	OWEC	Overwrite Error Clear 0: Does not clear the overwrite error. 1: Clears the overwrite error.
1	PEC	Parity Error Clear 0: Does not clear the parity error. 1: Clears the parity error.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

35.5.16 DSADCnER — Error Register

DSADCnER is a 32-bit read-only register that indicates errors.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ULE	LLE	—	—	—	ULECAP[2:0]			OWE	—	—	—	—	OWECAP[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE	—	—	—	—	PECAP[2:0]			—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 35.33 DSADCnER Register Contents (1/2)

Bit Position	Bit Name	Function
31	ULE ^(*)	Upper-Limit Error 0: No upper-limit error is present. 1: An upper-limit error is present. Setting condition The A/D converted value exceeds the specified upper-limit table range. Clearing condition Writing 1 to ULEC in DSADCnECR. When LLE is set to 1, ULE is not set to 1 even if an upper-limit error occurred.
30	LLE ^(*)	Lower-Limit Error 0: No lower-limit error is present. 1: A lower-limit error is present. Setting condition The A/D converted value exceeds the specified lower-limit table range. Clearing condition Writing 1 to ULEC in DSADCnECR. When ULE is set to 1, LLE is not set to 1 even if a lower-limit error occurred.
29 to 27	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
26 to 24	ULECAP[2:0]	Upper-Limit/Lower-Limit Error Capture These bits show the virtual channel number in which an upper-limit or lower-limit error occurred. When writing 1 to ULEC, these bits are cleared to 0 _H .

Table 35.33 DSADCnER Register Contents (2/2)

Bit Position	Bit Name	Function
23	OWE	<p>Overwrite Error</p> <p>0: No overwrite error is present. 1: An overwrite error is present.</p> <p>Setting condition Writing an A/D converted value to DSADCnDIRj.DR occurs when WFLG = 1. Even if DSADCnUDIR.UDR is overwritten when DSADCnUDIR.UWFLG = 1, this bit is not set to 1 if DSADCnDIRj.DR is not overwritten.</p> <p>Clearing condition Writing 1 to OWEC in DSADCnECR.</p>
22 to 19	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
18 to 16	OWECAP[2:0]	<p>Overwrite Error Capture</p> <p>These bits show the virtual channel number in which an overwrite error occurred. When writing 1 to OWEC, these bits are cleared to 0_H.</p>
15	PE	<p>Parity Error</p> <p>0: No parity error is present. 1: A parity error is present.</p> <p>Setting condition A parity error is detected.</p> <p>Clearing condition Writing 1 to PEC in DSADCnECR.</p>
14 to 11	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
10 to 8	PECAP[2:0]	<p>Parity Error Capture</p> <p>These bits show the virtual channel number in which a parity error occurred. When writing 1 to PEC, these bits are cleared to 0_H. When a parity error is indicated by DSADCnUDIR, the number of DSADCnUDIR.CHNUM[2:0] is shown.</p>
7 to 0	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>

NOTES

The ULE and OWE bits are updated when an A/D converted value is written to the DIR. The PE bit is updated when the DIR is read.

When an AD error interrupt request is generated, read the error register and check for the “1” bits in the interrupt exception handler, and then clear them to 0 by using the corresponding clear bits in the ECR.

When an error bit is not cleared and an error occurs during the next A/D conversion, the following operation takes place and the error cannot be detected or identified.

- When the same error (error bit = 1) occurs:
No AD error interrupt request is made.
The relevant error bit in the ER remains 1.
(* ULE and LLE are handled as the same error.
- When a different error (error bit = 0) occurs:
An AD error interrupt request is made.
The error bit that was generated in the previous A/D conversion remains 1 and the relevant error bit in the ER is set to 1.
(* ULE and LLE are handled as the same error.

35.5.17 DSADCnTDLVR — Pin Level Self-Diagnosis Level Register

DSADCnTDLVR is an 8-bit read/write register to specify pin level self-diagnosis levels. Write to this register, when the DSADCTDCR.TDE bit is in the state of 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	AN3NLV	AN3PLV	AN2NLV	AN2PLV	AN1NLV	AN1PLV	AN0NLV	AN0PLV
Value after reset	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 35.34 DSADCnTDLVR Register Contents

Bit Position	Bit Name	Function
7	AN3NLV	Pin Level Self-Diagnosis Level of DSANn3N 0: DSANn3N is discharged (ADSVSS). 1: DSANn3N is charged (ADSVCC).
6	AN3PLV	Pin Level Self-Diagnosis Level of DSANn3P 0: DSANn3P is discharged (ADSVSS). 1: DSANn3P is charged (ADSVCC).
5	AN2NLV	Pin Level Self-Diagnosis Level of DSANn2N 0: DSANn2N is discharged (ADSVSS). 1: DSANn2N is charged (ADSVCC).
4	AN2PLV	Pin Level Self-Diagnosis Level of DSANn2P 0: DSANn2P is discharged (ADSVSS). 1: DSANn2P is charged (ADSVCC).
3	AN1NLV	Pin Level Self-Diagnosis Level of DSANn1N 0: DSANn1N is discharged (ADSVSS). 1: DSANn1N is charged (ADSVCC).
2	AN1PLV	Pin Level Self-Diagnosis Level of DSANn1P 0: DSANn1P is discharged (ADSVSS). 1: DSANn1P is charged (ADSVCC).
1	AN0NLV	Pin Level Self-Diagnosis Level of DSANn0N 0: DSANn0N is discharged (ADSVSS). 1: DSANn0N is charged (ADSVCC).
0	AN0PLV	Pin Level Self-Diagnosis Level of DSANn0P 0: DSANn0P is discharged (ADSVSS). 1: DSANn0P is charged (ADSVCC).

NOTE

To prevent a malfunction, set DSADCnTDLVR when ADACT = 0 (before starting A/D conversion) and STTRGE = 0.

35.5.18 DSADCnULTBR0/1/2/3 — Upper-Limit/Lower-Limit Table Register 0/1/2/3

DSADCnULTBR0 to DSADCnULTBR3 are 32-bit read/write registers to set the upper-limit and lower-limit values of the A/D converted values.

Value after reset: 7FFF 8000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ULMTB[15:0]															
Value after reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LLMTB[15:0]															
Value after reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 35.35 DSADCnULTBR0/1/2/3 Register Contents

Bit Position	Bit Name	Function
31 to 16	ULMTB[15:0]	<p>Upper-Limit Table</p> <p>These bits specify the upper-limit value of the A/D converted value. When the following condition is met, the ULE (upper-limit error) is set to 1.</p> <p style="padding-left: 20px;">ULMTB[15:0] < A/D converted value</p> <p>The ULMTB[15:0] format is the same as the DR format.</p> <p>For the unsigned format, it is compared with the A/D converted value as unsigned.</p> <p>The A/D converted values are compared by using the values before they are masked by DFMT[3:0].</p>
15 to 0	LLMTB[15:0]	<p>Lower-Limit Table</p> <p>These bits specify the lower-limit value of the A/D converted value. When the following condition is met, the LLE (lower-limit error) is set to 1</p> <p style="padding-left: 20px;">LLMTB[15:0] > A/D converted value</p> <p>The LLMTB[15:0] format is the same as the DR format.</p> <p>For the unsigned format, it is compared with the A/D converted value as unsigned.</p> <p>The A/D converted values are compared by using the values before they are masked by DFMT[3:0].</p>

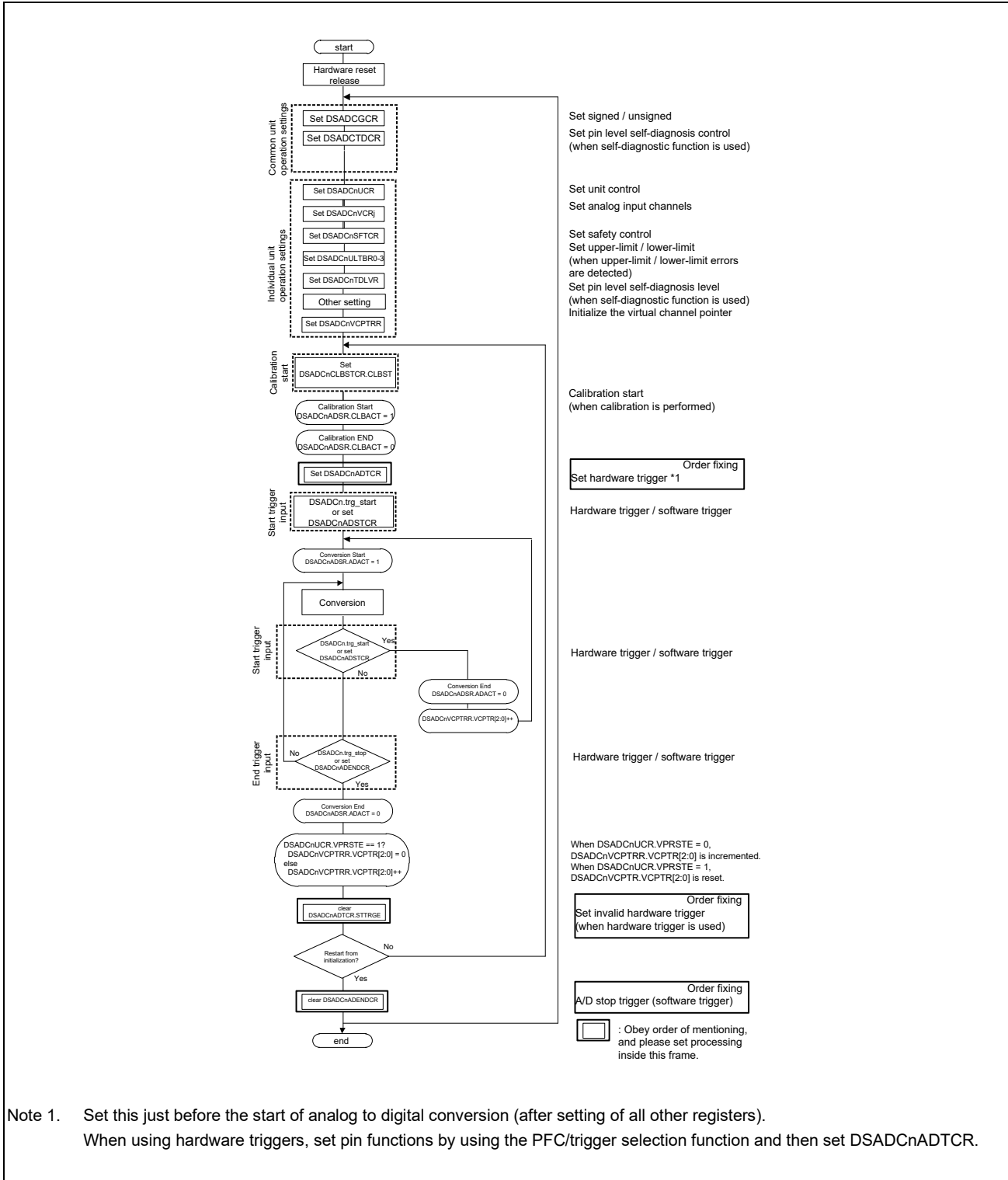
NOTE

To prevent a malfunction, set DSADCnULTBR0 to DSADCnULTBR3 when ADACT = 0 (before starting A/D conversion) and STTRGE = 0.

35.6 Operation

35.6.1 A/D Conversion Flow

The delta-sigma ADC starts A/D conversion by setting the registers shown in **Figure 35.4**.



Note 1. Set this just before the start of analog to digital conversion (after setting of all other registers).
When using hardware triggers, set pin functions by using the PFC/trigger selection function and then set DSADChADTCR.

Figure 35.4 A/D Conversion Flow

35.6.2 Filter Type Setting

Filter types vary with the combinations of the values set in DSADCnADGCR, DSADCnUCR, and DSADCnVCRj. Set the filter types according to the purpose, as shown below.

Table 35.36 Register Set Values for Each Filter Type

Item	2nd Stage Used Case 1 (F1a)	2nd Stage Used Case 2 (F1b)	2nd Stage Used Case 3 (F2)	2nd Stage Used Case 4 (F3a)	2nd Stage Used Case 5 (F3b)	2nd Stage Used Case 6 (F4)	Case 7 (F5)	(F7)	(F6)
Target performance									
Fos	8 Msps	8 Msps	8 Msps	8 Msps	8 Msps	8 Msps	8 Msps	8 Msps	8 Msps
Fs	100 ksps	200 ksps	200 ksps	100 ksps	200 ksps	200 ksps	400 ksps	1.6 Msps	800 ksps
BW	30 kHz	30 kHz	60 kHz	30 kHz	30 kHz	60 kHz	90 kHz	200 kHz	100 kHz
ENOB (high resolution, differential, gain × 2)	13 bit	13 bit	13 bit	13 bit	13 bit	13 bit	10.5 bit	9 bit	10 bit
ENOB (high impedance, differential, gain × 1)	12 bit	12 bit	12 bit	12 bit	12 bit	12 bit	10.5 bit	9 bit	10 bit
Group delay	15 μs ^{*1}	15 μs ^{*1}	10 μs ^{*1}	32 μs ^{*1}	32 μs ^{*1}	32 μs ^{*1}	4.5 μs ^{*1}	2.025 μs ^{*1}	3.875 μs ^{*1}
Register setting									
DSADCnVCRj.DSDFTyp	0 _H	0 _H	0 _H	0 _H	0 _H	0 _H	0 _H	1 _H	2 _H
DSADCnVCRj.ORT	1 _H	0 _H	0 _H	1 _H	0 _H	0 _H	0 _H	0 _H	0 _H
DSADCnVCRj.TPVSL [2:0]	1 _H	1 _H	2 _H	3 _H	3 _H	4 _H	0 _H	0 _H	0 _H

Note 1. Software delay excluded

35.6.3 A/D Conversion Time

After the ADST bit in DSADCnADSTCR is set to 1 and the A/D conversion start delay time (tD) and the internal stabilization time (tW) have passed, the delta-sigma ADC starts A/D conversion processing.

Figure 35.5 shows the A/D conversion start timing and end timing. **Figure 35.6** shows the A/D conversion restart timing. **Figure 35.7** shows the time budget of A/D conversion processing. The A/D conversion processing time (ADST setting during A/D conversion stop) (tAD) includes the A/D conversion start delay time (tD), the internal stabilization time (tW), and the delta-sigma ADC post-filter processing delay time (tDF). The A/D conversion processing time (second time and later) (tAD) is the sampling cycle (tS). A/D conversion processing time (ADST setting during A/D conversion operating) (tAD) includes the A/D conversion restart delay time (tR), the internal stabilization time (tW), and the delta-sigma ADC post-filter processing delay time (tDF). **Table 35.37** shows the A/D conversion processing time.

The A/D conversion processing time (tAD) is calculated by the following formula.

A/D conversion processing time (ADST setting during A/D conversion stop): $tAD = tD + tW + tDF$

A/D conversion processing time (ADST setting during A/D conversion operating): $tAD = tR + tW + tDF$

A/D conversion processing time (From the second time for both cases): $tAD = tS$

Table 35.37 A/D Conversion Time (When Register Access Clock = 40 MHz, ADCLK = 80 MHz, Unit: Register Access Clock)

Item	2nd Stage Used Case 1 (F1a)	2nd Stage Used Case 2 (F1b)	2nd Stage Used Case 3 (F2)	2nd Stage Used Case 4 (F3a)	2nd Stage Used Case 5 (F3b)	2nd Stage Used Case 6 (F4)	Case 7 (F5)	(F7)	(F6)
tAD (ADST setting during A/D conversion stop)	2595.5* ¹ to 2602.5* ¹	2595.5* ¹ to 2602.5* ¹	2595.5* ¹ to 2602.5* ¹	2595.5* ¹ to 2602.5* ¹	2595.5* ¹ to 2602.5* ¹	2595.5* ¹ to 2602.5* ¹	383.5* ¹ to 390.5* ¹	180.5* ¹ to 187.5* ¹	333.5* ¹ to 340.5* ¹
tD	16.5* ¹ to 23.5* ¹								
tW	10								
tDF	2569	2569	2569	2569	2569	2569	357	154	307
tAD (ADST setting during A/D conversion operating)	2612.5* ¹ to 2820* ¹	2612.5* ¹ to 2820* ¹	2612.5* ¹ to 2820* ¹	2612.5* ¹ to 2820* ¹	2612.5* ¹ to 2820* ¹	2612.5* ¹ to 2820* ¹	400.5* ¹ to 608* ¹	197.5* ¹ to 255* ¹	350.5* ¹ to 458* ¹
tR	33.5* ¹ to 241* ¹							33.5* ¹ to 91* ¹	33.5* ¹ to 141* ¹
tAD (From the second time for both cases)	400	200	200	400	200	200	100	25	50
tS	400	200	200	400	200	200	100	25	50

Note 1. DSADCA*² is added 2.5 states (half period of over sampling clock).

Note 2. DSADCA is DSADC00, DSADC20 and DSADC13.
DSADCB is DSADC10, DSADC12 and DSADC11.

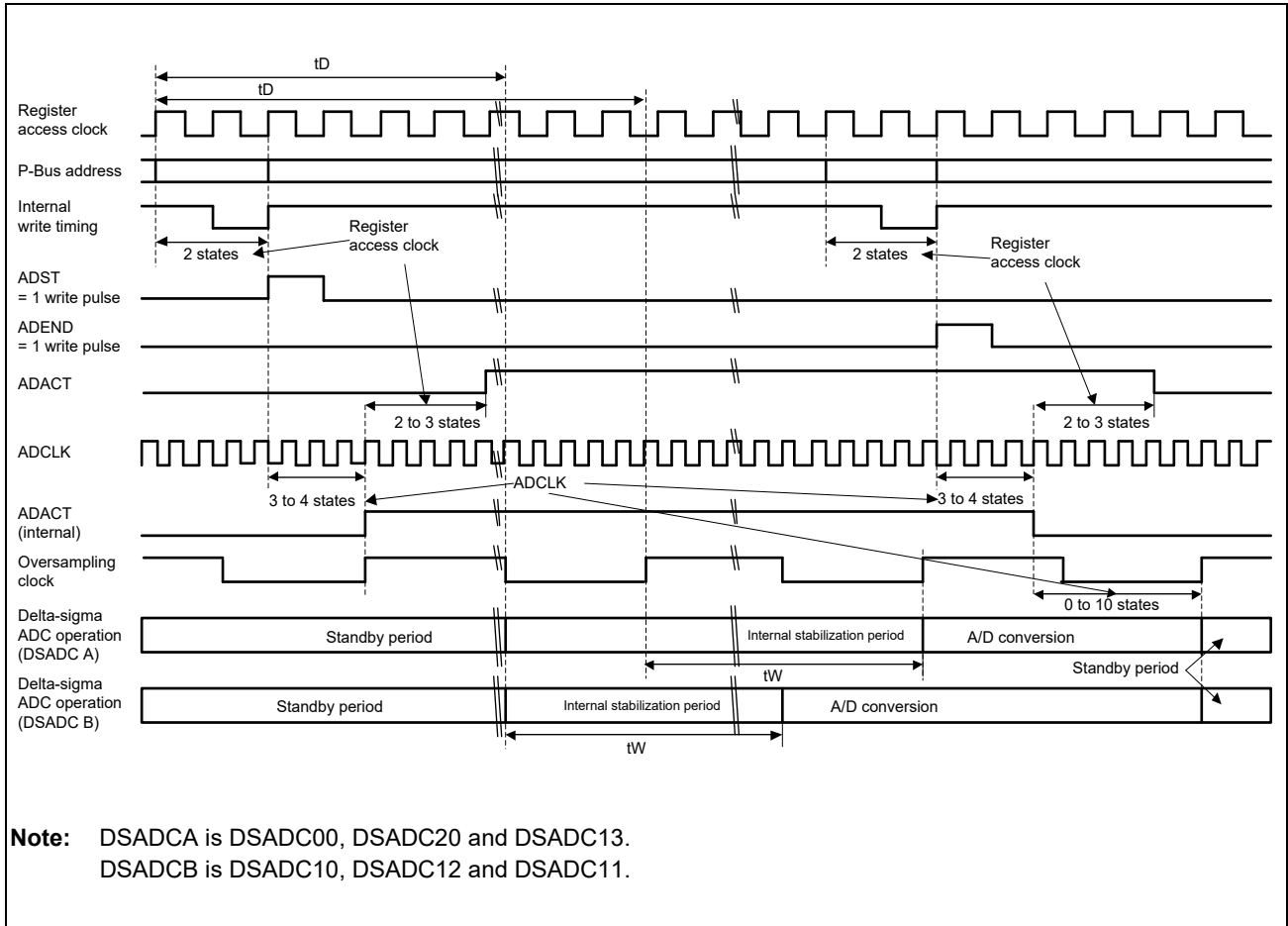


Figure 35.5 A/D Conversion Start and End (when ADST Is Set during Conversion Stop)

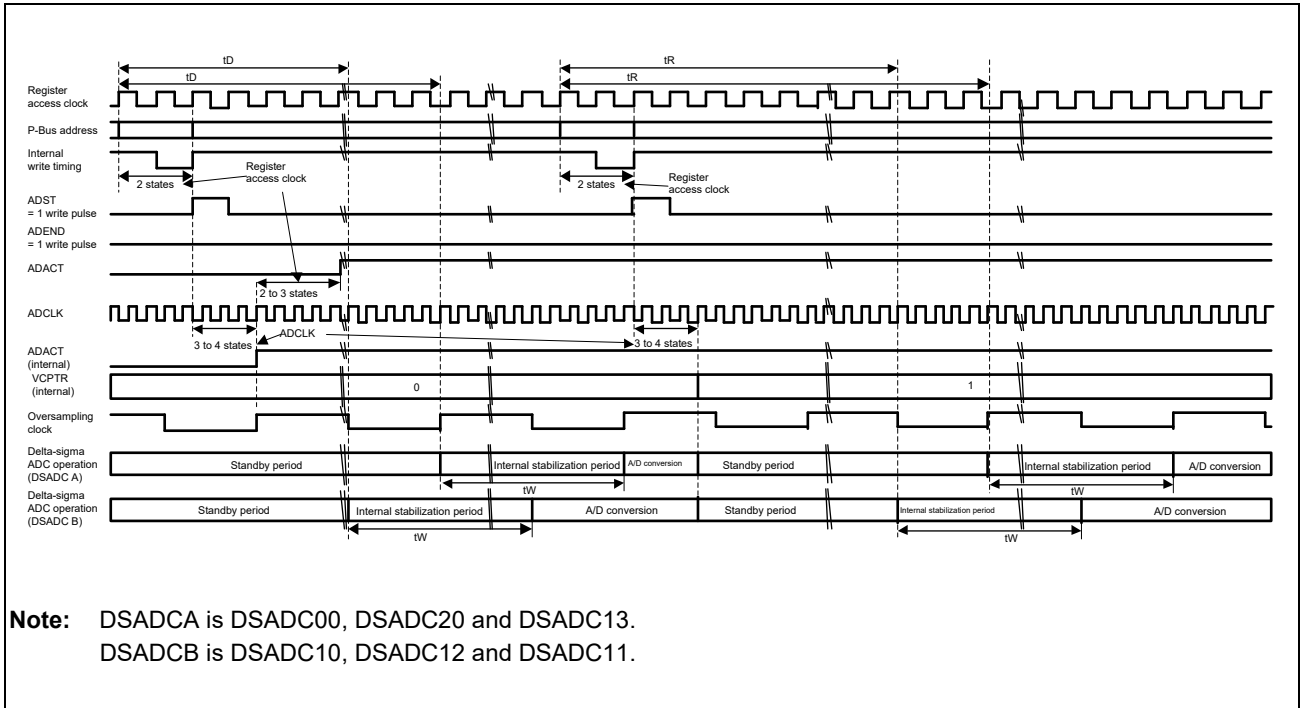


Figure 35.6 A/D Conversion Stop and Restart (when ADST Is Set during Conversion)

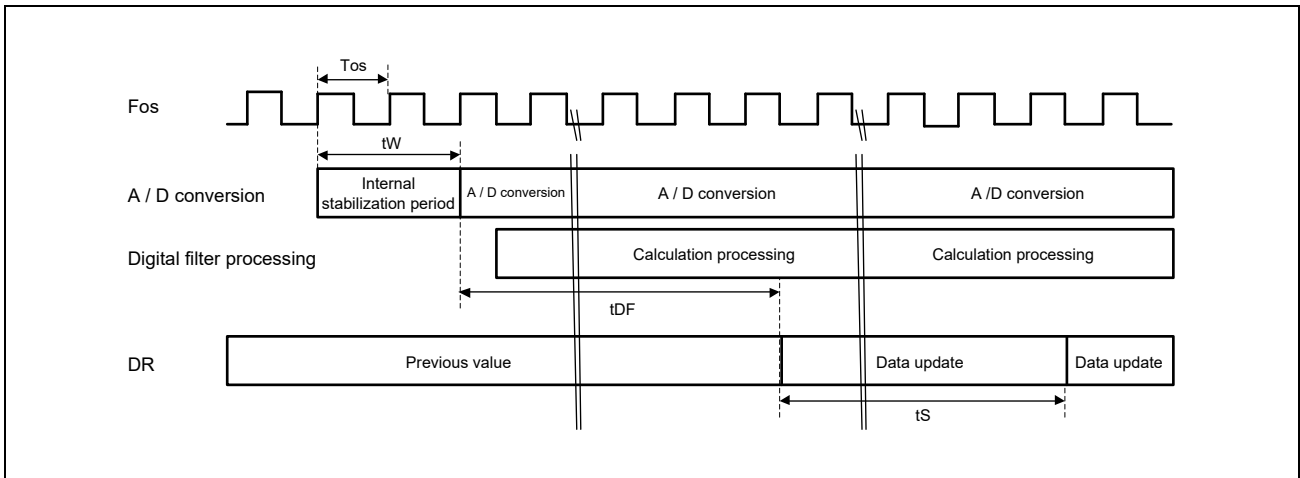


Figure 35.7 Budget of A/D Conversion Processing

35.6.4 Starting an A/D Conversion by an External Trigger

A/D conversion of the DSADC_n can be started by DSADTRG_n input. To start a DSADC_n by DSADTRG_n, input a high level to DSADTRG_n, and then set STTRGE in DSADC_nADTCR to 1. When a low level is input to the DSADTRG_n in this state, the DSADC_n detects a falling edge and sets ADACT to 1.

Figure 35.8 shows the external trigger input timing. ADACT is set to 1 in 3 to 4 states of ADCLK and 2 to 3 states of the register access clock after the falling edge of DSADTRG_n.

The timing after ADACT is set to 1 until DSADC_n is started is the same as the case where ADST is set to 1 by the software. For the pin function settings, see **Section 2, Pin Function**. For selecting the A/D conversion start trigger n source, see **Section 33, Peripheral Interconnection (PIC)**.

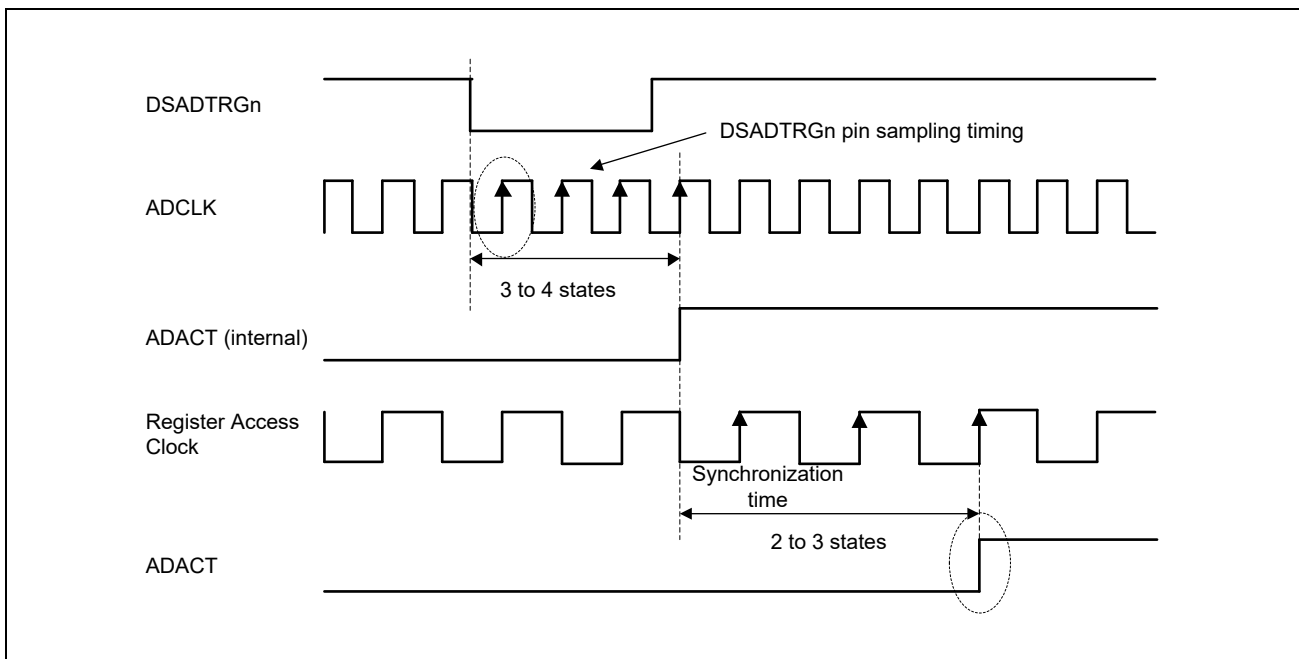


Figure 35.8 External Trigger Input Timing

35.6.5 Terminating an A/D Conversion by an External Trigger

The DSADCn can be terminated by the DSADTRGn input. To terminate the DSADCn by the DSADTRGn, set the pin functions by using the PFC (pin function controller) and select an A/D conversion end trigger n source by using the trigger selection function. Input a high level to DSADTRGn and then set ENDTRGE in DSADCnADTCR to 1. When a low level is input to DSADTRGn in this state, the DSADCn detects a falling edge and clears ADACT to 0 to terminate the DSADCn.

ADACT is cleared to 0 in 3 to 4 states of the ADCLK and 2 to 3 states of the register access clock after the falling edge of DSADTRGn.

For the pin function settings, see **Section 2, Pin Function**. For selecting the A/D conversion end trigger n source, see **Section 33, Peripheral Interconnection (PIC)**.

To terminate an A/D conversion by an external trigger, input an external trigger when ADACT = 1. If an external trigger is input when ADACT = 0, this input is ignored.

35.6.6 Starting an A/D Conversion by a Timer Trigger

The DSADCn can be started by an arbitrary timer trigger. To start the DSADCn by using a timer trigger, select a timer trigger as an A/D conversion start trigger n source by using the trigger selection function. Set STTRGE in DSADCnADTCR to 1.

When the selected timer trigger is input in this state, ADACT is set to 1. The period after ADACT is set to 1 until the DSADCn is started is the same as the case where ADST is set to 1 by the software.

For selecting the A/D conversion start trigger n source, see **Section 33, Peripheral Interconnection (PIC)**.

To stop the DSADCn, write 1 to ADEND in DSADCnADENDCR.

35.6.7 Terminating an A/D Conversion by a Timer Trigger

The DSADCn can be terminated by an arbitrary timer trigger. To terminate the DSADCn by using a timer trigger, select a timer trigger as an A/D conversion end trigger n source by using the trigger selection function. Set ENDTRGE in DSADCnADTCR to 1.

When the selected timer trigger is input in this state, ADACT is cleared to 0 to terminate the DSADCn.

For selecting the A/D conversion end trigger n source, see **Section 33, Peripheral Interconnection (PIC)**.

To terminate an A/D conversion by a timer trigger, input a timer trigger when ADACT = 1. If a timer trigger is input when ADACT = 0, the timer trigger is ignored.

35.6.8 Switching Virtual Channels via Start and End Triggers

When the VCEP[2:0] bits are set as a channel besides 0 in DSADCnUCR, they are incremented (+) at each input of an external trigger (start trigger or stop trigger) or a software trigger (start trigger or stop trigger) during A/D conversion. Virtual channels can be sequentially switched from DSADCnVCR0 to DSADCnVCRj (j = VCEP[2:0] set value) for A/D conversion by inputting a trigger continuously at more than 1 cycle (ADCLK) intervals from the external timer. The channel switching is not performed normally by inputting a start trigger or a stop trigger continuously at less than 1 cycle (ADCLK) intervals. When VPRSTE is set to 1, virtual channels always return to DSADCnVCR0 at a stop trigger. By using this operation, an A/D conversion can also be started from any DSADCnVCRj by continuously generating a start trigger after a stop trigger. If a trigger is input during an A/D conversion of the channel set by VCEP[2:0], the channel returns to DSADCnVCR0 again and then an A/D conversion is performed. **Figure 35.9** to **Figure 35.11** show examples of virtual channel switching made by start triggers and end triggers.

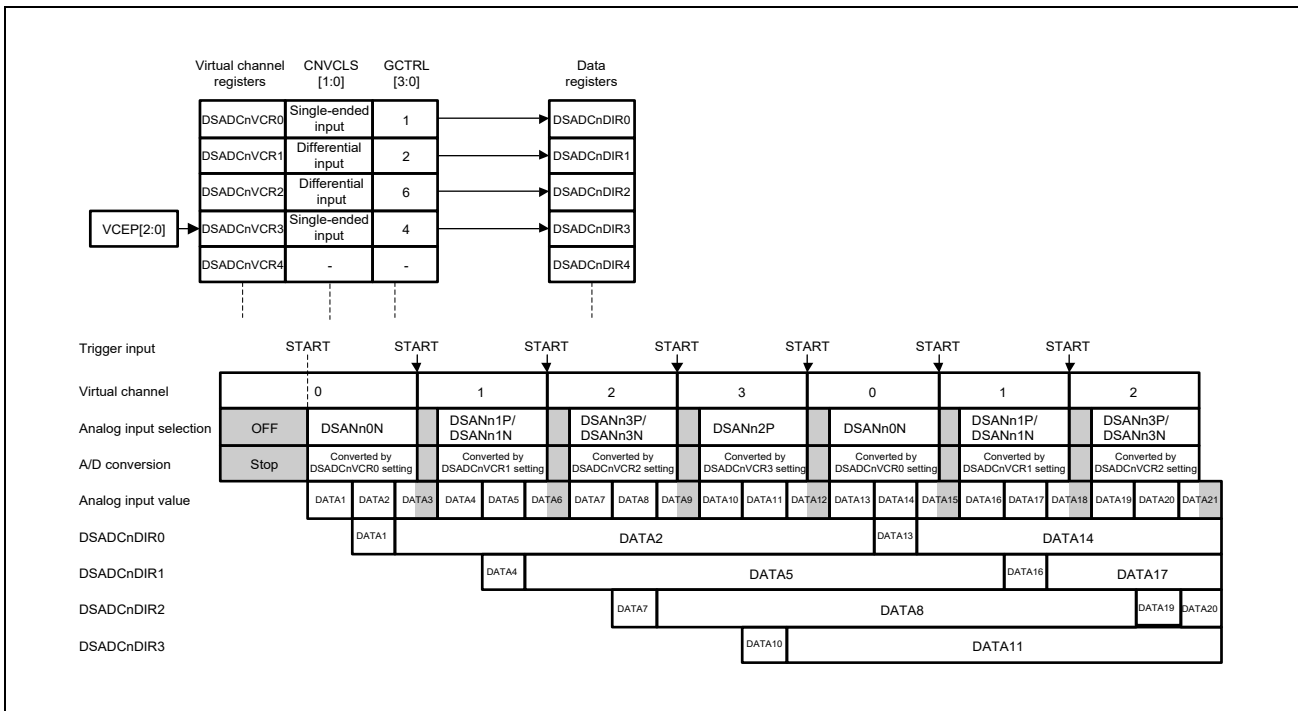


Figure 35.9 Switching Virtual Channels by a Start Trigger

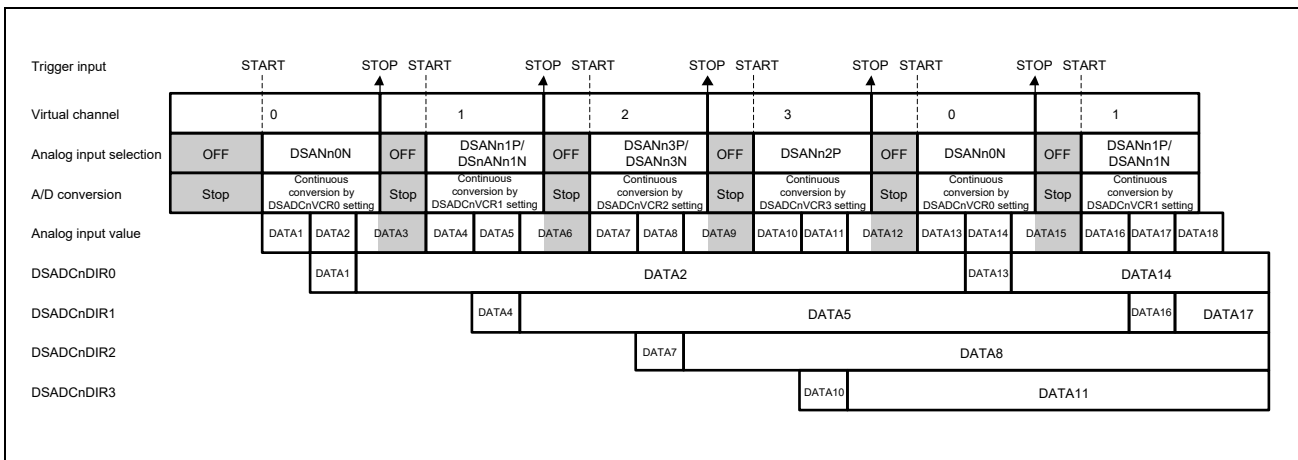


Figure 35.10 Switching Virtual Channels by an End Trigger (VPRSTE = 0)

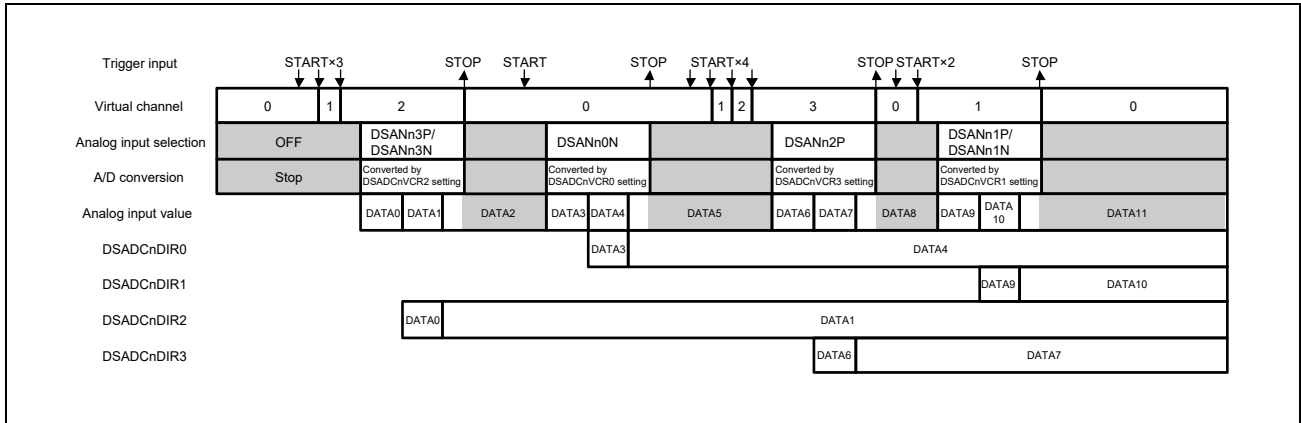


Figure 35.11 Switching Virtual Channels by an End Trigger (VPRSTE = 1)

35.6.9 Calculating Correction Values by Using Calibration

Accurate A/D conversions can be performed by calculating an offset correction value and a gain error correction value under current operating conditions by using calibration. In calibration, the reference voltage is A/D converted in self-diagnosis mode and the best correction values are determined from errors included in the conversion result. Calibration operation starts by writing 1 to CLBST.

35.6.9.1 Calibration Time

Table 35.38 Calibration Time (When Register Access Clock = 40 MHz, ADCLK = 80 MHz, Unit: Register Access Clock)

Conversion Mode	Calibration Time
High Resolution Mode	34907.5 to 35017.5
High Impedance Mode	69827.5 to 70047.5

35.6.10 A/D Conversion Result Error Correction Processing

The post-filter 2nd stage in the Delta-Sigma ADC has the gain error. The gain error specified in Electrical Characteristic are the gain error of A/D conversion result corrected by software processing or by DFE. Correction formula and correction coefficient are shown the following.

Table 35.39 Correction Formula and Correction Coefficient for the Post-Filter 2nd Stage

Filter type	Correction Coefficient
F1a, F1b	4076/4096
F2	4119/4096
F3a, F3b	4073/4096
F4	4089/4096
2nd stage not used (F5, F6, F7)	4096/4096

Correction formula

$$\text{Corrected A/D conversion result} = \text{A/D conversion result} \times \text{Correction coefficient}$$

35.6.11 A/D Conversion Monitoring Function Using Monitor Pins

The A/D conversion of the DSADCn can be monitored by the DSADENDn. The DSADENDn pins directly output A/D conversion result of DSADCn. CLBACT is not output. For the pin settings, see **Section 2, Pin Function**.

Figure 35.12 shows an example of A/D conversion monitor pin output.

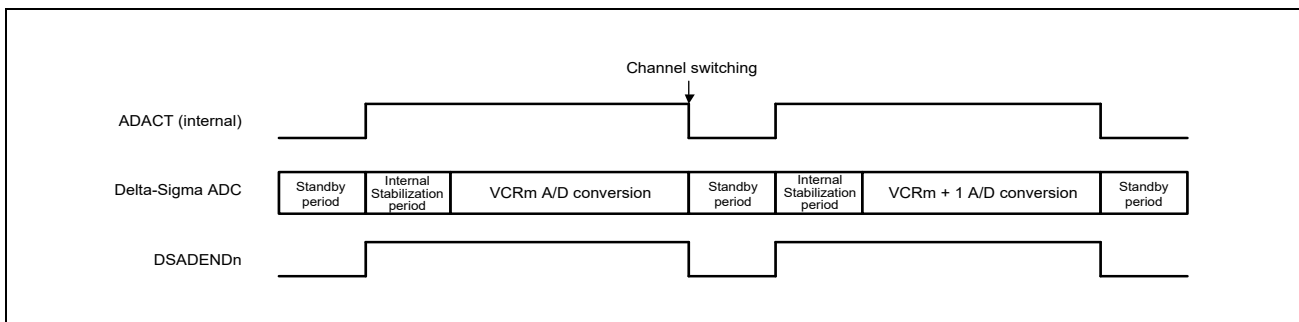


Figure 35.12 Example of A/D Conversion Monitor Pin Output

35.6.12 DMA Request Source

The Delta-Sigma ADC generates DMA request sources (DSADIn) to activate the DMAC (without generating an interrupt to the INTC). When the ADIE in DSADCnVCRj is set to 1, the DSADIn is output at the end of an A/D conversion. When the ADIE in DSADCnADTCR is cleared to 0, the DSADIn output is disabled even at the end of an A/D conversion.

For DMAC settings, see **Section 7, sDMA Controller (sDMAC)** and **Section 8, DTS Controller**.

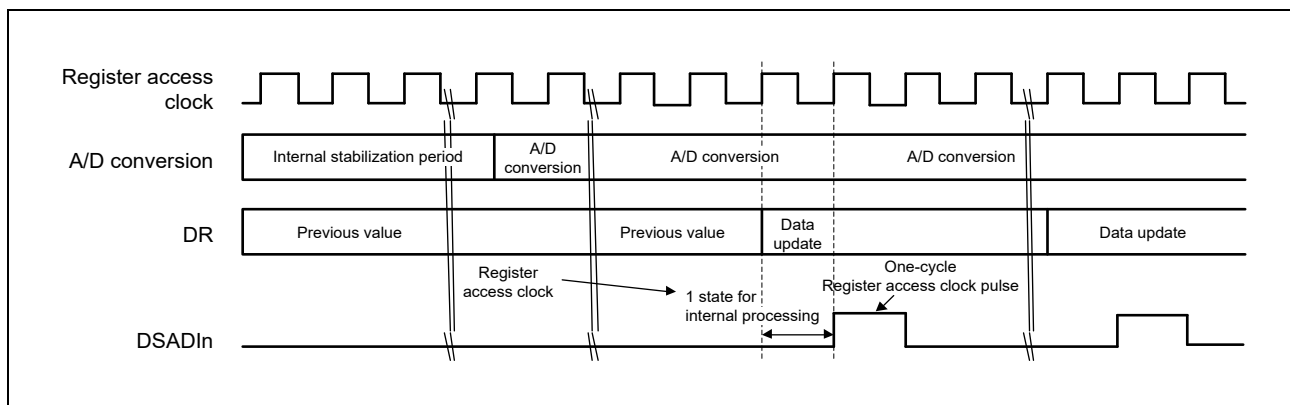


Figure 35.13 Example of DMA Request Source

35.6.13 AD Error Interrupt Request and AD Parity Error Interrupt Request

The Delta-Sigma ADC can generate an AD error interrupt request (DSADEn) to the INTC and an AD parity error interrupt request (DSADPEn) to the ECM (Error Control Module). Any of the error sources for which ULEIE in DSADCnVCRj or IDEIE or OWEIE in DSADCnSFTCR is set to 1 are output as DSADEn. Error sources DSADEn for which ULEIE in DSADCnVCRj or IDEIE or OWEIE in DSADCnSFTCR are set to 0 can be disabled. When PEIE in DSADCnSFTCR is set to 1, DSADPEn is enabled. When PEIE is set to 0, DSADPEn is disabled.

Table 35.40 Correspondence between Interrupt Sources and Interrupt Signals

Status	Interrupt Signal	Interrupt Enable
LLE	DSADEn	DSADCnVCRj.ULEIE and DSADCnVCRj.VCLLME
ULE		DSADCnVCRj.ULEIE and DSADCnVCRj.VCULME
IDE		IDEIE
OWE		OWEIE
PE	DSADPEn	PEIE

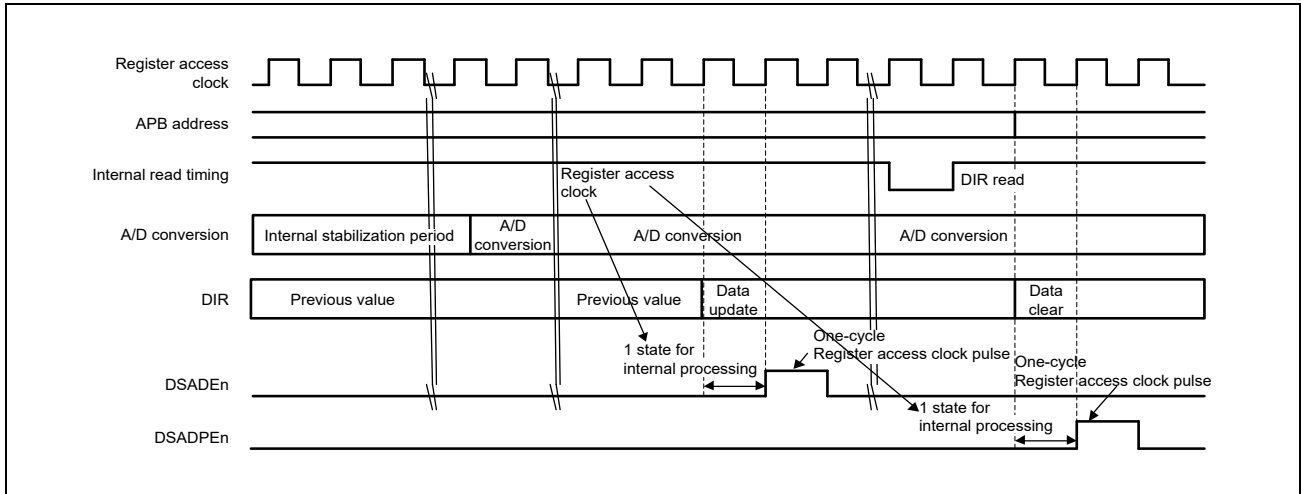


Figure 35.14 Example of an AD Error Interrupt and an AD Parity Error Interrupt

35.6.13.1 Upper-Limit/Lower-Limit Errors

These errors indicate that the A/D conversion result has exceeded the set value.

When the comparison result between the A/D conversion result and the upper-limit/lower-limit table register value (ULTBR0 to ULTBR3.ULMTB[15:0], LLMTB[15:0]) by setting VCULME and VCLLME of DSADCnVCRj is as follows, it is determined as an error.

A/D conversion result > ULMTB[15:0] (When upper-limit table is effective: DSADCnVCRj.VCULME = 1)

A/D conversion result < LLMTB[15:0] (When lower-limit table is effective: DSADCnVCRj.VCLLME = 1)

Detection of these errors is made each time an A/D conversion result is stored.

35.6.13.2 Overwrite Error

This error indicates that a new A/D conversion result has been overwritten to the data register before the data supplementary information register j (DSADCnDIRj) is read.

When the write flag (DSADCnDIRj.WFLG) in the data supplementary information register j is 1 and a new A/D conversion result is stored in the relevant data supplementary information register j (DSADCnDIRj), an overwrite error is detected.

35.6.13.3 ID Error

This error indicates that the physical channel set by the virtual channel register j (DSADCnVCRj) is different from the physical channel in which the A/D conversion was actually performed. In other words, this error indicates the occurrence of a malfunction of the analog switch in the I/O buffer.

The ID error detection result is stored in the ID Error bit in the data supplementary information register j (DSADCnDIRj) corresponding to the virtual channel (DSADCnVCRj) in which an ID error was detected.

The level of the feedback signal from the I/O buffer is checked.

The general control bit (DSADCnVCRj.GCTRL[3:0]) value in the virtual channel register j is compared with the I/O buffer number that is actually activated. If these values do not match, an ID error is detected.

Because the I/O buffer is controlled in a one-hot manner, this error is detected also when the feedback signal is not a one-hot signal.

Either of the following are detected.

- (1) Whether the specified physical channel number does not match the activated I/O buffer number
- (2) Whether differential inputs of two channels or more or single-ended input I/O buffers are simultaneously activated during an A/D conversion (one-hot error)

35.6.14 Boundary Flag Generating Signal Output Function

The boundary flag is a level signal that shows whether the input voltage is within the specified voltage range.

The Delta-Sigma ADC outputs upper-limit/lower-limit error pulse signals (DSADUEn, DSADLEn) and a virtual channel conversion end pulse signal (DSADCnDSADVCIj) as boundary flag generating signals.

When A/D conversion in a virtual channel that corresponds has ended, a virtual channel conversion end pulse (DSADCnDSADVCIj) is always output in spite of setting for ADIE in DSADCnVCRj.

When setting VCULME in DSADCnVCRj as 1, if upper limit error has been detected, the upper limit error pulse (DSADUEn) is output.

When setting VCLLME in DSADCnVCRj as 1, if lower limit error was detected, the lower limit error pulse (DSADLEn) is output.

A boundary flag is generated by ACFG (ADC Boundary Flag Generator). When using the boundary flag, set VCULME and VCLLME in DSADCnVCRj to 1. For ACFG, see **Section 34, Analog to Digital Converter (ADCH)**.

On the other hand, after every unit is merged*¹, a virtual channel conversion end pulse (DSADCnDSADVCIj)*² is output to the GTM*³ as a conversion end pulse (DSADCnDSADCEP).

Note 1. The merging method is OR.

Note 2. DSADCnDSADVCIj is output irrespective of the settings for VCULME and VCLLME in DSADCnVCRj.

Note 3. DSADCnDSADCEP is chosen by the PIC (Peripheral Interconnection), and those are forwarded to the GTM. For details, see **Section 33, Peripheral Interconnection (PIC)**.

35.6.15 Timestamp Function

The lapsed time from the last results output is stored in the DSADCnTSVAL register at the rising edge timing of the DSADRGTn from timer (ATU, GTM) as timestamp information. This function allows measurement of the time difference between the DSADRGTn and the input sampling time. When RDMA in DSADCnUCR is set to 1, the DSADIn is output only while the DSADRGTn is asserted.

The DSADIn output is valid only when DSADRGTn from a timer (ATU, GTM) is asserted.

To continuously toggle DSADRGTn while inputting the operation clock, maintain a pulse width (the Low width and the High width) as wide as at least 2 PCLK clocks (register access clock).

Maintain the rising edge interval of DSADRGTn more than 10 clocks (register access clock).

When the rising edge interval of DSADRGTn is below 10 clocks (register access clock), the second and the following time stamp updates at the rising edge timing of DSADRGTn is waited until the former time stamp updating is complete. Therefore, DMAC cannot read updated time stamp values sometimes.

Figure 35.15 shows the operation of the timestamp function.

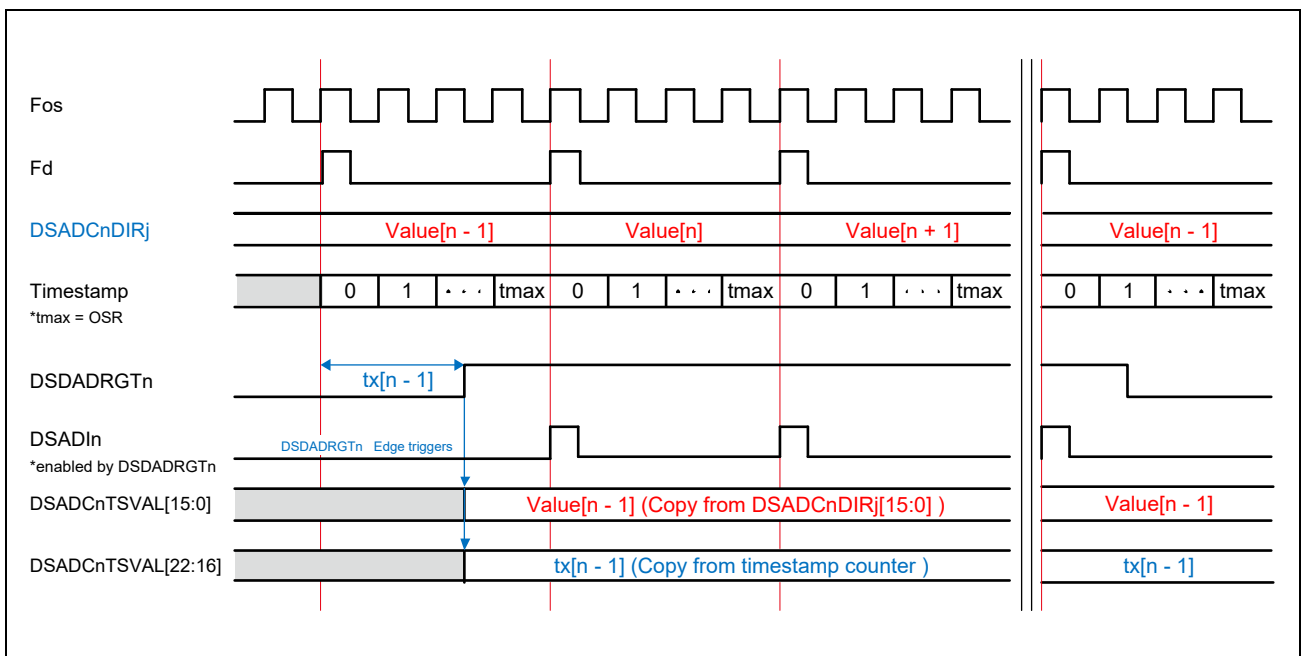


Figure 35.15 Example of the Timestamp Function Operation (RDMA = 1)

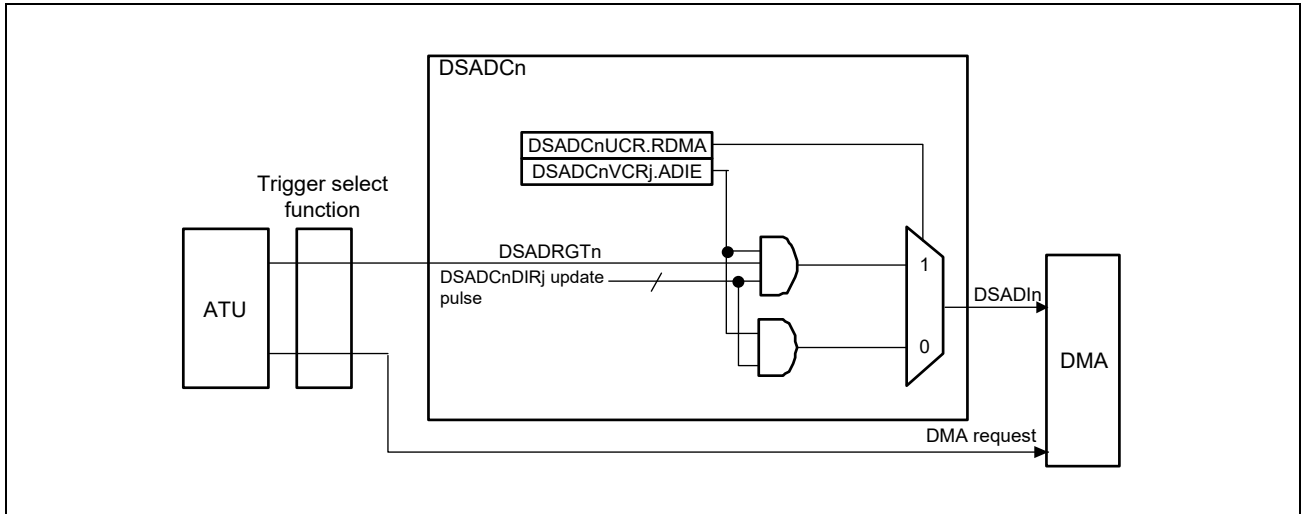


Figure 35.16 DMA Request by DSADRGTn from the ATU

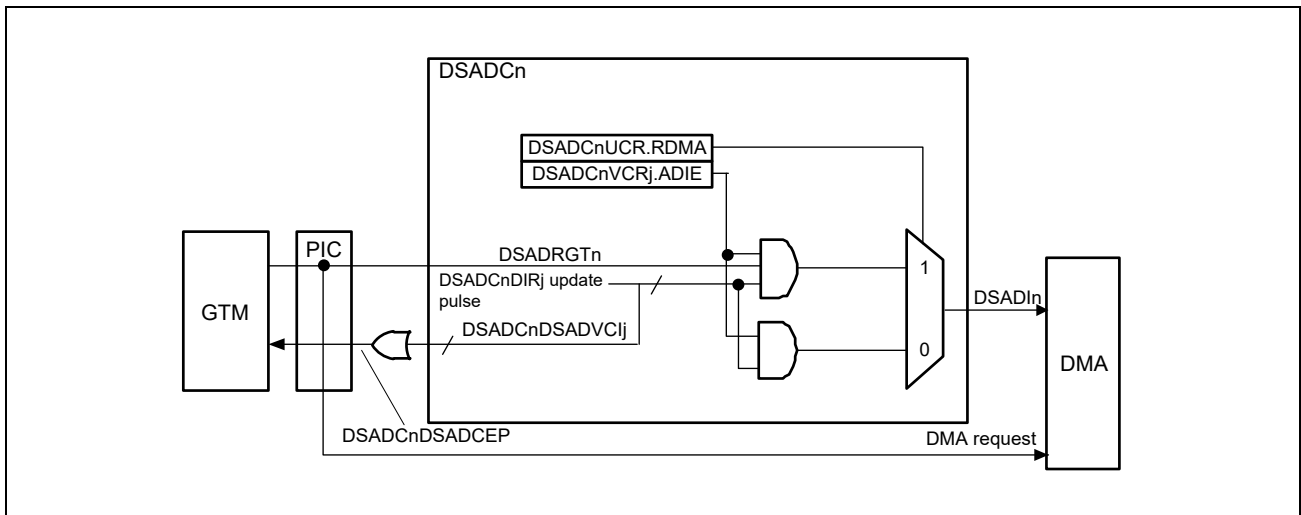


Figure 35.17 DMA Request by DSADRGTn from the GTM

35.6.16 Wiring-Break Detection Function

The wiring-break detection function is used to detect wiring-breaks of DSANn0P/DSANn0N to DSANn3P/DSANn3N.

If a wiring-break is present, the conversion results attenuates to approximately 0 V, and an abnormal value is detected in the conversion results by repeating the A/D conversion several times.

This can be determined as detection of wiring-break.

It is not possible to use the wiring-break detection function simultaneously with normal A/D conversion.

Use a wiring-break detection function before normal A/D conversion is started or between normal A/D conversions.

[Feature]

(1) Users can select desired physical channels for which wiring-break is to be detected by setting virtual channels.

[Recommended flow]

- (1) Make settings according to the initial settings (**Figure 35.4**).
- (2) Set CNVCLS[1:0] in the virtual channel register j (DSADCnVCRj) to 0_H and set GCTRL[3:0] to select the desired channels.
- (3) Set ODE in the AD control register (DSADCADGCR) to 1.
- (4) Set ADST in the AD start control register (DSADCnADSTCR) to 1, and then perform A/D conversion.
- (5) When the read A/D conversion results are attenuated to approximately 0V after repeating A/D conversion several times, judge that wiring-break occurred.
- (6) Repeat steps (4) to (5) for all channels which have DSADCnVCRj set.

35.6.17 Self-Diagnosis

35.6.17.1 Self-Diagnosis of Wiring-Break Detection Function

The self-diagnosis function of the wiring-break detection function is used to check that the wiring-break detection function is working normally.

Enable the I/O buffer self-diagnosis and wiring detection of DSANn0P/DSANn0N to DSANn3P/DSANn3N by setting A/D control register (DSADCADGCR), and then perform A/D conversion.

The conversion results attenuates to approximately 0 V, and an abnormal value is detected in the conversion results by repeating the A/D conversion several times.

This means that the wiring-break detection function is normally processed.

[Feature]

(1) Users can select desired physical channels for which self-diagnosis of wiring-break detection function is performed by setting virtual channels.

[Recommended flow]

- (1) Make settings according to the initial settings (**Figure 35.4**).
- (2) Set CNVCLS[1:0] in the virtual channel register j (DSADCnVCRj) to 0_H and set GCTRL[3:0] to select the desired channels.
- (3) Set both ODE and ODDE in the A/D control register (DSADCADGCR) to 1.
- (4) Set ADST in the A/D start control register (DSADCnADSTCR) to 1, and then perform the A/D conversion.
- (5) When the read A/D conversion results are attenuated to approximately 0V after repeating A/D conversion several times, judge that self-diagnosis of wiring-break detection function is working normally.
- (6) Repeat steps (4) to (5) for all channels which DSADCnVCRj is set.

35.6.17.2 A/D Self-Diagnosis Function

A/D self-diagnosis is used to check that the A/D conversion works normally.

If the expectation value does not match the A/D conversion results, there is a possibility that the A/D converter is broken.

The voltage value is set by setting CNVCLS[1:0] of virtual channel register j to 3_H and selecting from GCTRL [3:0].

It is possible that A/D conversion for $ADSVREFH \times 1$, $ADSVREFH \times 1/2$, $-ADSVREFH \times 1/2$, $-ADSVREFH \times 1$ or $ADSVREFH \times 0$ is performed.

When using A/D self-diagnosis, set gain $\times 1$ or gain $\times 2$ for high accuracy mode, and set gain $\times 1$, gain $\times 2$ or gain $\times 4$ for high impedance mode.

Settings other than above are prohibited.

[Setting]

- (1) Make settings according to the initial settings (**Figure 35.4**).
- (2) Set CNVCLS[1:0] in the virtual channel register j (DSADCnVCRj) to 3_H and set GCTRL[3:0] to select the desired A/D self-diagnosis voltage level.
- (3) Make other appropriate settings (**Figure 35.4**).
- (4) Set ADST in the A/D start control register (DSADCnADSTCR) to 1, and then perform the A/D conversion.

35.6.17.3 Pin Level Self-Diagnosis Function

Pin level self-diagnosis is the function to diagnose abnormal analog input by applying different voltages and performing A/D conversion for each physical channel.

When using pin level self-diagnosis, set gain $\times 1$.

Settings other than above are prohibited.

[Analog Input Process]

I/O Buffer to A/D Converter

[Setting]

Make the settings for A/D control register (DSADCADGCR), virtual channel register j (DSADCnVCRj), pin level self-diagnosis control register (DSADCTDCR) and pin level self-diagnosis level register (DSADCnTDLVR) according to the initial settings.

- (1) DSADCADGCR

ODDE	ODE
1	0

- (2) DSADCTDCR

TDE
1

(3) DSADC_nVCR_j

VCR Number	CNVCLS[1:0]	GCTRL[3:0]
N	0 _H , 1 _H , 2 _H	Physical Channel Number

(4) DSADC_nTDLVR

AN0PLV/AN0NLV AN1PLV/AN1NLV AN2PLV/AN2NLV AN3PLV/AN3NLV	Injecting Voltage of Physical Channel
0	ADSVSS
1	ADSVCC

Perform A/D conversion according to the A/D conversion flow.

NOTE

Set ODDE of the A/D control register (DSADCADGCR) to 1 before setting TDE of the pin level self-diagnosis control register (DSADCTDCR) to 1.

Do not use the pin level self-diagnosis function and wiring-break detection function at the same time.

35.6.18 DFE Control

Entry to the DFE or GTM is enabled by setting the DFENT bit in the DSADCnVCRj register to 1. This A/D has two DFE channels and uses one of them. A DFE channel to use can be selected for each A/D unit according to the settings of the DFES bit in the DSADCnUCR register.

35.7 Usage Notes

35.7.1 Notes on Using Analog Input Pins

Do not perform an A/D conversion on the value of the same analog pin simultaneously by the Delta-Sigma ADC and the SAR-ADC. Also, do not allow the Delta-Sigma ADC to perform an A/D conversion on the value of an analog pin currently selected for digital input. This may degrade A/D accuracy.

When the Delta-Sigma ADC is used with a single-ended input, and only the analog input pins on the P side (or N side) of a differential input are used, the analog input pins on the N side (or P side) can be used for other ADC inputs.

35.7.2 Module Standby Function

The Delta-Sigma ADC has a module standby function.

Module standby stops the clock supply to DSADC00 and DSADC10 by setting MSR_DAD.MS_DAD00_10.

Module standby stops the clock supply to DSADC20 and DSADC12 by setting MSR_DAD.MS_DAD20_12.

Module standby stops the clock supply to DSADC13 and DSADC11 by setting MSR_DAD.MS_DAD13_11.

Module standby stops clock supply to DSADC (common). For the clock supply stop condition for the DSADC (common), see **Table 35.41**.

Table 35.41 Clock Supply Stop Condition of DSADC (common) by Module Standby

States	MSR_DAD. MS_DAD00_10	MSR_DAD. MS_DAD20_12	MSR_DAD. MS_DAD13_11	MSR_CAD. MS_CAD
Clock Supply	0	0 or 1	0 or 1	0 or 1
Clock Supply	1	0	0 or 1	0 or 1
Clock Supply	1	1	0	0 or 1
Clock Supply	1	1	1	0
Clock Supply Stop	1	1	1	1

To prevent a malfunction, do not set the module standby mode during an A/D conversion, calibration or while an A/D conversion start/end trigger is enabled. Before setting the module standby mode, set the related channels as follows.

Also, set module standby mode after maintaining the following state during 10 states of the register access clock:

- DSADCnADSR.ADACT = 0, and
- DSADCnADSR.CLBACT = 0, and
- DSADCnADTCR.ENDTRGE = 0, and
- DSADCnADTCR.STTRGE = 0

Section 36 Cyclic Analog to Digital Converter (CADC)

This section contains a description of the Cyclic A/D Converter (CADC).

The first part of this section describes all of the features specific to this product, such as the number of units and register base addresses. The remainder of the section describes the functions and registers of the CADC.

36.1 Features

36.1.1 Number of Units and Channels

This microcontroller has the following number of CADC units.

Table 36.1 Number of Units

Product Name	RH850/E2x-FCC1, E2M	
	292 Pins	373 Pins
Number of CADC Common Units	1	1
Number of CADC Channel Units	1	1
Name of CADC Units	CADC CADC _n (n = 00)	

A CADC_n unit has the same number of physical channels as the number of A/D input pins. The numbers of channels of individual products are listed below.

Table 36.2 Unit Configurations and Physical Channels

Unit Name (Number of Channels)	RH850/E2x-FCC1, E2M	
	292 Pins	373 Pins
CADC _n		
CADC00	8	8

Table 36.3 Unit Configurations and Virtual Channels

Unit Name (Number of Channels)	RH850/E2x-FCC1, E2M	
	292 Pins	373 Pins
CADC _n		
CADC00	8	8

Table 36.4 Indices

Index	Description
n	Throughout this section, the individual CADC units are identified by the index "n" (n = 00); For example, CADC _n CCR indicates the channel control register.
x	Throughout this section, the individual analog input attribute at differential input are identified by the index "x" (x = P, N); for example, CAN _n x.
j	The number of a virtual channel.

36.1.2 Register Base Addresses

The CADC and CADCn base addresses are listed in the following table.

The CADC and CADCn register addresses are given as offsets from the base address in general.

Table 36.5 Register Base Addresses

Base Address Name	Base Address	Bus Group
<CADC_base>	FFF4 0000 _H	Peripheral Group 7
<CADCn_base>	FFF4 1000 _H	Peripheral Group 7

36.1.3 Clock Supply

The CADC and CADCn clock supplies are shown in the following table.

Table 36.6 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
CADC	ADCLK	CLK_HSB
	Register access clock, PCLK	CLK_LSB
CADCn	ADCLK	CLK_HSB
	Register access clock, PCLK	CLK_LSB

36.1.4 Interrupt Requests

The CADCn interrupt requests are listed in the following table.

Table 36.7 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number (DMA or DTS)	Other Trigger Signals
CADCn				
CADEn	A/D error interrupt	*1	—	—
	Upper/lower limit error interrupt			
CADIn	A/D conversion end interrupt	—	*1	—

Note 1. CADEn and CADIn are chosen by the AIR (ADC Interrupt Router), and those are forwarded to the INTC and DMAC.

For Interrupt Number and DMA Trigger Numbers, see **Section 34, Analog to Digital Converter (ADCH)**.

36.1.5 Reset Sources

The CADC and CADCn reset sources are listed in the following table. The CADC and CADCn are initialized by these reset sources.

Table 36.8 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
CADC	All registers	✓	✓	✓	✓	—	—
CADCn	All registers	✓	✓	✓	✓	—	—

36.1.6 External Input/Output Signals

The External Input/Output signals of the CADC_n are listed below.

Table 36.9 CADC External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
CADC _n Common		
ADSVCC	Power supply pin for the analog part	ADSVCC
ADSVSS	Ground pin for the analog part	ADSVSS
ADSVREFH	Upper reference voltage pin for the analog part	ADSVREFH
ADSVREFL	Lower reference voltage pin for the analog part	ADSVREFL

Table 36.10 CADC00 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
CAN000P	Analog input pin (for single-ended/differential P)	CAN000P
CAN000N	Analog input pin (for single-ended/differential N)	CAN000N
CAN001P	Analog input pin (for single-ended/differential P)	CAN001P
CAN001N	Analog input pin (for single-ended/differential N)	CAN001N
CAN002P	Analog input pin (for single-ended/differential P)	CAN002P
CAN002N	Analog input pin (for single-ended/differential N)	CAN002N
CAN003P	Analog input pin (for single-ended/differential P)	CAN003P
CAN003N	Analog input pin (for single-ended/differential N)	CAN003N
CADTRG00	External trigger pin	CADTRG00
CADEND00	A/D conversion timing monitor pin	CADEND00

36.2 Overview

36.2.1 Function Overview

The following describes features (functional overview) of the CADC.

- A/D conversion method: Cyclic method
- Configuration of analog input channels
 - 1 module, 8 analog input channels in total.
 - CADC00: 8 single-ended inputs (4 differential inputs).

NOTES

1. One differential input can be used as two single-ended inputs.
2. All analog inputs are multiplexed with ADCH analog inputs.

- Input type: Single-ended input and differential input
Single-ended input or differential input can be selected for each virtual channels.

In single-ended mode:

The A/D conversion value format of unsigned 16-bit resolution is generated when ADSVREFL is chosen as the common mode voltage.

Range of the A/D conversion value: 0000_H to FFFF_H

The A/D conversion value format of signed 15-bit resolution (15-bit value preceded by the sign bit) is generated when ADSVREFH/2 is chosen as the common mode voltage.

Range of the A/D conversion value: C000_H to 3FFF_H

In differential mode:

The A/D conversion value format of signed 15-bit resolution (15-bit value preceded by the sign bit) is generated.

Range of the A/D conversion value: 8000_H to 7FFF_H

- Input gain function (PGA): × 1 only.
Sampling rate: 1 Msps (Fos (Over sampling rate) is 2Msps)
- Pass band: 500 kHz
- Virtual channels
See **Section 36.1.1, Number of Units and Channels**.
- Calibration function
The CADC has a calibration function, and the execution of the calibration routine realize a high accuracy A/D conversion.
- Data register
A data register is contained in the data supplementary information register j.
- A/D conversion start trigger
Each CADC can start an A/D conversion with software, each timer trigger, or an external trigger (CADTRGn).
- A/D conversion end trigger
Each CADC can terminate an A/D conversion with software, each timer trigger, or an external trigger (CADTRGn).
- Entry to the digital filter engine, and the Generic Timer Module
A/D converted values can be entered directly into the digital filter engine (DFE), or Generic Timer Module (GTM).

Whether to enable or disable DFE entry and a TAG to determine channels for entry can be set for each virtual channel.

For details of DFE, see **Section 37, Digital Filter Engine (DFE)**.

For details of GTM, see **Section 31, Generic Timer Module (GTM)**.

- DMA transfer based on A/D conversion end timing

Each time an A/D conversion ends, each CADC outputs a DMA request on completion of an A/D conversion (CADIn), which can generate DMA request sources (CADIn) to the DMAC or activate the DMAC.

The output of the CADIn is capable of being limited by a read-gate (CADRG_{Tn}) from a timer (ATU GTM).

For details of the ATU, see **Section 30, ATU-V**.

For details of the GTM, see **Section 31, Generic Timer Module (GTM)**.

- Boundary flag output function

It is possible to output a boundary flag generating signal (CADUE_n, CADLE_n and CADC_nVCI_j) to the ABFG (ADC Boundary Flag Generator).

For details of the ABFG, see **Section 34, Analog to Digital Converter (ADCH)**.

- AD error interrupt request and AD parity error notification

Each CADC can generate an AD error interrupt request (CADE_n) to the INTC and an AD parity error notification (CADPE_n) to the ECM (error control module).

- Settable analog conversion voltage range

An analog conversion voltage range can be set using the ADSVREFH pin and the ADSVREFL pin. When performing an A/D conversion by using the single-ended input, either ADSVREFL or ADSVREFH/2 is selectable as a common voltage.

- A/D conversion monitor output

A/D conversion timing can be output to the A/D conversion monitor output pin (CADEND_n).

- A wide range of safety functions

A variety of safety functions are provided, including self-diagnosis, pin-level self-diagnosis, wiring-break detection, normality check for analog selection, upper-limit/lower-limit check for data registers, data register parity check for data registers, overwrite check for data registers, and read and clear function for data registers.

36.2.2 Block Diagram

The block diagram of the CADC is shown in **Figure 36.1**.

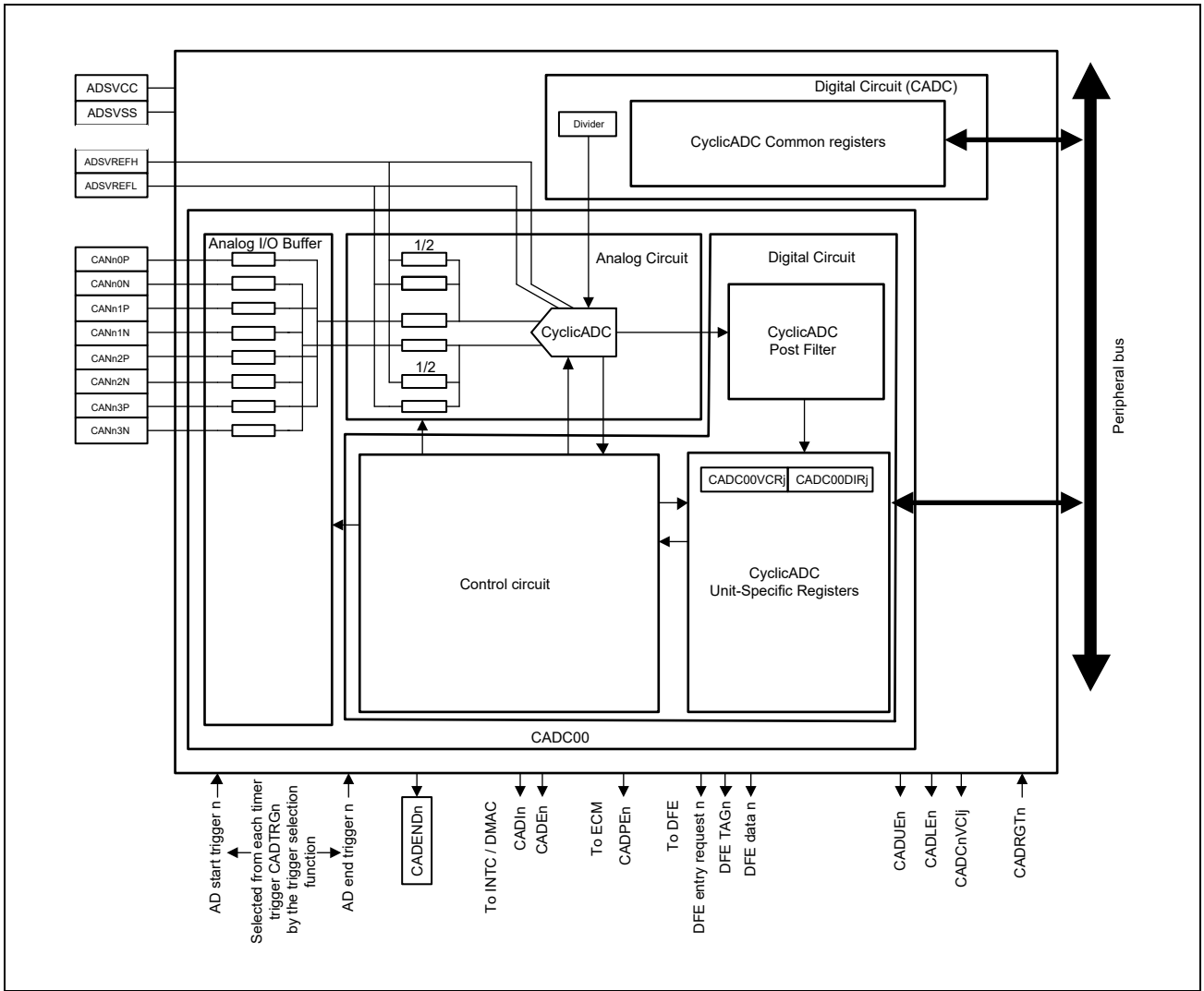


Figure 36.1 Block Diagram of CADC

36.3 Registers

36.3.1 List of Registers

The CADC registers are listed in the following table.

For detail about <CADC_base> and <CADCn_base>, see **Section 36.1.2, Register Base Address**.

Table 36.11 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
CADC	AD Control Register	CADCADGCR	<CADC_base> + 04 _H	8	—
CADC	Pin Level Self-Diagnosis Control Register	CADCTDCR	<CADC_base> + 0C _H	8	—
CADCn	Virtual Channel Register 0	CADCnVCR0	<CADCn_base> + 00 _H	8, 16, 32	—
CADCn	Virtual Channel Register 1	CADCnVCR1	<CADCn_base> + 04 _H	8, 16, 32	—
CADCn	Virtual Channel Register 2	CADCnVCR2	<CADCn_base> + 08 _H	8, 16, 32	—
CADCn	Virtual Channel Register 3	CADCnVCR3	<CADCn_base> + 0C _H	8, 16, 32	—
CADCn	Virtual Channel Register 4	CADCnVCR4	<CADCn_base> + 10 _H	8, 16, 32	—
CADCn	Virtual Channel Register 5	CADCnVCR5	<CADCn_base> + 14 _H	8, 16, 32	—
CADCn	Virtual Channel Register 6	CADCnVCR6	<CADCn_base> + 18 _H	8, 16, 32	—
CADCn	Virtual Channel Register 7	CADCnVCR7	<CADCn_base> + 1C _H	8, 16, 32	—
CADCn	Data Supplementary Information Register 0	CADCnDIR0	<CADCn_base> + 20 _H	16 ^{*1} , 32	—
CADCn	Data Supplementary Information Register 1	CADCnDIR1	<CADCn_base> + 24 _H	16 ^{*1} , 32	—
CADCn	Data Supplementary Information Register 2	CADCnDIR2	<CADCn_base> + 28 _H	16 ^{*1} , 32	—
CADCn	Data Supplementary Information Register 3	CADCnDIR3	<CADCn_base> + 2C _H	16 ^{*1} , 32	—
CADCn	Data Supplementary Information Register 4	CADCnDIR4	<CADCn_base> + 30 _H	16 ^{*1} , 32	—
CADCn	Data Supplementary Information Register 5	CADCnDIR5	<CADCn_base> + 34 _H	16 ^{*1} , 32	—
CADCn	Data Supplementary Information Register 6	CADCnDIR6	<CADCn_base> + 38 _H	16 ^{*1} , 32	—
CADCn	Data Supplementary Information Register 7	CADCnDIR7	<CADCn_base> + 3C _H	16 ^{*1} , 32	—
CADCn	AD Start Control Register	CADCnADSTCR	<CADCn_base> + 40 _H	8	—
CADCn	AD Stop Control Register	CADCnADENDCR	<CADCn_base> + 44 _H	8	—
CADCn	Calibration Start Control Register	CADCnCLBSTCR	<CADCn_base> + 48 _H	8	—
CADCn	Calibration Stop Control Register	CADCnCLBEDCR	<CADCn_base> + 4C _H	8	—
CADCn	AD Conversion Trigger Control Register	CADCnADTCR	<CADCn_base> + 50 _H	8	—
CADCn	Unit Control Register	CADCnUCR	<CADCn_base> + 54 _H	8, 16, 32	—
CADCn	Virtual Channel Pointer Register	CADCnVCPTRR	<CADCn_base> + 58 _H	8	—
CADCn	AD Conversion Status Register	CADCnADSR	<CADCn_base> + 60 _H	8	—
CADCn	Unit Data Pointer Register	CADCnUDPTRR	<CADCn_base> + 64 _H	8	—
CADCn	Unit Data Supplementary Information Register	CADCnUDIR	<CADCn_base> + 68 _H	32	—
CADCn	Sampling Rate Control Register	CADCnSMPCR	<CADCn_base> + 6C _H	8, 16, 32	—
CADCn	Safety Control Register	CADCnSFTCR	<CADCn_base> + 70 _H	8	—
CADCn	Error Clear Register	CADCnECR	<CADCn_base> + 74 _H	8	—
CADCn	Error Register	CADCnER	<CADCn_base> + 78 _H	32	—
CADCn	Pin Level Self-Diagnosis Level Register	CADCnTDLVR	<CADCn_base> + 7C _H	8	—
CADCn	Upper-Limit/Lower-Limit Table Register 0	CADCnULTBR0	<CADCn_base> + 80 _H	16, 32	—
CADCn	Upper-Limit/Lower-Limit Table Register 1	CADCnULTBR1	<CADCn_base> + 84 _H	16, 32	—
CADCn	Upper-Limit/Lower-Limit Table Register 2	CADCnULTBR2	<CADCn_base> + 88 _H	16, 32	—
CADCn	Upper-Limit/Lower-Limit Table Register 3	CADCnULTBR3	<CADCn_base> + 8C _H	16, 32	—

Note 1. When reading CADCnDIRj in 16-bit units, see **Section 36.5.2, CADCnDIRj — Data Supplementary Information Register j**.

36.4 Cyclic ADC Common Registers

This section describes the registers shared by all of the Cyclic ADC units.

36.4.1 CADCADGCR — AD Control Register

CADCADGCR is an 8-bit read/write register to totally control A/D conversion.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	CADCODDE	CADCODE	—	—	—	CADCUNSND
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R/W

Table 36.12 CADCADGCR Register Contents

Bit Position	Bit Name	Function
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	CADCODDE	Wiring-break Detection Function Self-Diagnosis Enable 0: Self-diagnosis of the wiring-break detection function is disabled. 1: Self-diagnosis of the wiring-break detection function is enabled. When CADCODDE is set to 1, wiring-break detection function self-diagnosis and pin level self-diagnosis are enabled. When doing pin level self-diagnosis, set CADCODDE to 1 before setting CADCTDE to 1. If CADCTDE is set to 1 with CADCODDE = 0, the self-diagnosis pin level voltage is output to the chip pin, which might affect the external circuit of the connected device.
4	CADCODE	Wiring-break Detection Enable 0: Wiring-break detection is disabled. 1: Wiring-break detection is enabled. When CADCODE is set to 1, wiring-break detection is enabled for all analog pins. Setting this bit in 1 simultaneously with pin level self-diagnosis (CADCTDE=1) is prohibited. When setting CADCADGCR.CADCODE as 1, if CADCnVCRj.CADCnCNVCLS[1:0] value is set to anything but 0 _H , A/D conversion is prohibited.
3 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CADCUNSND	Unsigned Conversion Result Output Enable When unsigned output is enabled in this register, the A/D conversion result (when ADSVREFL is selected for the single-ended common voltage) can be output as an unsigned value. 0: A/D conversion result is output as a signed value. 1: When the single-ended common voltage = ADSVREFL, A/D conversion result is output as an unsigned value.*1 For details about the difference in the data format for this bit, see Section 36.5.2, CADCnDIRj — Data Supplementary Information Register j.

Note 1. When the calculation result after correction is a negative value, an A/D conversion result of 0 is output.

NOTE

To prevent a malfunction, set CADCADGCR when CADCnADACT and CADCnCLBACT of the Cyclic ADC = 0 (before starting A/D conversion) and CADCnSTTRGE of all Cyclic ADCs = 0.

36.4.2 CADCTDCR — Pin Level Self-Diagnosis Control Register

CADCTDCR is an 8-bit read/write register to control pin level self-diagnosis.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CADCTDE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 36.13 CADCTDCR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CADCTDE	<p>Pin Level Self-Diagnosis Enable</p> <p>0: Pin level self-diagnosis is disabled. 1: Pin level self-diagnosis is enabled.</p> <p>When CADCTDE is set to 1, all analog pins are disconnected from the analog I/O buffer. When CADCTDE is set to 0, all analog pins are connected to the analog I/O buffer.</p> <p>When CADCTDE = 1, analog pins CANn0x to CANn3x are fixed to the level specified by CADCnTDLVR.AN0xLV to CADCnTDLVR.AN3xLV. By performing A/D conversion in this state and checking the A/D converted value, the path between the analog pins and Cyclic ADC can be diagnosed.</p> <p>Do not set this bit as 1 simultaneously with CADCODE = 1 (wiring-break detection enable) and CADCCNVCLS = 3_H (A/D self-diagnosis).</p> <p>When doing pin level self-diagnosis, set CADCODDE to 1 before setting CADCTDE to 1.</p> <p>If CADCTDE is set to 1 with CADCODDE = 0, the self-diagnosis pin level voltage is output to the chip pin, which might affect the external circuit of the connected device.</p>

NOTE

To prevent a malfunction, set CADCTDCR when CADCnADACT and CADCnCLBACT of the Cyclic ADC = 0 (before starting A/D conversion) and CADCnSTTRGE of all Cyclic ADCs = 0.

36.5 Cyclic ADC Unit-Specific Registers

This section describes the registers provided to control each individual Cyclic ADC unit controller.

36.5.1 CADCnVCRj — Virtual Channel Register j

CADCnVCRj is a 32-bit read/write register to control virtual channels.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	CADCnVCULME	CADCnVCLLME	CADCnVCULLMTBS[1:0]		—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CADCnADIE	CADCnULEIE	—	CADCnDFENT	CADCnDFTAG[3:0]				—	—	CADCnCNVCLS[1:0]		CADCnGCTRL[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 36.14 CADCnVCRj Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 28	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
27	CADCnVCULME	Virtual Channel Upper-Limit Excess Notification Enable 0: If the A/D conversion result is greater than the upper limit of the virtual channel, it is not notified. 1: If the A/D conversion result is greater than the upper limit of the virtual channel, it is notified.
26	CADCnVCLLME	Virtual Channel Lower-Limit Excess Notification Enable 0: If the A/D conversion result is less than lower limit of the virtual channel, it is not notified. 1: If the A/D conversion result is less than lower limit of the virtual channel, it is notified.

Table 36.14 CADcNVCRj Register Contents (2/3)

Bit Position	Bit Name	Function
25, 24	CADcNVCULLMTBS [1:0]	Virtual Channel Upper-Limit/Lower-Limit Table Register Select These bits select the upper-limit/lower-limit table register to be compared. 0 _H : Upper-limit/lower-limit values are checked by CADcNULTBR0. [Initial value] 1 _H : Upper-limit/lower-limit values are checked by CADcNULTBR1. 2 _H : Upper-limit/lower-limit values are checked by CADcNULTBR2. 3 _H : Upper-limit/lower-limit values are checked by CADcNULTBR3. When the A/D converted value is stored in the CADcNDR, upper-limit/lower-limit values are checked by using the upper-limit value/lower-limit value table.
23 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15	CADcNADIE	Conversion End Interrupt Enable 0: CADcN is not output when the CADcNDRj conversion result is output. 1: CADcN is output when the CADcNDRj conversion result is output.
14	CADcNULEIE	Upper-Limit/Lower-Limit Error Interrupt Enable 0: CADcN is not output upon an upper-limit or lower-limit error of CADcNDRj. 1: CADcN is output upon an upper-limit or lower-limit error of CADcNDRj.
13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12	CADcNDFENT	DFE Entry 0: Entry is not performed. 1: Entry is performed. This bit selects whether or not to perform entry to the DFE (digital filter engine).
11 to 8	CADcNDFTAG[3:0]	DFE-TAG These bits are used to perform entry to a channel of the DFE for which the TAG is identical to the setting of CADcNDFTAG[3:0]. If the TAG matches the setting of CADcNDFTAG[3:0] for multiple channels, entry to multiple DFE channels is performed.
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5, 4	CADcNCNVCLS[1:0]	Conversion Type 0 _H : Single-ended input, common voltage = ADVREFL 1 _H : Single-ended input, common voltage = ADVREFH/2 2 _H : Differential input 3 _H : Self-diagnosis

Table 36.14 CADCnVCRj Register Contents (3/3)

Bit Position	Bit Name	Function
3 to 0	CADCnGCTRL[3:0]	<p>General Control</p> <p>These bits set the A/D conversion in each conversion mode.</p> <p>Single-ended input (CADCnCNVCLS = 0_H, 1_H)</p> <p>Analog input channels are selected by the CADCnGCTRL[3:0] bits.</p> <p>0_H: CANn0P 1_H: CANn0N 2_H: CANn1P 3_H: CANn1N 4_H: CANn2P 5_H: CANn2N 6_H: CANn3P 7_H: CANn3N Other than the above: Setting prohibited.</p> <p>Differential input (CADCnCNVCLS = 2_H)</p> <p>Analog input channels are selected by the CADCnGCTRL[3:0] bits.</p> <p>0_H: CANn0P/CANn0N 1_H: Setting prohibited 2_H: CANn1P/CANn1N 3_H: Setting prohibited 4_H: CANn2P/CANn2N 5_H: Setting prohibited 6_H: CANn3P/CANn3N 7_H: Setting prohibited Other than the above: Setting prohibited.</p> <p>For self-diagnosis (CADCnCNVCLS[1:0] = 3_H), a self-diagnosis level is selected by the CADCnGCTRL[3:0] bits.</p> <p>7_H: ADSVREFH × 1 4_H: ADSVREFH × 1/2 0_H: ADSVREFH × 0 C_H: – ADSVREFH × 1/2 8_H: – ADSVREFH × 1 Other than the above: Setting prohibited</p>

NOTE

To prevent a malfunction, set CADCnVCRj when CADCnADACT and CADCnCLBACT = 0 (before starting A/D conversion) and CADCnSTTRGE = 0. When rewriting this register after A/D conversion stop, change the setting after clearing of CADCnADACT is confirmed by reading CADCnADSR.

36.5.2 ADCnDIRj — Data Supplementary Information Register j

ADCnDIRj is a 32-bit read-only register to store A/D converted values and their supplementary information. This register is cleared to 0000 0000_H by reading this register when ADCnRDCLRE is set to 1. Additionally, ADCnUDIR is also cleared to 0000 0000_H by reading this register, which is mirrored to ADCnUDIR when ADCnRDCLRE is set to 1. The ADCnWFLG bit is cleared by reading this register regardless of the ADCnRDCLRE value.

When reading the ADCnDIRj in 16-bit units, invalidate Read-and-Clear-Enable (ADCnSFTCR.ADCnRDCLRE = 0) and invalidate the overwriting function*¹.

Note 1. The expected value comparison of the ADCnWFLG of ADCnDIRj and Overwrite-Error of the Error-Register is not put into effect, and Overwrite-Error-Interrupt-Enable (ADCnOWEIE) of the Safety-Control-Register (ADCnSFTCR) is 0 (prohibited).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ADCnI DEF	ADCn WFLG	ADCn PRTY	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADCnDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 36.15 ADCnDIRj Register Contents

Bit Position	Bit Name	Function
31 to 27	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
26	ADCnIDEF	ID Error 0: An error has occurred. 1: No error has occurred.
25	ADCnWFLG	Write Flag Setting condition Storing an A/D converted value in ADCnDIRj Clearing condition Reading ADCnDIRj
24	ADCnPRTY	Parity This bit indicates the even parity bit of ADCnDR and ADCnIDEF.
23 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	ADCnDR[15:0]	These bits are used to store the A/D converted value. For the data format, see “Bits 15 to 0 DR Data Register Data Format”.

The setting timing of ADCnIDEF, ADCnWFLG and ADCnPRTY are same timing to set the data register to the A/D conversion result (ADCnDR).

If setting and clearing of ADCnIDEF, ADCnWFLG and ADCnPRTY are conflicted, the setting is high priority.

Bits 15 to 0 DR Data Register Data Format

The A/D converted value is stored in the CADCnDR. The data format of the CADCnDR is shown below. The format of data to be transferred to the DFE (digital filter engine) or the GTM (Generic Timer Module) is in the signed fixed-point format regardless of the UNSND setting.

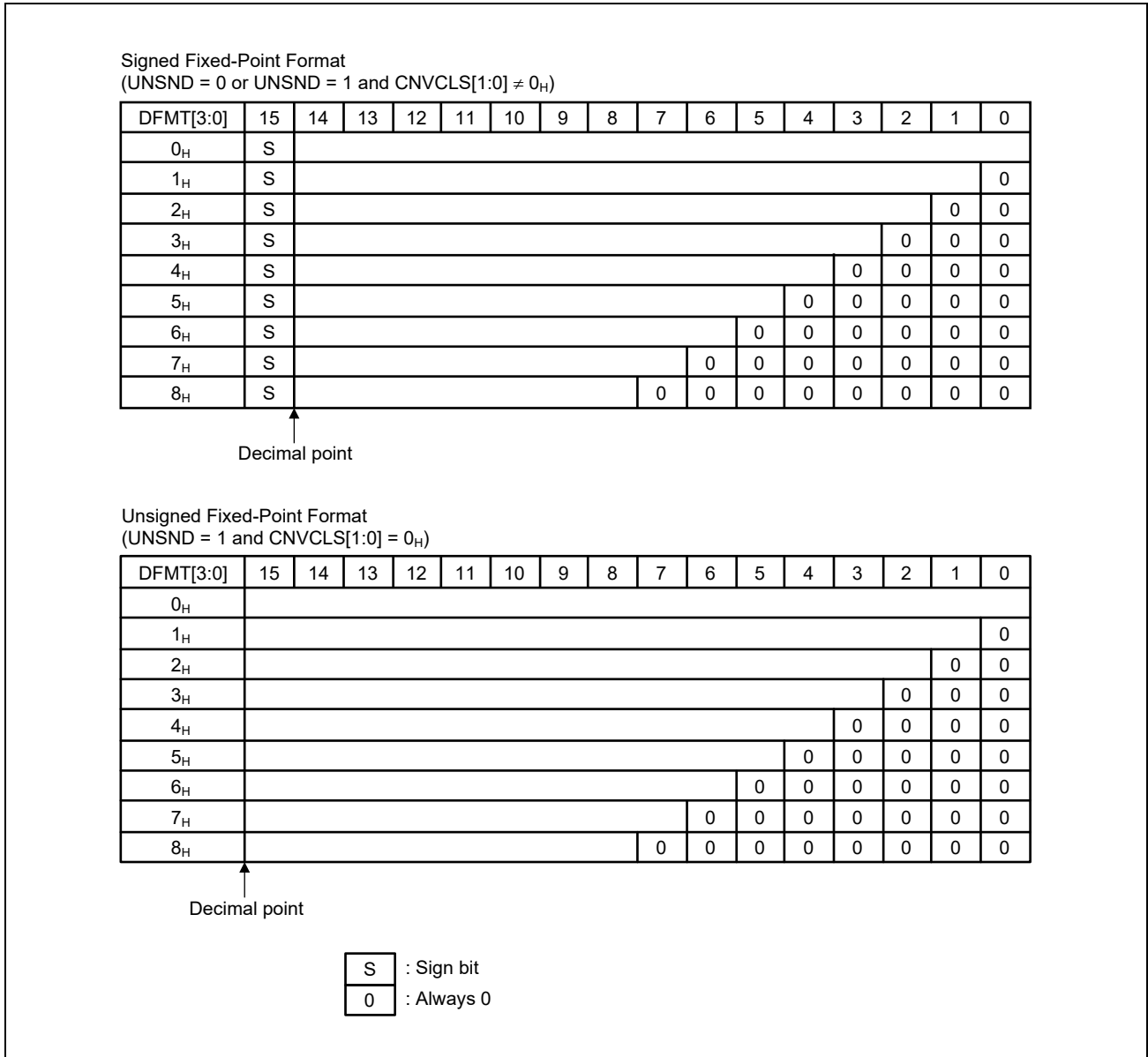


Figure 36.2 DR (Data Register) Data Format 1

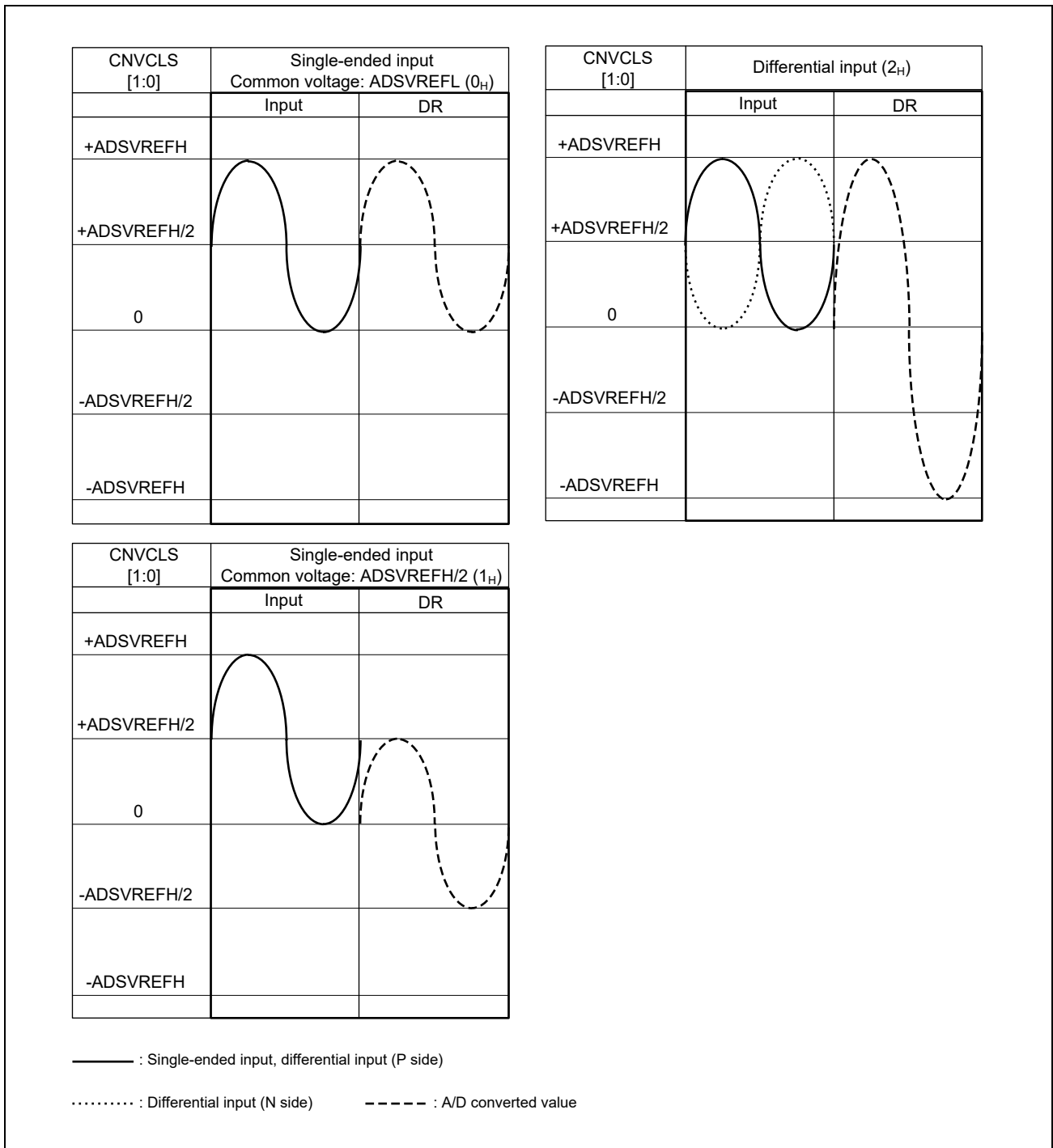


Figure 36.3 DR (Data Register) Data Format 2

36.5.3 CADCnADSTCR — AD Start Control Register

CADCnADSTCR is an 8-bit write-only register to control Cyclic ADC start. This register is always read as 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CADCnADST
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 36.16 CADCnADSTCR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CADCnADST	A/D Conversion Start A/D conversion start condition with CADCnADST Writing 1 to CADCnADST starts an A/D conversion When 1 is written to this bit while an A/D conversion is already in progress, if the CADCnVCEP[2:0] value in CADCnUCR is not 0 _H , the ongoing A/D conversion stops and A/D conversion of the next virtual channel starts. Writing 1 to this bit while ADACT = 1 and CADCnVCEP[2:0] in CADCnUCR is 0 _H suspends the A/D conversion and repeats the A/D conversion of virtual channel 0. When 1 is written to this bit while calibration is in progress, calibration stops and A/D conversion of the next virtual channel that is indicated by the CADCnVCPTR bit starts. Writing 0 to this bit is ignored.

NOTE

It is recommended that a value be written to the CADCnADST bit when CADCnSTTRGE = 0 and CADCnENDTRGE = 0. When the external trigger and to write to CADCnADST occurred at the same time, count value of channel isn't sometimes matched with the number of inputs of the trigger.

36.5.4 CADCnADENDCR — AD Stop Control Register

CADCnADENDCR is an 8-bit write-only register to control Cyclic ADC stop. This register is always read as 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CADCnADEND
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 36.17 CADCnADENDCR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CADCnADEND	A/D Conversion End A/D conversion stop condition with CADCnADEND Writing 1 to CADCnADEND stops an active A/D conversion. When A/D conversion is stopped with CADCnVPRSTE = 0, the virtual channel is incremented. When A/D conversion is stopped with CADCnVPRSTE = 1, the virtual channel is cleared to 0. Writing 1 to this bit when CADCnADACT = 0 is ignored. Writing 0 to this bit is ignored.

NOTE: When changing the other register setting after an A/D conversion is stopped, please be sure to confirm that CADCnADACT is 0.

NOTE

It is recommended that a value be written to the CADCnADEND bit when CADCnSTTRGE = 0 and CADCnENDTRGE = 0.

36.5.5 CADCnCLBSTCR — Calibration Start Control Register

CADCnCLBSTCR is an 8-bit write-only register to control calibration start. This register is always read as 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CADCnCLBST
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 36.18 CAD0nCLBSTCR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CADCnCLBST	Calibration Start Calibration start condition with CADCnCLBST Writing 1 to CADCnCLBST when both calibration and A/D conversion are stopped starts A/D calibration. Writing 1 to this bit when calibration is active or A/D conversion is active is ignored.

36.5.6 CADCnCLBEDCR — Calibration Stop Control Register

CADCnCLBEDCR is an 8-bit write-only register to control calibration stop. This register is always read as 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CADCnCLBEND
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 36.19 CADCnCLBEDCR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CADCnCLBEND	Calibration Stop Calibration stop condition with CADCnCLBEND Writing 1 to CADCnCLBEND when calibration is active, stops A/D calibration. Writing 1 to this bit when calibration is stopped is ignored. If calibration is terminated by CADCnCLBEND before calibration is completed, the correction value present before calibration is started is retained. Writing 0 to this bit is ignored.

36.5.7 CADCnADTCR — AD Conversion Trigger Control Register

CADCnADTCR is an 8-bit read/write register to control the Cyclic ADC.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CADCn ENDTRGE	CADCnSTTRGE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 36.20 CADCnADTCR Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CADCnENDTRGE	AD End Trigger Enable 0: AD end trigger n is disabled. 1: AD end trigger n is enabled. The trigger selection function is used to set the trigger source selection for AD end trigger n. For details, see Section 33, Peripheral Interconnection (PIC) .
0	CADCnSTTRGE	AD Start Trigger Enable 0: AD start trigger n is disabled. 1: AD start trigger n is enabled. The trigger selection function is used to set the trigger source selection for AD start trigger n. For details, see Section 33, Peripheral Interconnection (PIC) .

NOTE

Enable the module standby function when CADCnENDTRGE = 0 and CADCnSTTRGE = 0. For details, see **Section 36.7.2, Module Standby Function**.

36.5.8 CADCnUCR — Unit Control Register

CADCnUCR is a 32-bit read/write register to control the cyclic AD units.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	CADCn VP RSTE	CADCn RDMA	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CADCnDFMT[3:0]				—	—	—	—	—	CADCnVCEP[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 36.21 CADCnUCR Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
29	CADCnVPRSTE	Virtual Channel Pointer Reset Enable 0: The virtual channel pointer is incremented at the end of an A/D conversion by AD end trigger n or CADCnADEND. 1: The virtual channel pointer is cleared to 0 at the end of an A/D conversion by AD end trigger n or CADCnADEND.
28	CADCnRDMA	Read Gate DMA Mode This bit selects the DMA transfer request output condition. 0: A DMA transfer request is output for all A/D conversion results. 1: A DMA transfer request is output for the A/D conversion result only when CADRGTr is being asserted.
27 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 36.21 CADcNUCR Register Contents (2/2)

Bit Position	Bit Name	Function
11 to 8	CADcNDFMT[3:0]	<p>Data Format</p> <p>These bits specify the lower bits to be zero-masked (rounded in the $-\infty$ direction).</p> <p>0_H: Not masked 1_H: Lower 1 bit masked 2_H: Lower 2 bits masked 3_H: Lower 3 bits masked 4_H: Lower 4 bits masked 5_H: Lower 5 bits masked 6_H: Lower 6 bits masked 7_H: Lower 7 bits masked 8_H: Lower 8 bits masked Others: Setting prohibited</p> <p>The setting of these bits is the format of the data for the CADcNDR, DFE and GTM. For details of data format, see Section 36.5.2, CADcNDIRj — Data Supplementary Information Register j.</p>
7 to 3	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
2 to 0	CADcNVCEP[2:0]	<p>End Virtual Channel Pointer</p> <p>These bits set the end virtual channel number.</p>

NOTE

To prevent a malfunction, set CADcNUCR when CADcNADACT = 0 and CADcNCLBACT = 0 (before starting A/D conversion) and CADcNSTTRGE = 0.

36.5.9 CADCnVCPTRR — Virtual Channel Pointer Register

CADCnVCPTRR is an 8-bit read/write register to monitor the number of the virtual channels in which A/D conversion is in progress.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	CADCnVCPTR[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 36.22 CADCnVCPTRR Register Contents

Bit Position	Bit Name	Function
7 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	CADCnVCPTR[2:0]	A/D Conversion Ongoing Virtual Channel Number These bits output the number of virtual channels in which the A/D conversion is in progress. If a value is written to these bits while A/D conversion is not ongoing, the virtual channel number is updated and A/D conversion starts for the channel written at the next conversion start. Writing to these bits during A/D conversion is prohibited. The channel number for which A/D conversion is to be made next is retained during conversion stop or during calibration. A setting exceeding the end pointer is prohibited. When the virtual channel is switched by external trigger or CADCnADST setting, 20 register access cycles are required to update CADCnVCPTR register.

NOTE

To prevent a malfunction, set CADCnVCPTRR when CADCnADACT = 0, CADCnCLBACT = 0 (before starting A/D conversion) and CADCnSTTRGE, CADCnENDTRGE = 0. When writing to CADCnVCPTR with external trigger use, it is necessary to stop A/D conversion at the exact timing. When writing to CADCnVCPTR, set CADCnSTTRGE and CADCnENDTRGE as 0, and confirm CADCnADACT = 0, or write CADCnADEND = 1 to ensure that A/D conversion stops.

36.5.10 CADCnADSR — AD Conversion Status Register

CADCnADSR is an 8-bit read-only register that indicates the Cyclic ADC status.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CADCnCLBACT	CADCnADACT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 36.23 CADCnADSR Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CADCnCLBACT	Calibration Status 0: Calibration is not ongoing. 1: Calibration is in progress.
0	CADCnADACT	A/D Conversion Status 0: A/D conversion is stopped. 1: A/D conversion is ongoing. The CADCnADACT bit is set to 1 when an A/D conversion start source is generated, and is set to 0 when the A/D conversion end source is generated. This bit is continuously set to 1 unless an A/D conversion end source is generated. (This bit is not cleared to 0 by storing in the data register.) If an A/D conversion start source is generated when CADCnADACT = 1 (where the period of CADCnADACT=0 just after the CADCnADST set is included, or the period of CADCnADACT=1 just after the CADCnADEND set is not included), the ongoing A/D conversion is stopped and A/D conversion of the next virtual channel is started. An A/D conversion end source that is generated when CADCnADACT = 0 (where the period of CADCnADACT=0 just after the CADCnADST set is not included, or the period of CADCnADACT=1 just after the CADCnADEND set is included) is ignored. About the timing of setting or clearing of CADCnADACT, see Figure 36.5 .

NOTE

Enable the module standby function when CADCnADACT and CADCnCLBACT = 0. For details, see **Section 36.7.2, Module Standby Function**.

36.5.11 CADCnUDPTRR — Unit Data Pointer Register

CADCnUDPTRR is an 8-bit read-only register that indicates the virtual channel number of CADCnUDIR.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	CADCnUDPTR[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 36.24 CADCnUDPTRR Register Contents

Bit Position	Bit Name	Function
7 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	CADCnUDPTR[2:0]	Unit Data Pointer These bits indicate the virtual channel number in which the latest A/D conversion result is stored.

36.5.12 ADCnUDIR — Unit Data Supplementary Information Register

ADCnUDIR is a 32-bit read-only register to output all A/D conversion results of ADCn units regardless of the selection of the virtual channel. Except for ADCnCHNUM[2:0] and ADCnUPRTY, this register is a mirror register of ADCnDIRj specified by ADCnUDPTR[3:0]. When ADCnUDIR is read, ADCnWFLG of the actual ADCnDIRj is cleared at the same time. When ADCnRDCLRE is set to 1, the ADCnDR[15:0], ADCnPRTY, and ADCnIDEF bits in ADCnDIRj register are cleared in addition to the ADCnCHNUM[2:0] and ADCnUPRTY bits in this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADCnCHNUM[2:0]			—	—	ADCnUIDEF	ADCnUWFLG	ADCnUPRTY	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADCnUDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 36.25 ADCnUDIR Register Contents

Bit Position	Bit Name	Function
31 to 29	ADCnCHNUM[2:0]	Virtual Channel Number These bits indicate the virtual channel number of ADCnUDR[15:0].
28, 27	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
26	ADCnUIDEF	ID Error 0: An error is present. 1: No error is present.
25	ADCnUWFLG	Write Flag Setting condition Stores an A/D converted value in ADCnUDIR Clearing condition Reads ADCnUDIR or reading ADCnDIRj corresponding to ADCnUDPTR
24	ADCnUPRTY	Parity This bit indicates the even parity bit of ADCnUDR, ADCnUIDEF, and ADCnCHNUM.
23 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	ADCnUDR[15:0]	All A/D converted values of ADCn are stored in these bits. The data format is the same as ADCnDIRj.

36.5.13 CADCNMPCR — Sampling Rate Control Register

CADCNMPCR is a 32-bit read/write register to set the sampling rate.

Value after reset: 0001 0707_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CADCN ODAV
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 36.26 CADCNMPCR Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	CADCNODAV	Output Data Averaging 0: Data is output without averaging (oversampling rate = 1) 1: Data is output with averaging (oversampling rate = 2)
15 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

NOTE

To prevent a malfunction, set CADCNMPCR when CADCNADACT and CADCNCLBACT = 0 (before starting A/D conversion) and CADCNSTTRGE = 0.

36.5.14 CADCnSFTCR — Safety Control Register

CADCnSFTCR is an 8-bit read/write register for safety control.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CADCnRDCLRE	—	—	—	—	CADCnOWEIE	CADCnPEIE	CADCnIDEIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R/W	R/W	R/W

Table 36.27 CADCnSFTCR Register Contents

Bit Position	Bit Name	Function
7	CADCnRDCLRE	Read and Clear Enable 0: CADCnDIRj (CADCnUDIR) is not cleared by reading CADCnDIRj (CADCnUDIR). 1: CADCnDIRj (CADCnUDIR) is cleared by reading CADCnDIRj (CADCnUDIR). The WFLG of CADCnDIRj (CADCnUDIR) is cleared by reading CADCnDIRj (CADCnUDIR) regardless of the RDCLRE value. When CADCnUDIR is read after CADCnRDCLRE has been set to 1, CADCnDIRj which corresponds with CADCnUDPTR is also cleared at the same time. When CADCnDIRj which corresponds with CADCnUDPTR is read, CADCnUDIR is also cleared at the same time.
6 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	CADCnOWEIE	Overwrite Error Interrupt Enable 0: Disabled 1: Enabled The Overwrite Error Interrupt and DMA transfer using CADIn with read-gate cannot be used at the same time. Because overwrite errors occur in spite of using read gate function, using read gate function and overwrite error function at the same time is prohibited.
1	CADCnPEIE	Parity Error Interrupt Enable 0: Disabled 1: Enabled
0	CADCnIDEIE	ID Error Interrupt Enable 0: Disabled 1: Enabled

NOTE

To prevent a malfunction, set CADCnSFTCR when CADCnADACT = 0 (before starting A/D conversion) and CADCnSTTRGE = 0.

36.5.15 CADCnECR — Error Clear Register

CADCnECR is an 8-bit write-only register to control error clear. This register is always read as 0.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	CADCnULEC	CADCnLLEC	CADCnOWEC	CADCnPEC	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	W	W	W	W	R

Table 36.28 CADCnECR Register Contents

Bit Position	Bit Name	Function
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	CADCnULEC	Upper-Limit Error Clear 0: Does not clear the upper-limit error. 1: Clears the upper-limit error.
3	CADCnLLEC	Lower-Limit Error Clear 0: Does not clear the lower-limit error. 1: Clears the lower-limit error.
2	CADCnOWEC	Overwrite Error Clear 0: Does not clear the overwrite error. 1: Clears the overwrite error.
1	CADCnPEC	Parity Error Clear 0: Does not clear the parity error. 1: Clears the parity error.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

36.5.16 CADCnER — Error Register

CADCnER is a 32-bit read-only register that indicates errors.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CADCnULE	CADCnLLE	—	—	—	CADCnULECAP[2:0]			CADCnOWE	—	—	—	—	CADCnOWECAP[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CADCnPE	—	—	—	—	CADCnPECAP[2:0]			—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 36.29 CADCnER Register Contents (1/2)

Bit Position	Bit Name	Function
31	CADCnULE	Upper-Limit Error 0: No upper-limit error is present. 1: An upper-limit error is present. Setting condition The A/D converted value exceeds the specified upper-limit table range. Clearing condition Writing 1 to CADCnULEC When CADCnLLE is set already, CADCnULE is not set.
30	CADCnLLE	Lower-Limit Error 0: No lower-limit error is present. 1: A lower-limit error is present. Setting condition The A/D converted value exceeds the specified lower-limit table range. Clearing condition Writing 1 to CADCnLLEC When CADCnULE was set already, CADCnLLE is not set.
29 to 27	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
26 to 24	CADCnULECAP[2:0]	Upper-Limit/Lower-Limit Error Capture These bits show the virtual channel number in which an upper-limit or lower-limit error occurred. When writing 1 to CADCnULEC, these bits are cleared to 0 _H .
23	CADCnOWE	Overwrite Error 0: No overwrite error is present. 1: An overwrite error is present. Setting condition Writing the A/D converted value to CADCnDIRj.DR when CADCnWFLG = 1 Even if CADCnUDIR.CADCnUDR is overwritten when CADCnUDIR.CADCnUWFLG = 1, this bit is not set to 1 if CADCnDIRj.CADCnDR is not overwritten. Clearing condition Writing 1 to CADCnOWEC
22 to 19	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 36.29 CADCnER Register Contents (2/2)

Bit Position	Bit Name	Function
18 to 16	CADCnOWECAP[2:0]	Overwrite Error Capture These bits show the virtual channel number in which an overwrite error occurred. When writing 1 to CADCnOWEC, these bits are cleared to 0 _H .
15	CADCnPE	Parity Error 0: No parity error is present. 1: A parity error is present. Setting condition A parity error is detected. Clearing condition Writing 1 to CADCnPEC
14 to 11	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	CADCnPECAP[2:0]	Parity Error Capture These bits show the virtual channel number in which a parity error occurred. When writing 1 to CADCnPEC, these bits are cleared to 0 _H . When a parity error occurred in CADCnUDIR, these bits indicate the number of CADCnCHNUM.
7 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

NOTE

The CADCnULE and CADCnOWE bits are updated when an A/D converted value is written to the CADCnDIR.

The CADCnPE bit is updated when the CADCnDIR is read.

When an AD error interrupt request is generated, read the CADCnER and check for the “1” bits in the interrupt exception handling, and then clear them to 0 by using the corresponding clear bits in the CADCnECCR.

When an error bit is not cleared and an error occurs during the next A/D conversion, the following operation takes place and the error cannot be detected or identified.

- (1) When the same error (error bit = 1) occurs:
 - No AD error interrupt request is made.
 - The relevant error bit in the CADCnER remains 1.
 - CADCnULE and CADCnLLE are treated as the same error.
- (2) When a different error (error bit = 0) occurs:
 - An AD error interrupt request is made.
 - The error bit that was generated in the previous A/D conversion remains 1 and the relevant error bit in the CADCnER is set to 1.
 - CADCnULE and CADCnLLE are treated as the same error.

36.5.17 ADCnTDLVR — Pin Level Self-Diagnosis Level Register

ADCnTDLVR is a 8-bit readable/writable register to specify pin level self-diagnosis levels.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CADCnAN3NLV	CADCnAN3PLV	CADCnAN2NLV	CADCnAN2PLV	CADCnAN1NLV	CADCnAN1PLV	CADCnAN0NLV	CADCnAN0PLV
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 36.30 ADCnTDLVR Register Contents

Bit Position	Bit Name	Function
7	CADCnAN3NLV	Pin Level Self-Diagnosis Level of CANn3N 0: CANn3N is discharged (ADSVSS). 1: CANn3N is charged (ADSVCC).
6	CADCnAN3PLV	Pin Level Self-Diagnosis Level of CANn3P 0: CANn3P is discharged (ADSVSS). 1: CANn3P is charged (ADSVCC).
5	CADCnAN2NLV	Pin Level Self-Diagnosis Level of CANn2N 0: CANn2N is discharged (ADSVSS). 1: CANn2N is charged (ADSVCC).
4	CADCnAN2PLV	Pin Level Self-Diagnosis Level of CANn2P 0: CANn2P is discharged (ADSVSS). 1: CANn2P is charged (ADSVCC).
3	CADCnAN1NLV	Pin Level Self-Diagnosis Level of CANn1N 0: CANn1N is discharged (ADSVSS). 1: CANn1N is charged (ADSVCC).
2	CADCnAN1PLV	Pin Level Self-Diagnosis Level of CANn1P 0: CANn1P is discharged (ADSVSS). 1: CANn1P is charged (ADSVCC).
1	CADCnAN0NLV	Pin Level Self-Diagnosis Level of CANn0N 0: CANn0N is discharged (ADSVSS). 1: CANn0N is charged (ADSVCC).
0	CADCnAN0PLV	Pin Level Self-Diagnosis Level of CANn0P 0: CANn0P is discharged (ADSVSS). 1: CANn0P is charged (ADSVCC).

NOTE

To prevent a malfunction, set ADCnTDLVR when ADCnADACT = 0 (before starting A/D conversion) and ADCnSTTRGE = 0.

36.5.18 CADCnULTBR0/1/2/3 — Upper-Limit/Lower-Limit Table Register 0/1/2/3

CADCnULTBR0 to CADCnULTBR3 are 32-bit readable/writable registers to set upper-limit and lower-limit values of A/D converted values.

Value after reset: 7FFF 8000_H

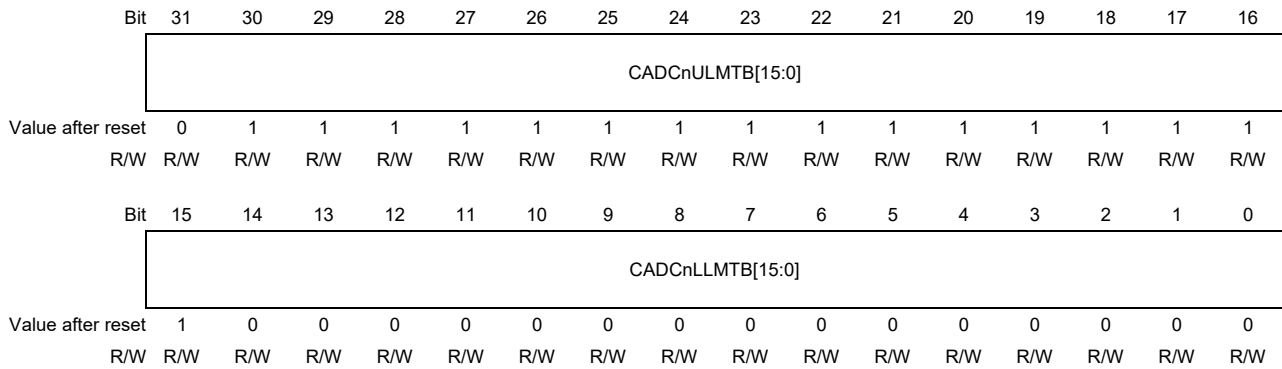


Table 36.31 CADCnULTBR0/1/2/3 Register Contents

Bit Position	Bit Name	Function
31 to 16	CADCnULMTB[15:0]	<p>Upper-Limit Table</p> <p>These bits specify the upper-limit value of the A/D converted value. When the following condition is met, CADCnULE (upper-limit error) is set to 1.</p> <p style="padding-left: 20px;">ULMTB[15:0] < A/D converted value</p> <p>The ULMTB[15:0] format is the same as the DR format. For unsigned format, the value is compared as unsigned. A/D converted values are compared by using the values before they are masked by DFMT[3:0].</p>
15 to 0	CADCnLLMTB[15:0]	<p>Lower-Limit Table</p> <p>These bits specify the lower-limit value of the A/D converted value. When the following condition is met, CADCnLLE (lower-limit error) is set to 1</p> <p style="padding-left: 20px;">LLMTB[15:0] > A/D converted value</p> <p>The LLMTB[15:0] format is the same as the DR format. For unsigned format, the value is compared as unsigned. A/D converted values are compared by using the values before they are masked by DFMT[3:0].</p>

NOTE

To prevent a malfunction, set CADCnULTBR0/1/2/3 when CADCnADACT = 0 (before starting A/D conversion) and CADCnSTTRGE = 0.

36.6 Operation

36.6.1 A/D Conversion Flow

The Cyclic ADC starts A/D conversion by setting the registers shown in **Figure 36.4**.

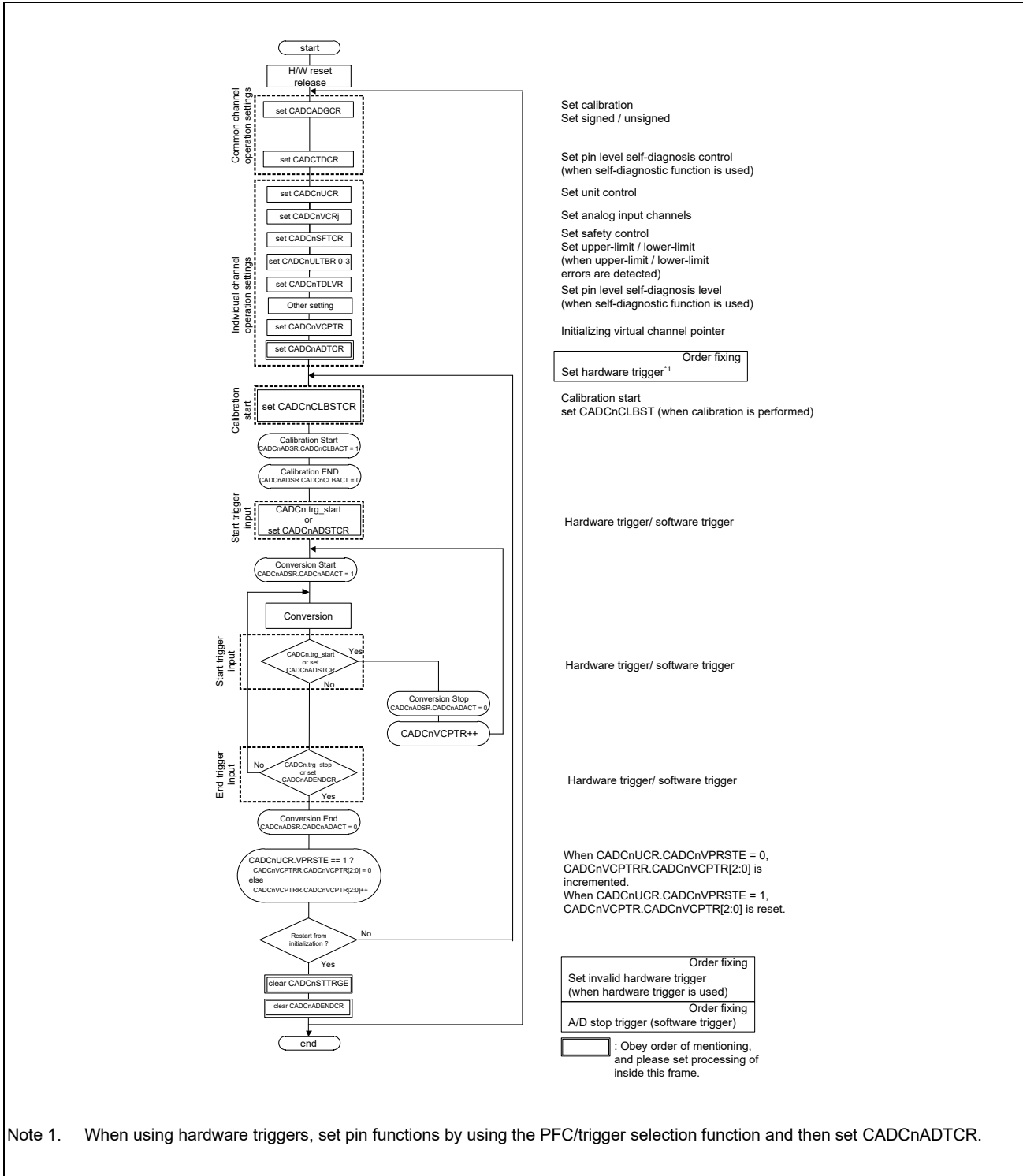


Figure 36.4 A/D Conversion Flow

36.6.2 A/D Conversion time

The Cyclic ADC starts A/D conversion processing immediately after the ADST bit in ADSTCR is set to 1.

Figure 36.5 shows the A/D conversion start timing and end timing. **Figure 36.6** and **Figure 36.7** show the budget of A/D conversion processing. The first A/D processing time (t_{AD}) includes the A/D conversion start delay time (t_D), the internal stabilization time (t_W), and the result calculation processing delay time (t_{DF}). Subsequent A/D conversion processing time (t_{AD}) is the sampling period (t_S). **Table 36.32** shows A/D conversion processing time.

A/D conversion processing time (t_{AD}) is calculated by the following formula.

$$\text{First A/D processing time } t_{AD} = t_D + t_W + t_{DF}$$

$$\text{Subsequent A/D processing time } t_{AD} = t_S$$

Table 36.32 Conversion Time (When Register Access Clock = 40 MHz, ADCLK = 80 MHz, Unit: Register Access Clocks)

Item	Without averaging	With averaging
t_{AD}	907 to 913	927 to 933
t_D	18 to 22	18 to 22
t_W	840.5 to 842.5	840.5 to 842.5
t_{DF}	48.5	68.5
t_S	20	40

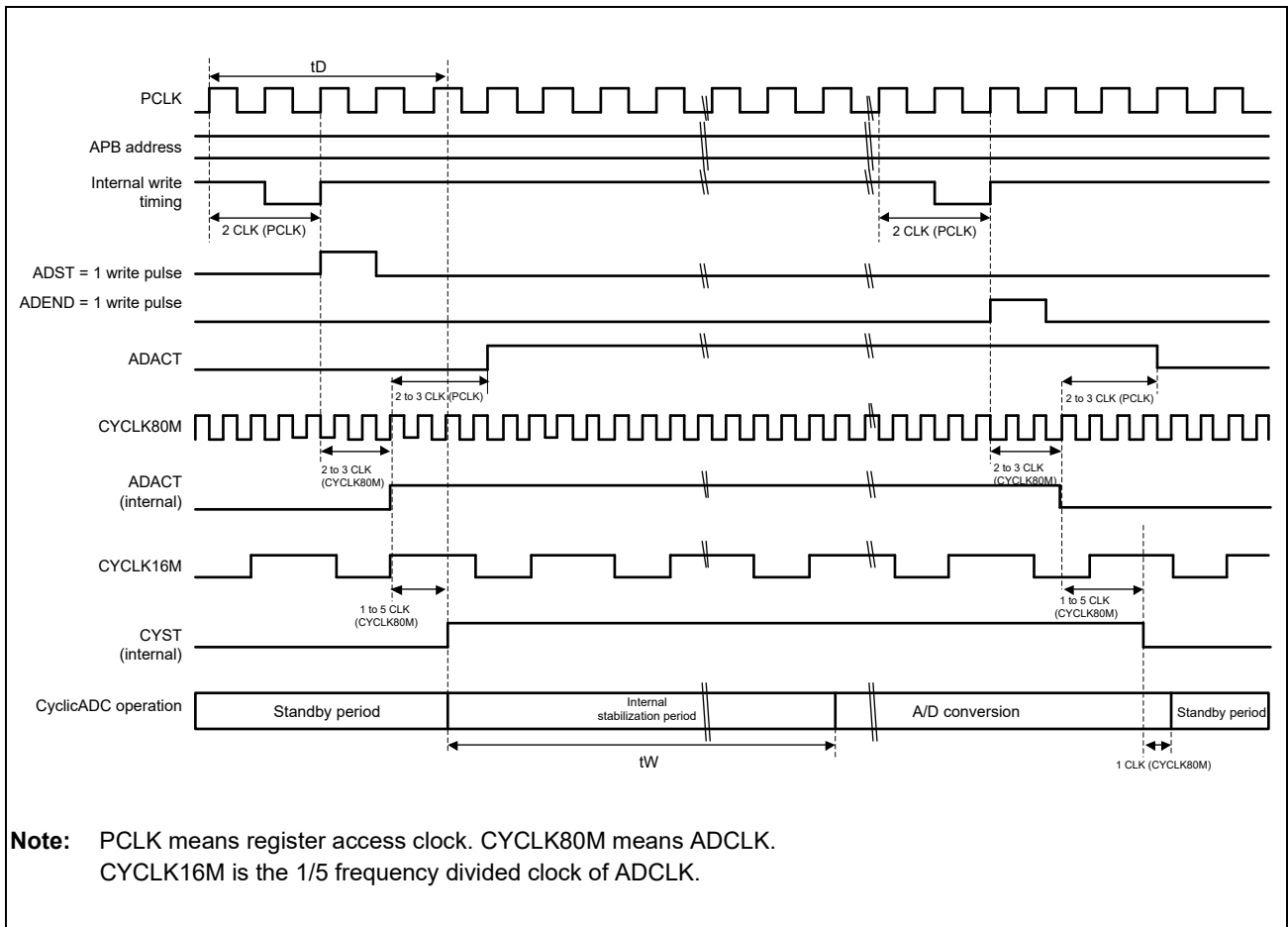


Figure 36.5 A/D Conversion Start and End

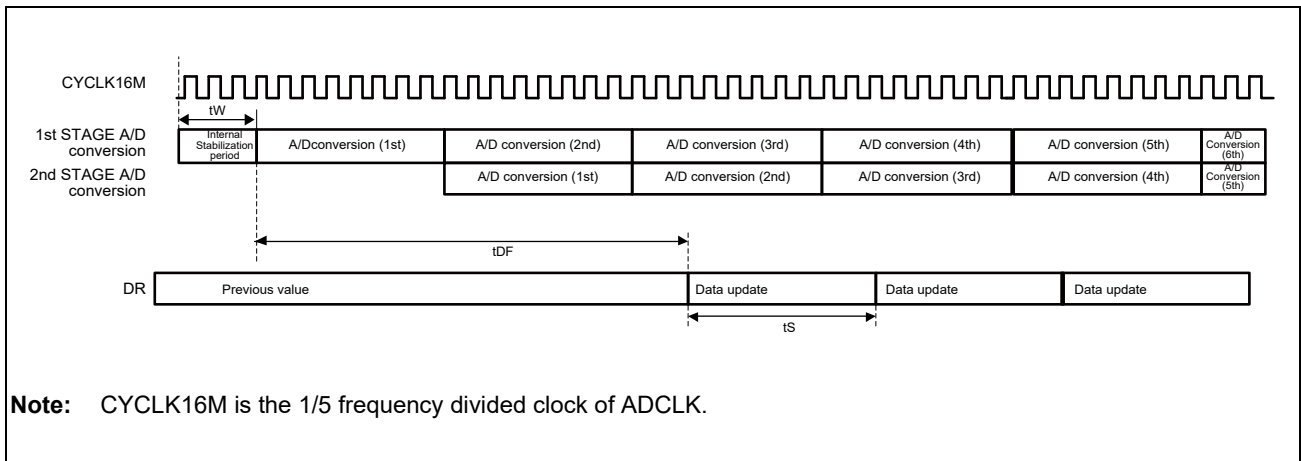


Figure 36.6 Budget of A/D Conversion Processing (CADCnODAV = 0)

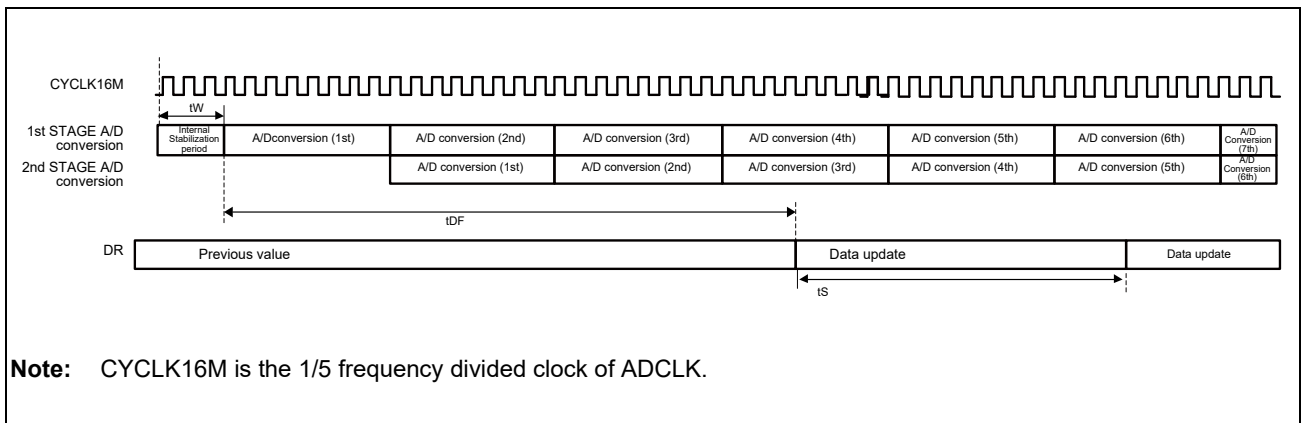


Figure 36.7 Budget of A/D Conversion Processing (CADCnODAV = 1)

36.6.3 Starting an A/D Conversion by External Triggers

A ADCn A/D conversion can be started by a CADTRGn input. To start a ADCn by a CADTRGn, input a high level to CADTRGn, and then set CADCnSTTRGE in CADCnADTCR to 1. When a low level is input to CADTRGn in this state, the ADCn detects a pulse falling edge and sets CADCnADACT to 1.

Figure 36.8 shows the external trigger input timing. CADCnADACT is set to 1 in 3 to 4 states of the ADCLK and 2 to 3 states of the register access clock after the CADTRGn falling edge is sampled.

The timing after CADCnADACT is set to 1 until ADCn is started is the same as when the CADCnADST is set to 1 by the software. For pin functions, see **Section 2, Pin Function**. For selecting the AD start trigger n source, see **Section 33, Peripheral Interconnection (PIC)**.

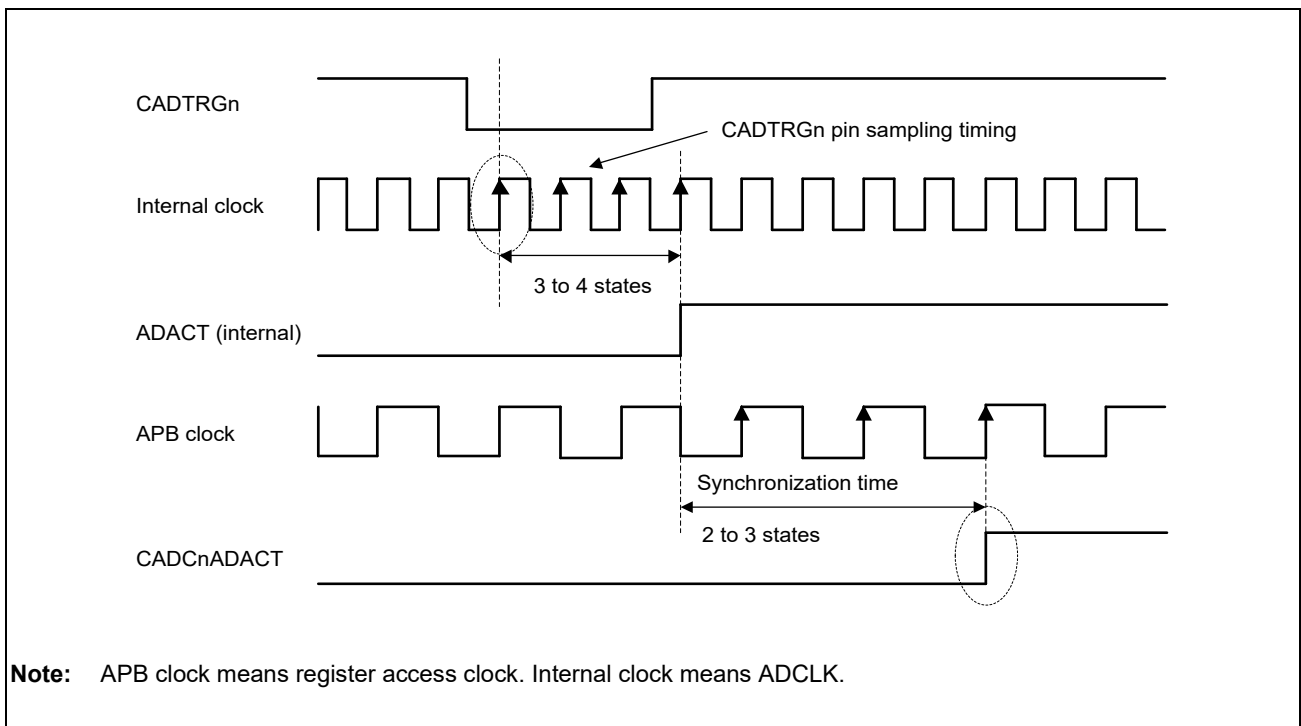


Figure 36.8 External Trigger Input Timing

36.6.4 Terminating an A/D Conversion via an External Trigger

A ADCn conversion can be terminated via a CADTRGn input. To terminate a ADCn conversion via a CADTRGn, set pin functions by using the PFC (pin function controller) and select the AD end trigger n source by using the trigger selection function. Input a high level to CADTRGn, and then set CADCnENDTRGE in CADCnADTCR to 1. When a low level is input to CADTRGn in this state, ADCn detects a pulse falling edge and clears CADCnADACT to 0 to terminate the ADCn conversion.

CADCnADACT is cleared to 0 in 3 to 4 states of the ADCLK and 2 to 3 states of the register access clock after the CADTRGn falling edge is sampled.

For pin functions setting, see **Section 2, Pin Function**. For selecting AD end trigger n source, see **Section 33, Peripheral Interconnection (PIC)**.

To terminate A/D conversion via an external trigger, input an external trigger when CADCnADACT = 1. If an external trigger is input when CADCnADACT = 0, the external trigger is ignored.

36.6.5 Starting an A/D Conversion via a Timer Trigger

A ADCn conversion can be started via an arbitrary timer trigger. To start a ADCn conversion via a timer trigger, select a timer trigger as an AD start trigger n source by using the trigger selection function. Set CADCnSTTRGE in CADCnADTCR to 1.

When the selected timer trigger is input in this state, CADCnADACT is set to 1. The timing after CADCnADACT is set to 1 until the ADCn conversion is started is the same as the case where CADCnADST is set to 1 by the software.

For selecting the AD start trigger n source, see **Section 33, Peripheral Interconnection (PIC)**.

To stop ADCn conversion, write 1 to CADCnADEND.

36.6.6 Terminating an A/D Conversion via a Timer Trigger

A ADCn conversion can be terminated via an arbitrary timer trigger. To terminate a ADCn conversion via a timer trigger, select a timer trigger as an AD end trigger n source by using the trigger selection function. Set CADCnENDTRGE in CADCnADTCR to 1.

When the selected timer trigger is input in this state, CADCnADACT is cleared to 0 to terminate ADCn.

For selecting the AD end trigger n source, see **Section 33, Peripheral Interconnection (PIC)**.

To terminate A/D conversion via a timer trigger, input a timer trigger when CADCnADACT = 1. If a timer trigger is input when CADCnADACT = 0, the timer trigger is ignored.

36.6.7 Switching Virtual Channels via Start and End Triggers

When the $CADCnVCEP[2:0]$ bits in the $CADCnUCR$ register are set to a channel other than 0, they are incremented (+) at each input of an external trigger (start trigger or stop trigger) or a software trigger (start trigger or stop trigger) during A/D conversion. Virtual channels can be sequentially switched from $CADCnVCR0$ to $CADCnVCRj$ ($j = CADCnVCEP[2:0]$ set value) for A/D conversion by inputting a trigger continuously at constant intervals over $ADCLK \times 1$ cycle from the external timer. In addition, when start or end pulse inputting interval is less than $ADCLK \times 1$ cycle, switching virtual channels cannot operate correctly. When $CADCnVPRSTE$ is set to 1, virtual channels always return to $CADCnVCR0$ at a stop trigger. By using this operation, an A/D conversion can also be started from any $CADCnVCRj$ by continuously generating a start trigger after a stop trigger. If a trigger is input during A/D conversion of the channel set by $CADCnVCEP[2:0]$, the channel returns to $CADCnVCR0$ again, and then A/D conversion is performed.

Figure 36.9 to Figure 36.11 show examples of normal virtual channel switching made by a start trigger and an end trigger.

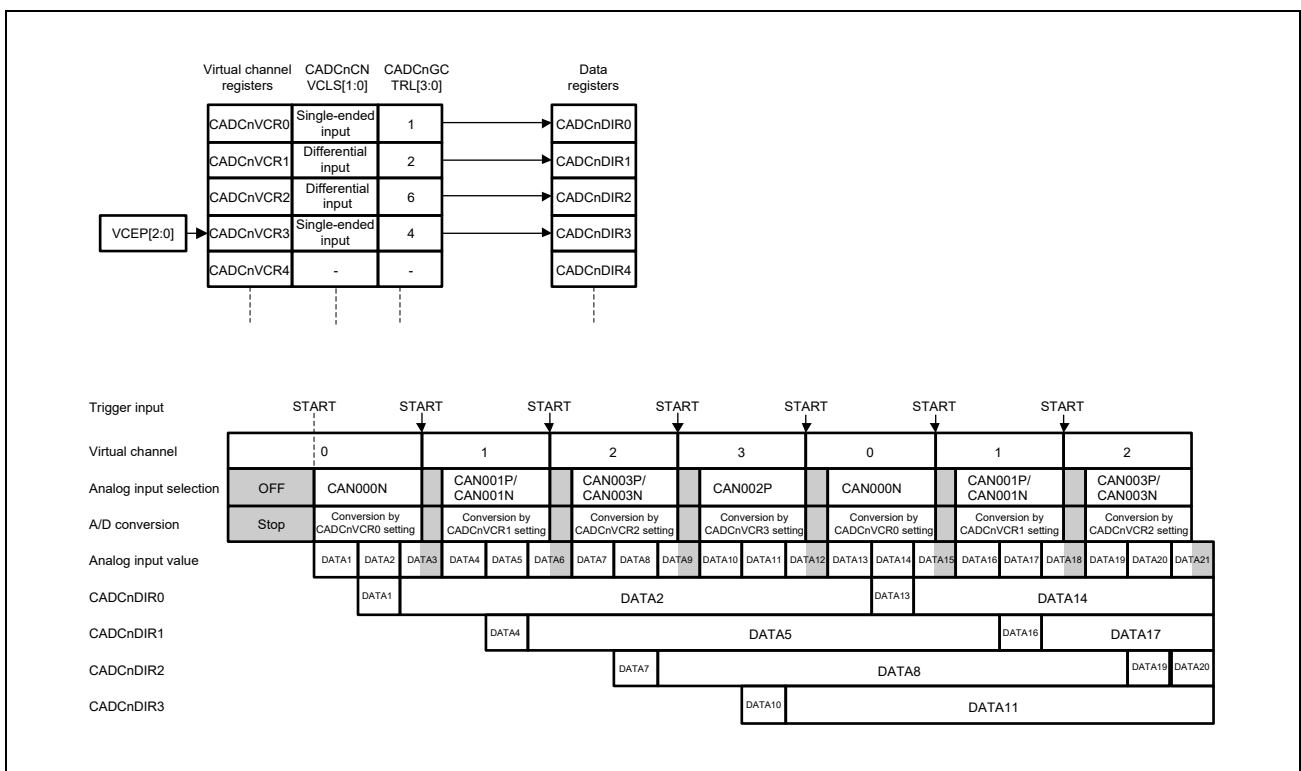


Figure 36.9 Switching Virtual Channels via a Start Trigger

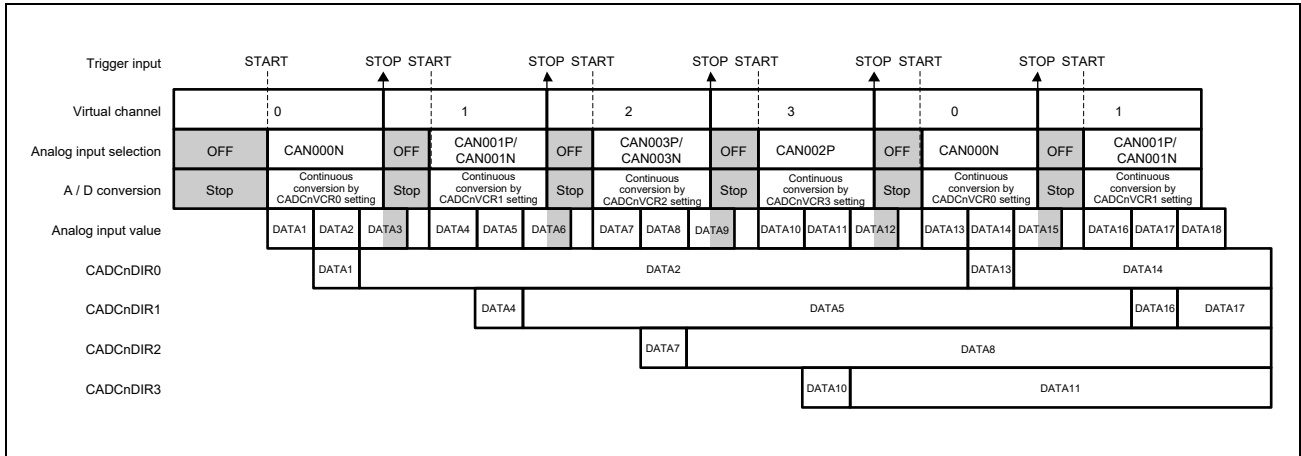


Figure 36.10 Switching Virtual Channels via an End Trigger (VPRSTE = 0)

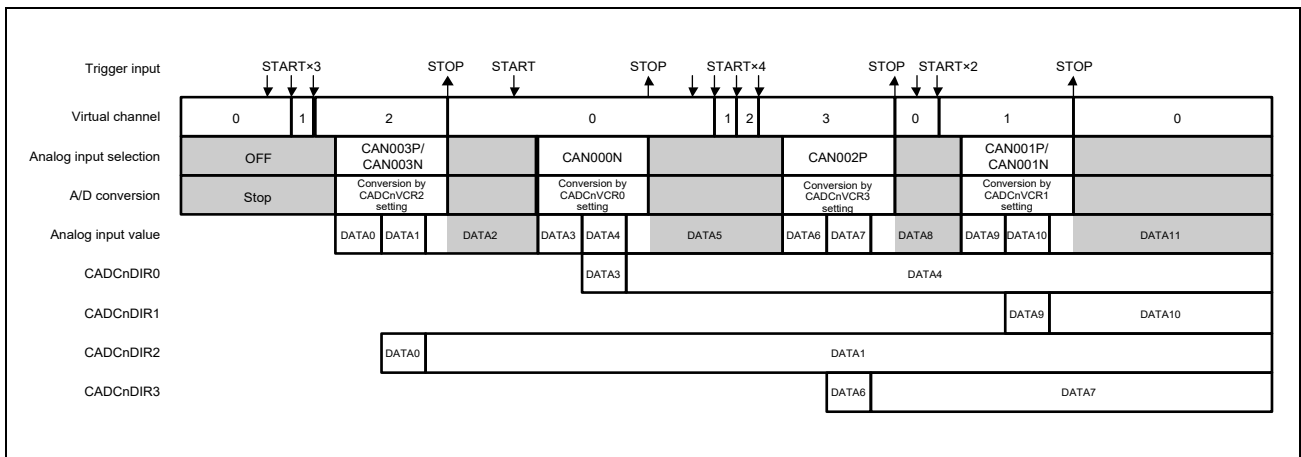


Figure 36.11 Switching Virtual Channels via an End Trigger (VPRSTE = 1)

36.6.8 A/D Conversion Monitoring Function Using Monitor Pins

A/D conversion of the CADC_n can be monitored by the CADEND_n. CADEND_n pins directly output CADC_nADACT of unit n. For the pin settings, see **Section 2, Pin Function**.

Figure 36.12 shows an example of A/D conversion monitor pin output.

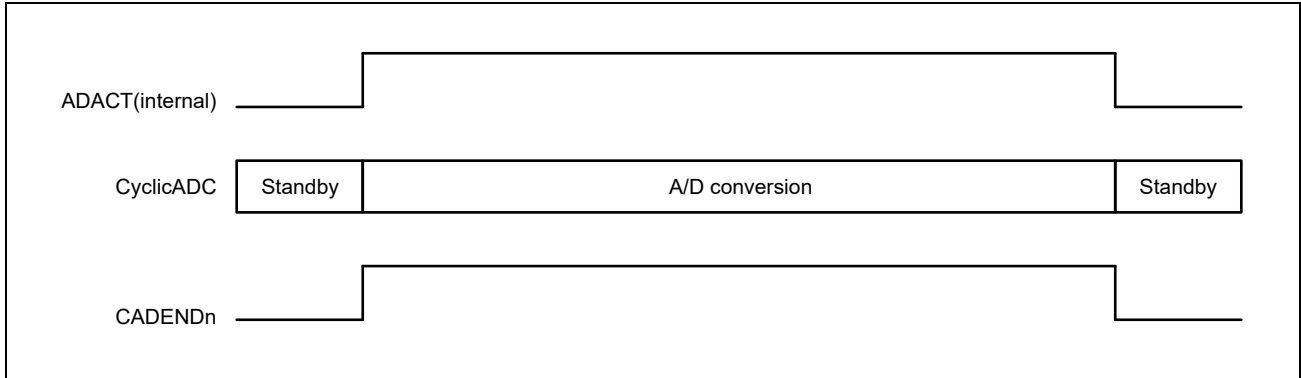


Figure 36.12 Example of A/D Conversion Monitor Pin Output

36.6.9 DMA Request Sources

The Cyclic ADC generates DMA request sources (CADIn) to activate the DMAC (without generating an interrupt to the INTC). When the CADnADIE in CADnVCRj is set to 1, the CADIn is output at the end of the A/D conversion. When the CADnADIE in CADnVCRj is cleared to 0, the CADIn output is disabled even at the end of the A/D conversion.

For the DMAC settings, see **Section 7, sDMA Controller (sDMAC)**.

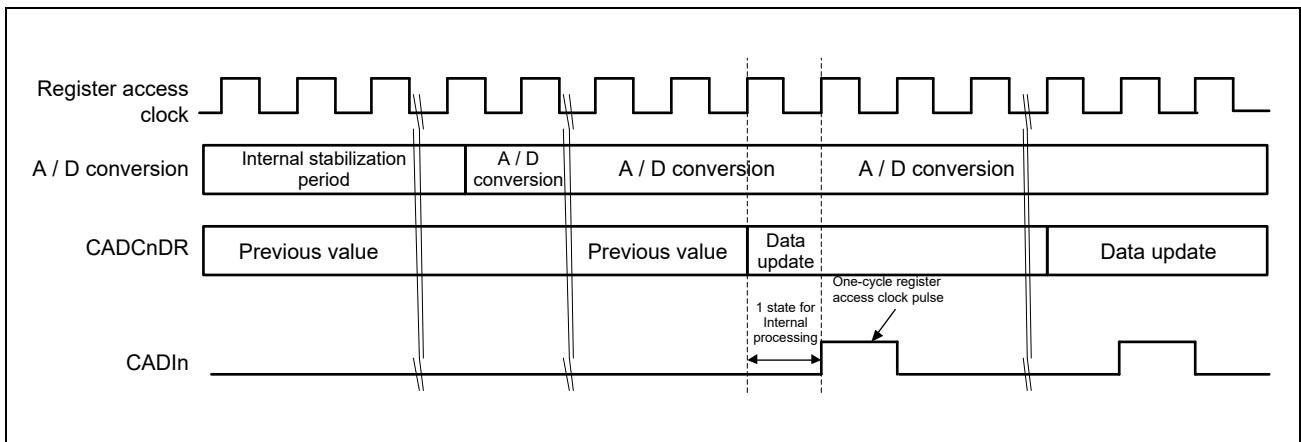


Figure 36.13 Example of DMA Request Source

36.6.10 AD Error Interrupt Request and AD Parity Error Interrupt Request

The Cyclic ADC can generate an AD error interrupt request (CADEn) to the INTC and an AD parity error interrupt request (CADPEn) to the ECM (error control module). If CADCnULEIE in CADCnVCRj, or CADCnIDEIE or CADCnOWEIE in CADCnSFTCR, is set to 1, that can be an error source, CADEn. If CADCnULEIE in CADCnVCRj, or CADCnIDEIE or CADCnOWEIE in CADCnSFTCR, is set to 0, that cannot be an error source, CADEn. When CADCnPEIE in CADCnSFTCR is set to 1, CADPEn is enabled. When CADCnPEIE is set to 0, CADPEn is disabled.

Table 36.33 Correspondence between Interrupt Sources and Interrupt Signals

Status	Interrupt Signal	Interrupt Enable
CADCnLLE	CADEn	CADCnVCRj.CADCnULEIE and CADCnVCRj.CADCnVCLLME
CADCnULE		CADCnVCRj.CADCnULEIE and CADCnVCRj.CADCnVCULME
CADCnIDE	CADPEn	CADCnIDEIE
CADCnOWE		CADCnOWEIE
CADCnPE		CADCnPEIE

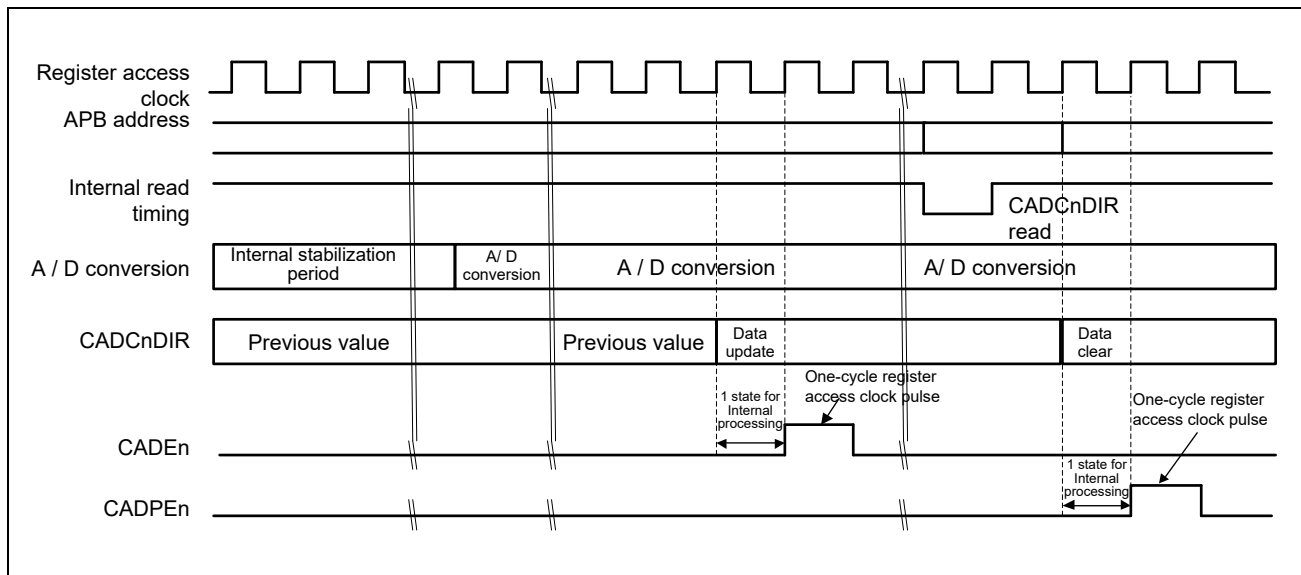


Figure 36.14 Example of AD Error Interrupt and AD Parity Error Interrupt

Upper-limit/lower-limit errors

These errors indicate that the A/D conversion results exceeded the set value.

When the comparison result between the A/D conversion result and the upper-limit/lower-limit table register value (CADCnULTBR0 to CADCnULTBR3.CADCnULMTB[15:0], CADCnLLMTB[15:0]) by CADCnVCULME and CADCnVCLLME is as follows, it is determined to be an error.

A/D conversion result > CADCnULMTB[15:0] (When upper-limit table is enabled: CADCnVCULME = 1)

A/D conversion result < CADCnLLMTB[15:0] (When lower-limit table is enabled: CADCnVCLLME = 1)

Detection of these errors is made each time the A/D conversion result is stored.

Overwrite error

This error indicates that a new A/D conversion result was overwritten to the data register before the data supplementary information register j (CADCNDIRj) was read.

When the write flag (CADCNDIRj.CADCnWFLG) in the data supplementary information register j is 1 and a new A/D conversion result is stored in the relevant data supplementary information register j (CADCNDIRj), an overwrite error is detected.

ID error

This error indicates that the physical channel set by the virtual channel register j (CADCNVCRj) is different from the physical channel upon which A/D conversion was actually performed, that is, this error indicates the occurrence of a malfunction of the analog switch in the I/O buffer.

The ID error detection result is stored in the ID Error bit in the data supplementary information register j (CADCNDIRj) corresponding to the virtual channel (CADCNVCRj) in which an ID error was detected.

The level of the feedback signal from the I/O buffer is checked.

The general control bit (CADCNVCRj.CADCnGCTRL[3:0]) value in the virtual channel register j is compared with the I/O buffer number that is actually activated. If these values do not match, an ID error is detected.

Because the I/O buffer is controlled in a one-hot manner, this error is detected also when the feedback signal is not a one-hot signal.

Either of the following are detected:

- (1) Whether the specified physical channel number does not match the activated I/O buffer number
- (2) Whether differential inputs of two or more channels or single-ended input I/O buffers are simultaneously activated during A/D conversion (one-hot error)

36.6.11 Boundary Flag Output Function

The boundary flag is a level signal that shows whether the input voltage is within the specified voltage range.

The Cyclic ADC outputs upper-limit/lower-limit error pulse signals (CADUEn, CADLEn) and a virtual channel conversion end pulse signal (CADCnVCIj) as boundary flag generating signals. A virtual channel conversion end pulse signal (CADCnVCIj) is always output in spite of CADCnADIE setting when corresponding virtual channel conversion is finished. CADUEn is output when an upper limit error is detected while CADCnVCULME is set to 1. CADLEn is output when a lower limit error is detected while CADCnVCLLME is set to 1. A boundary flag is generated by the ABFG (ADC Boundary Flag Generator). When using the boundary flag, set VCULME and VCLLME in CADCnVCRj to 1. For the ABFG, see **Section 34, Analog to Digital Converter (ADCH)**.

And after being merged*¹, a virtual channel conversion end pulse (CADCnVCIj)*² is output to the GTM as a conversion end pulse (CADCnCADCEP)*³.

Note 1. The merging method is OR.

Note 2. CADCnVCIj is output irrespective of the setting for VCULME and VCLLME in CADCnVCRj.

Note 3. CADCnCADCEP is chosen by the PIC (Peripheral Interconnection), and those are forwarded to the GTM. For details, see **Section 33, Peripheral Interconnection (PIC)**.

36.6.12 Read Gate Function

When CADCnUCR.RDMA is set to 1, CADIn is output only while CADRGTn is asserted.

Figure 36.15 shows a schematic of the DMA request circuit having a read gate from the ATU.

Figure 36.16 shows a schematic of the DMA request circuit having a read gate from the GTM.

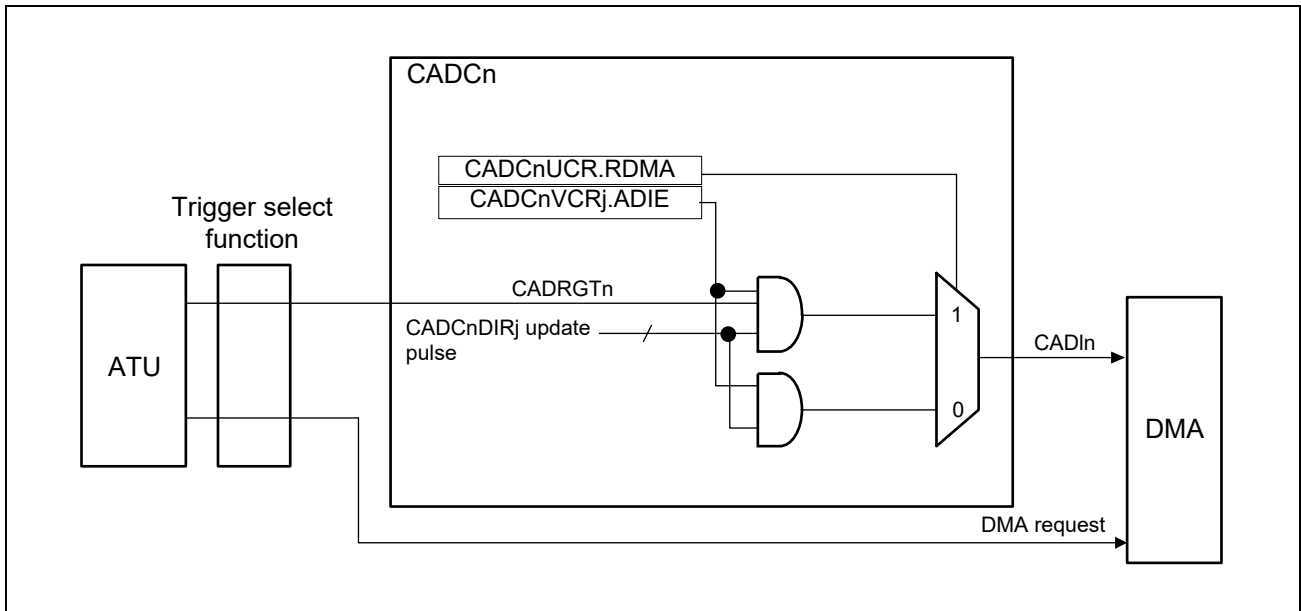


Figure 36.15 DMA Request by CADRGTn from the ATU

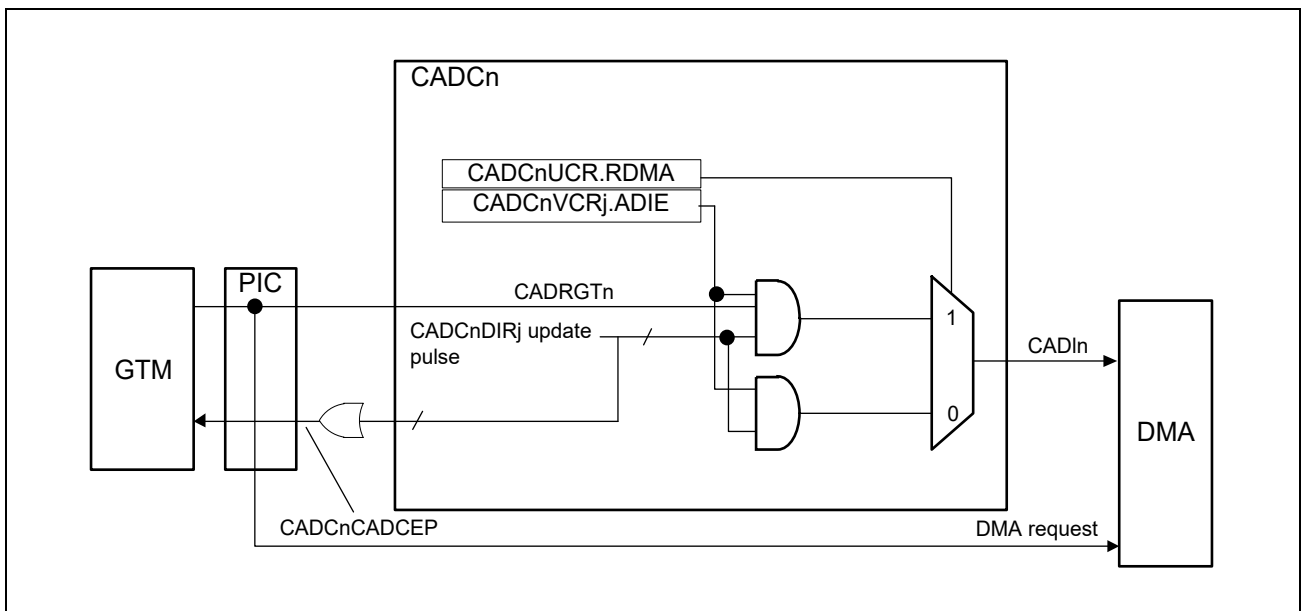


Figure 36.16 DMA Request by CADRGTn from the GTM

36.6.13 Wiring-break Detection Function

The wiring-break detection function is a function that can detect the wiring-breaks for the CAN000P-CAN003P/CAN000N-CAN003N. When A/D conversion is repeated, the conversion result attenuates to approximately 0 V and the abnormal value is detected, which means a wiring-break is detected.

This can be determined as detection of wiring-break.

It is not possible to use the wiring-break detection function simultaneously with normal A/D conversion.

Use a wiring-break detection function before normal A/D conversion is started or between normal A/D conversions.

[Feature]

- 1) Users can select desired physical channels for which wiring-break is to be detected by setting virtual channels.

[Recommended flow]

- 1) Make settings according to the initial value (see **Figure 36.4**)
- 2) Set the CADCnVCRj register for CADCnCNVCLS[1:0] = 0_H and for arbitrary physical channel by CADCnGCTRL[3:0].
- 3) Set the CADCODE bit to 1.
- 4) Execute the A/D conversion with the CADCnADST bit setting.
- 5) After repeating the A/D conversion, read the results of the A/D conversion value. When the value attenuates to approximately 0V, it is detected as a wiring-break.
- 6) Repeat steps 4) and 5) until the conversion is finished for all the channels.

36.6.14 Self-Diagnosis Function

There are three self-diagnosis functions for CADC.

36.6.14.1 Self-Diagnosis of Wiring-break Detection Function

The self-diagnosis function of wiring-break detection function is used to check that the wiring-break detection function is working normally.

Enable the wiring-break detection function and I/O buffer self-diagnosis of CAN000P-CAN003P/CAN000N-CAN003N input by CADCADGCR setting, then perform A/D conversion. When A/D conversion is repeated and the A/D conversion result attenuates to approximately 0V, wiring-break detection works correctly.

[Feature]

- 1) Users can select desired physical channels for which self-diagnosis of wiring-break detection function is performed by setting virtual channels.

[Recommended flow]

- 1) Make settings according to the initial value (see **Figure 36.4**)
- 2) Set the CADCnVCRj register for CADCnCNVCLS[1:0] = 0_H and for arbitrary physical channel by CADCnGCTRL[3:0].
- 3) Set CADCODE and CADCODDE bit to 1.
- 4) Execute the A/D conversion with the CADCnADST bit setting.
- 5) After repeating the A/D conversion, read the result of A/D conversion value. When the value attenuates to approximately 0V, it is detected as a wiring break.
- 6) Repeat steps 4) and 5) until the conversion is finished for all the channels.

36.6.14.2 A/D Converter Self-Diagnosis Function

AD self-diagnosis can check that A/D conversion is working normally. When the A/D conversion result differs from the expecting result, A/D converter have some potential of failure. The voltage value is set by setting CNVCLS[1:0] of virtual channel register j to 3_H and selecting from GCTRL[3:0]. It is possible that A/D conversion of ADSVREFH x 1, ADSVREFH x 1/2, - ADSVREFH x 1/2, - ADSVREFH x 1 or ADSVREFH x 0 is performed.

[Setting]

- 1) Make settings according to the initial value (see **Figure 36.4**)
- 2) Set CADCnCNVCLS[1:0] in the virtual channel register j (CADCnVCRj) to 3_H and set CADCnGCTRL[3:0] to select desired AD self-diagnosis voltage level.
- 3) According to the initial setting flow (see **Figure 36.4**), make the other needed setting for A/D conversion.
- 4) Execute the A/D conversion with the CADCnADST bit setting.

36.6.14.3 Pin Level Self-Diagnosis Function

Pin level self-diagnosis can check the analog input path with an A/D conversion by inputting the different voltage levels of each physical channel. When the A/D conversion result is not static, A/D converter have some potential of failure.

[Analog Input Process]

I/O Buffer to A/D Converter

[Setting]

Make the settings for A/D control register (CADCADGCR), virtual channel register j (CADCnVCRj), pin level self-diagnosis control register (CADCTDCR) and pin level self-diagnosis level register (CADCnTDLVR) according to the initial settings.

- (1) CADCADGCR

CADCODDE	CADCODE
1	0

- (2) CADCTDCR

CADCTDE
1

(3) CADCnVCRj

VCR Number	CADCnCNVCLS[1:0]	CADCnGCTRL[3:0]
N	0 _H , 1 _H , 2 _H	Physical Channel Number

(4) CADCnTDLVR

AN0PLV/AN0NLV AN1PLV/AN1NLV AN2PLV/AN2NLV AN3PLV/AN3NLV	Injecting Voltage of Physical Channel
0	ADSVSS
1	ADSVCC

Execute the A/D conversion with the A/D conversion flow.

Note 1: Set CADCODDE bit to 1 before setting the CADCTDE bit to 1.

Note 2: When CADCTDE = 1, do not set CADCODE bit to 1.

36.6.15 DFE Control

Entry to DFE or GTM is enabled by setting the ADCnDFENT bit in the ADCnVCRj register to 1.

For allocation of each ADCn and each channel of DFE, see **Section 37, Digital Filter Engine (DFE)**.

For allocation of each ADCn and each channel of GTM, see **Section 31, Generic Timer Module (GTM)**.

36.7 Usage Notes

36.7.1 Notes on Using Analog Input Pins

Do not perform an A/D conversion on the value of the same analog pin simultaneously with the Cyclic ADC and SAR-ADC. Also, do not allow the Cyclic ADC to perform A/D conversion on the value of an analog pin currently selected for digital input. This may diminish A/D accuracy.

When the Cyclic ADC is used with a single-ended input, and only the analog input pins on the P side (or N side) of a differential input are used, the analog input pins on the N side (or P side) can be used for other ADC inputs.

36.7.2 Module Standby Function

The Cyclic ADC has a module standby function.

Module standby stops the clock supply to CADC00 by setting MSRCAD.MS_CAD.

Module standby stops the clock supply to CADC (common). For the clock supply stop condition of the CADC (common), see **Table 36.34**.

Table 36.34 Clock Supply Stop Condition of CADC (common) by Module Standby

States	MSR_DAD. MS_DAD00_10	MSR_DAD. MS_DAD20_12	MSR_DAD. MS_DAD13_11	MSR_CAD. MS_CAD
Clock Supply	0	0 or 1	0 or 1	0 or 1
Clock Supply	1	0	0 or 1	0 or 1
Clock Supply	1	1	0	0 or 1
Clock Supply	1	1	1	0
Clock Supply Stop	1	1	1	1

To prevent a malfunction, do not set module standby mode during A/D conversion or during calibration or while A/D conversion start/end trigger is enabled. Before setting module standby mode, set the related channels as follows, and wait for 10 states of the register access clock.

- CADCnADSR.CADCnADACT = 0
- CADCnADSR.CADCnCLBACT = 0
- CADCnADTCR.CADCnENDTRGE = 0
- CADCnADTCR.CADCnSTTRGE = 0

Section 37 Digital Filter Engine (DFE)

This section contains a description of the Digital Filter Engine (DFE) and DFE FIFO.

The first part in this section describes the features specific to this product, including register base addresses and input/output signals. The subsequent sections describe the functions and registers of DFE and DFE FIFO.

37.1 Features

37.1.1 Units and Channels

This microcontroller has following number of units and channels for the Digital Filter Engine (DFE) and DFE FIFO.

Table 37.1 Number of Units for DFE

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Units	2	2
Unit Name	DFEj (j = 0,1)	DFEj (j = 0,1)

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Units	2	2
Unit Name	DFEj (j = 0,1)	DFEj (j = 0,1)

Table 37.2 Number of Units for DFE FIFO

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Units	1	1
Unit Name	DFE FIFO	DFE FIFO

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Units	1	1
Unit Name	DFE FIFO	DFE FIFO

Table 37.3 Number of Channels

DFEj	Number of Channels
DFE0	12
DFE1	4

Table 37.4 Index

Index	Description
j	The individual DFE units are identified by the index "j" (j = 0, 1).

37.1.2 Register Base Address

The DFE and DFE FIFO base addresses are listed in the following table.

The DFE and DFE FIFO register addresses are given as offsets from the base address.

Table 37.5 Register Base Address

Base Address Name	Base Address	Bus Group
<DFE0_base>	FFBF 0000 _H	Peripheral Group 5
<DFE1_base>	FFBF 4000 _H	Peripheral Group 5
<DFE_FIFO_base>	FFBF 8000 _H	Peripheral Group 5

37.1.3 Clock Supply

The DFE and DFE FIFO clock supplies are shown in the following table.

Table 37.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
DFE _j	PCLKHIN	CLK_UHSB
	PCLKLIN	CLK_HSB
DFE FIFO	PCLKLIN	CLK_HSB

Note: j = 0, 1

37.1.4 Interrupt Requests

The DFE and DFE FIFO interrupt requests are listed in the following table.

Table 37.7 Interrupt Requests (1/2)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
DFE0					
INTDFE0DOUTCND0	INTDFE0DOUT[0]	CH0 output data interrupt	309	group0-64	group0-64
	INTDFE0CND0[0]	CH0 condition match interrupt 0			
INTDFE0DOUTCND1	INTDFE0DOUT[1]	CH1 output data interrupt	310	group0-65	group0-65
	INTDFE0CND0[1]	CH1 condition match interrupt 0			
INTDFE0DOUTCND2	INTDFE0DOUT[2]	CH2 output data interrupt	311	group0-66	group0-66
	INTDFE0CND0[2]	CH2 condition match interrupt 0			
INTDFE0DOUTCND3	INTDFE0DOUT[3]	CH3 output data interrupt	312	group0-67	group0-67
	INTDFE0CND0[3]	CH3 condition match interrupt 0			
INTDFE0DOUTCND4	INTDFE0DOUT[4]	CH4 output data interrupt	313	group0-68	group0-68
	INTDFE0CND0[4]	CH4 condition match interrupt 0			
INTDFE0DOUTCND5	INTDFE0DOUT[5]	CH5 output data interrupt	314	group0-69	group0-69
	INTDFE0CND0[5]	CH5 condition match interrupt 0			
INTDFE0DOUTCND6	INTDFE0DOUT[6]	CH6 output data interrupt	315	group0-70	group0-70
	INTDFE0CND0[6]	CH6 condition match interrupt 0			
INTDFE0DOUTCND7	INTDFE0DOUT[7]	CH7 output data interrupt	316	group0-71	group0-71
	INTDFE0CND0[7]	CH7 condition match interrupt 0			
INTDFE0DOUTCND8	INTDFE0DOUT[8]	CH8 output data interrupt	317	group0-72	group0-72
	INTDFE0CND0[8]	CH8 condition match interrupt 0			
INTDFE0DOUTCND9	INTDFE0DOUT[9]	CH9 output data interrupt	318	group0-73	group0-73
	INTDFE0CND0[9]	CH9 condition match interrupt 0			
INTDFE0DOUTCND10	INTDFE0DOUT[10]	CH10 output data interrupt	319	group0-74	group0-74
	INTDFE0CND0[10]	CH10 condition match interrupt 0			
INTDFE0DOUTCND11	INTDFE0DOUT[11]	CH11 output data interrupt	320	group0-75	group0-75
	INTDFE0CND0[11]	CH11 condition match interrupt 0			
INTDFE0CND10	INTDFE0CND1[0]	CH0 condition match interrupt 1	322	group0-76	group0-76
INTDFE0CND11	INTDFE0CND1[1]	CH1 condition match interrupt 1	323	group0-77	group0-77
INTDFE0CND12	INTDFE0CND1[2]	CH2 condition match interrupt 1	324	group0-78	group0-78
INTDFE0CND13	INTDFE0CND1[3]	CH3 condition match interrupt 1	325	group0-79	group0-79
INTDFE0ERR	INTDFE0ERR	Error interrupt	321	—	—
INTDFE0SUBOUT0	INTDFE0SUBOUT[0]	Subtraction data interrupt 0	326	group0-80	group0-80
INTDFE0SUBOUT1	INTDFE0SUBOUT[1]	Subtraction data interrupt 1	327	group0-81	group0-81
INTDFE0SUBOUT2	INTDFE0SUBOUT[2]	Subtraction data interrupt 2	328	group0-82	group0-82

Table 37.7 Interrupt Requests (2/2)

Interrupt Symbol Name	Unit Interrupt Signal	Outline	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
DFE1					
INTDFE1DOUTCND0	INTDFE1DOUT[0]	CH0 output data interrupt	329	group0-83	group0-83
	INTDFE1CND0[0]	CH0 condition match interrupt 0			
INTDFE1DOUTCND1	INTDFE1DOUT[1]	CH1 output data interrupt	330	group0-84	group0-84
	INTDFE1CND0[1]	CH1 condition match interrupt 0			
INTDFE1DOUTCND2	INTDFE1DOUT[2]	CH2 output data interrupt	331	group0-85	group0-85
	INTDFE1CND0[2]	CH2 condition match interrupt 0			
INTDFE1DOUTCND3	INTDFE1DOUT[3]	CH3 output data interrupt	332	group0-86	group0-86
	INTDFE1CND0[3]	CH3 condition match interrupt 0			
INTDFE1CND10	INTDFE1CND1[0]	CH0 condition match interrupt 1	334	—	—
INTDFE1CND11	INTDFE1CND1[1]	CH1 condition match interrupt 1	335	—	—
INTDFE1CND12	INTDFE1CND1[2]	CH2 condition match interrupt 1	336	—	—
INTDFE1CND13	INTDFE1CND1[3]	CH3 condition match interrupt 1	337	—	—
INTDFE1ERR	INTDFE1ERR	Error interrupt	333	—	—
INTDFE1SUBOUT0	INTDFE1SUBOUT[0]	Subtraction data interrupt 0	338	group0-87	group0-87
INTDFE1SUBOUT1	INTDFE1SUBOUT[1]	Subtraction data interrupt 1	339	group0-88	group0-88
INTDFE1SUBOUT2	INTDFE1SUBOUT[2]	Subtraction data interrupt 2	340	group0-89	group0-89
DFE_FIFO					
INTDFEFIFOOUTA	INTDFEFIFOOUTA	Buffer-A capture finished interrupt request	341	group0-90	group0-90
INTDFEFIFOOUTB	INTDFEFIFOOUTB	Buffer-B capture finished interrupt request	342	group0-91	group0-91
INTDFEFIFOERR	INTDFEFIFOERR	Error interrupt	321	—	—

NOTE

INTDFEFIFOERR is merged with INTDFE0ERR

37.1.5 Reset Sources

The DFE and DFE FIFO reset sources are listed in the following table. DFE and DFE FIFO are initialized by these reset sources.

Table 37.8 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
DFE0	All registers	✓	✓	✓	✓	—	—
DFE1	All registers	✓	✓	✓	✓	—	—
DFE FIFO	All registers	✓	✓	✓	✓	—	—

37.1.6 External Input/Output Signals

DFE_j (j = 0, 1) and DFE FIFO have no external input/output signals.

37.2 Overview

37.2.1 Features

- Incorporates a 12- or 4-channel FIR and IIR digital filter unit in a DFE IP (A 64-tap FIR filter can be selected only with even-numbered channels).
- Filter coefficients and data are stored in the on-chip RAM.
- Incorporates an accumulation circuit (for accumulation processing or decimation processing for filtered data) and a peak-hold circuit (for peak hold processing or comparison processing).
- Subtraction circuit is incorporated for subtraction on output data from two channels.
- Peak-Hold circuit is incorporated for peak holding and comparison
- With the peak-holding function, either upper peaks or lower peaks can be selected.
- On a single arbitrary channel, three different peak values can be detected and held during a single Peak-Hold process.
- Generates an interrupt request if the comparison calculation result is true.
- Inputs converted data from the on-chip A/D converter*1.
- Two types of FIFOs can be used: 6-input 8-stage buffer and 1-input 8-stage buffer.

Note 1. In this section, ADCH, DSADC and CADC are called SAR-ADC, DS-ADC and C-ADC, respectively.

37.2.2 Overall Configuration

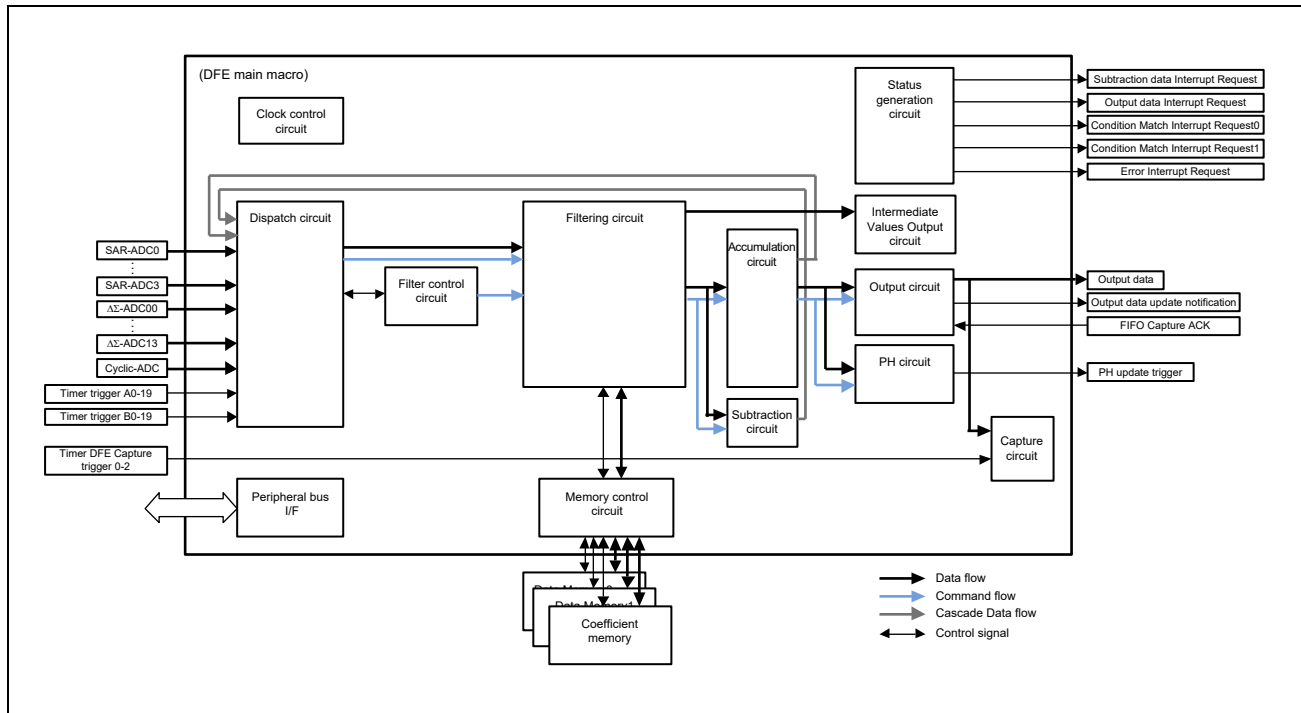


Figure 37.1 Overall Configuration of the DFE

The digital filter engine (DFE) is able to execute 16-channel filter processing on a time-division basis. FIR (up to 64 taps) or IIR (up to 6 stages) is selectable as a filter algorithm. When data is transferred from the A/D converter via private bus or is written to the software data input register through peripheral bus, the DFE executes FIR/IIR filter processing automatically. In normal mode, all filter processing results are stored in the output data register. In decimation mode, output data is stored in the output data register once every fourth filter processing cycle, for example. When data is stored in the output data register, an interrupt request can be issued. Furthermore, the digital filter processing result can be input again to the filtering circuit. This process is referred to as a cascade input in these specifications.

The DFE can execute subtraction on the values of the output data registers of any two channels. The subtraction result is stored in the register as subtraction output data, similar to the output data. When data is stored in the subtraction output data register, an interrupt can be requested. The subtraction result can also be selected as a cascade input.

37.2.3 Overall Configuration of the FIFO Macro

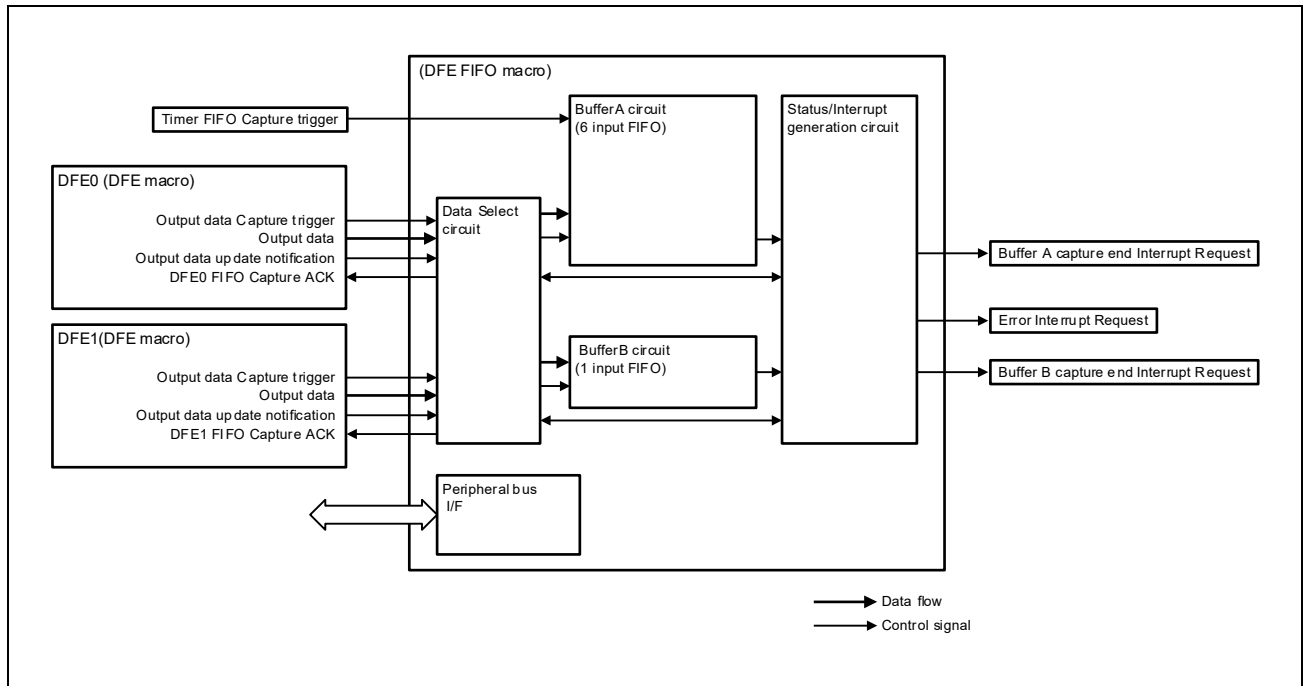


Figure 37.2 Overall Configuration of the FIFO Macro

With the FIFO macro, it is possible to select data from up to 12 DFE0 channels and 4 DFE1 channels for a 6-input 8-stage buffer (hereinafter referred to as buffer A) or a 1-input 8-stage buffer (hereinafter referred to as buffer B), and store the data in the FIFO.

(1) Buffer A Circuit

- Simultaneously stores six items of output data from DFE channels in the FIFO, in response to the capture trigger output from the timer. The Buffer A circuit has FIFO channels to store output data. The Buffer A circuit can have a maximum number of six FIFO channels
- Issues the buffer A capture end interrupt request after storing the valid data in the FIFO. For details of the interrupt request issue timing, refer to **Section 37.5.27.2(1), Buffer A Capture End Interrupt Request**.
- Issues the error interrupt request caused by an overflow error if more than 8-stage data is stored in the FIFO.

(2) Buffer B Circuit

- Automatically stores data in the FIFO each time the data of the target channel is updated.
- Issues the buffer B capture end interrupt request after storing the data in the FIFO four times. For details of the interrupt request issue timing, refer to **Section 37.5.27.2(2), Buffer B Capture End Interrupt Request**.
- Issues the error interrupt request caused by an overflow error if more than 8-stage data is stored in the FIFO.

37.2.4 Data Format

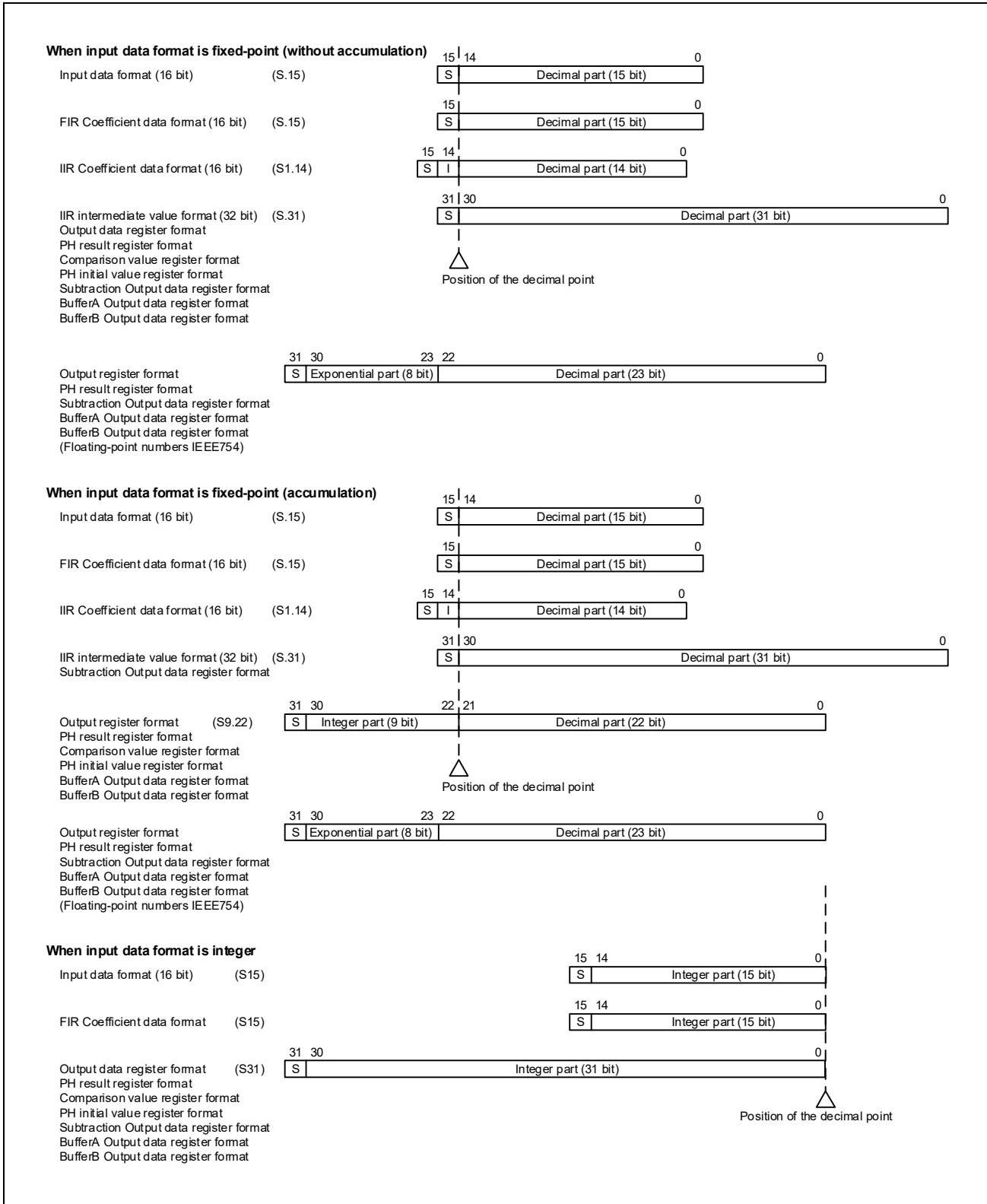


Figure 37.3 Data Format

37.2.5 Filtering Circuit

37.2.5.1 FIR Filter

- Number of taps: Selectable from 8, 16, 24, 32, and 64
- Input data and coefficients: 16-bit signed fixed-point (S.15) /16-bit integer (S15)
- Intermediate values: 38-bit signed fixed-point data
- Output data: 32-bit signed fixed-point/32-bit integer
- Product-sum operation accuracy: 32 bits
- Filter processing cycles: 64 taps (no initialization) takes 50 cycles.

37.2.5.2 IIR Filter

- Number of secondary biquad stages: Selectable from 1, 2, and 3 (in the case of secondary biquad 2 or 3 stages, the gain adjustment function is selectable).
- Input data: 16-bit signed fixed-point data (S.15)
- Coefficient data format: 16-bit signed fixed-point data (S1.14)
- Intermediate data and output data: 32-bit signed fixed-point data
- Product-sum operation accuracy: 32 bits
- Filter processing cycles: Secondary biquad 3 stage (no initialization) takes 44 cycles.

37.2.6 Dispatch Circuit

The dispatch circuit accepts data to be filtered from the A/D converter and peripheral bus. It assigns the accepted data to sixteen channels, then selects data to be filtered and outputs that data to the filter circuit. The data that can be input is specified for each channel. **Table 37.9** lists the data that can be input for each channel.

Table 37.9 Input Data for Each Channel

AD unit Number		SAR-ADC(ADCH)				DS-ADC(DSADC)						C-ADC(CADC)	Cascade	SW
		0	2	1	3	00	10	20	11	12	13	0	0	0
DFE0	CH0	✓	✓	✓	×	✓	✓	✓	✓	×	×	×	✓	✓
	CH1	✓	✓	✓	×	✓	✓	✓	×	✓	×	×	✓	✓
	CH2	✓	✓	✓	×	✓	✓	✓	×	×	✓	×	✓	✓
	CH3	✓	✓	✓	×	✓	✓	✓	×	×	×	×	✓	✓
	CH4	✓	✓	✓	×	✓	✓	✓	×	×	×	×	✓	✓
	CH5	✓	✓	✓	×	✓	✓	✓	✓	×	×	×	✓	✓
	CH6	✓	✓	×	✓	✓	✓	✓	×	✓	×	×	✓	✓
	CH7	✓	✓	×	✓	✓	✓	✓	×	×	✓	×	✓	✓
	CH8	✓	✓	×	✓	✓	✓	✓	×	×	×	×	✓	✓
	CH9	✓	✓	×	✓	✓	✓	✓	×	×	×	×	✓	✓
	CH10	✓	✓	×	✓	✓	✓	✓	✓	×	×	×	✓	✓
CH11	✓	✓	×	✓	✓	✓	✓	×	✓	×	×	✓	✓	
DFE1	CH0	×	×	×	×	✓	✓	✓	✓	×	×	✓	✓	✓
	CH1	×	×	×	×	✓	✓	✓	×	×	×	✓	✓	✓
	CH2	×	×	×	×	✓	✓	✓	×	×	×	✓	✓	✓
	CH3	×	×	×	×	✓	✓	✓	×	×	×	✓	✓	✓

Note: ✓: Data can be input; ×: Data cannot be input

Please refer to Section ADCH, DSADC about useable AD unit.

37.2.7 Accumulation Circuit

The accumulation circuit allows absolute value calculation, accumulation, and decimation processing for filtering results of all channels. Either accumulation or decimation can be specified.

- **Absolute value:** The absolute value of each filtering result is calculated.
- **Accumulation:** Accumulation of filtering results can be obtained. The maximum accumulation count is 511. Filtering results are of the S9.22 format. After accumulation processing for the predetermined number of times has been completed, one accumulation result is sent to the next processing step.
- **Decimation:** Filtering results are decimated. If the decimation count is set to 4, for example, the filtering result is output once for every five filter processes. The maximum decimation count is 511.

37.2.8 Subtraction Circuit

The subtraction circuit can execute subtraction on the filtering results of any two channels.

The subtraction circuit outputs the subtraction output data when the filtering results of both the minuend and subtrahend channels are updated. The subtraction results are stored in the subtraction output data registers for each channel. When the data of any two channels is both input in the fixed-point format, the format of the subtraction output data can be converted to the floating-point format (IEEE754).

$$[\textit{Subtraction output data}] = [\textit{Minuend channel filtering result} - \textit{Subtrahend channel filtering result}]$$

37.2.9 Peak-Hold (PH) Circuit

The Peak-Hold circuit has the peak-holding function and a comparison function. With the Peak-Hold circuit, it is possible to specify and execute one of the peak-holding and comparison functions; and it is also possible to select both the peak-holding and comparison functions simultaneously.

The peak-hold circuit allows peak hold or comparison calculation for all channels. The peak-hold circuit can be linked to the accumulation circuit. For example, when accumulation/decimation is set with a specified count, peak hold processing or comparison calculation is executed on each specified count. When accumulation/decimation is not set, peak-hold processing or comparison calculation is executed on each filter processing cycle.

- **Peak-Hold (PH):** The peaks of filter results or the accumulation circuit output results can be held. Either the upper or lower peaks can be selected. A Peak-Hold update notification can be issued when the Peak-Hold result register of the selected channel is updated.

NOTE

The table below shows the conditions for updating the Peak-Hold result registers for the upper or lower peaks and the conditions for outputting the Peak-Hold update notification.

Peak	Function	Number Times of PH	Issue Condition
Upper peaks	PH update notification	First time* ¹	PH circuit input data* ² > The value of the peak-hold initial value setting register* ³
		After the first time	PH circuit input data* ² > The value of the DFEjPHCHn* ⁴
Lower peaks	PH update notification	First time* ¹	PH circuit input data* ² < The value of the peak-hold initial value setting register* ³
		After the first time	PH circuit input data* ² < The value of the DFEjPHCHn* ⁴

Peak	Function	Number Times of PH	Update Condition	Stored Data in DFEjPHCHn
Upper peaks	Peak-Hold Result register	First time* ¹	PH circuit input data* ² > The value of the peak-hold initial value setting register* ³	PH circuit input data
			PH circuit input data* ² ≤ The value of the peak-hold initial value setting register* ³	The value of the peak-hold initial value setting register* ³
		After the first time	PH circuit input data* ² > The value of the DFEjPHCHn* ⁴	PH circuit input data
			PH circuit input data* ² ≤ The value of the DFEjPHCHn* ⁴	Holds the value of DFEjPHCHn
Lower peaks	Peak-Hold Result register	First time* ¹	PH circuit input data* ² < The value of the peak-hold initial value setting register* ³	PH circuit input data
			PH circuit input data* ² ≥ The value of the peak-hold initial value setting register* ³	The value of the peak-hold initial value setting register* ³
		After the first time	PH circuit input data* ² < The value of the DFEjPHCHn* ⁴	PH circuit input data
			PH circuit input data* ² ≥ The value of the DFEjPHCHn* ⁴	Holds the value of DFEjPHCHn

Note 1. The first time means when filter processing is executed after PH initialization flag is valid.

Note 2. PH input data is the output data of the filter circuit or accumulation/decimation circuit.

Note 3. The register means one of the DFEjPHIA-DFEjPHID registers.

Note 4. These values are compared in either fixed-point format or integer format regardless of the DFEjCTLBCHn.PFMT bit setting.

- **Peak-hold index:** Indicates the round of peak-hold processing in which the peak-hold result register was updated to hold its current value.
- **Comparison:** A filter result or accumulation circuit output result is compared with the predetermined value (=, <, >, ≥, ≤) and an interrupt can be requested according to the condition matched.

The comparison value can be selected as follows.

Selectable Values to be Compared	Comparison Selected Exclusively	PH processing and Comparison Selected Simultaneously	PH23 Function
Comparison value registers A to D	OK	OK	NG* ²
PH result register + α* ¹		NG* ²	OK

Note 1. α is the value of the comparison offset value register.

Note 2. Setting the NG conditions is prohibited.

- **PH23 function:** Expansion of the PH processing. Three peaks can be held for a 1-channel filtering result or accumulation circuit output result. A PH update notification can be separately issued when the PH result register for each peak is updated.

37.2.10 Output Circuit

In the output circuit, calculation results are stored in the output data register for each channel. When accumulation or decimation is specified, an interrupt can be requested at the same time data is stored. When fixed-point format data is input, the data format can be converted to the floating-point (IEEE754) format and the data can be stored in the output data register.

37.2.11 Intermediate Values Output Circuit

Even number channels have an intermediate value output register, which holds intermediate values (values from filter calculations prior to processing for rounding) from the intermediate value output circuit. When FIR filtering is selected, the values are input in fixed-point format. Conversion to fixed or floating point (IEEE754) format for storage is available.

37.2.12 Capture Circuit

The capture circuit stores the values of the output data register in the capture result register upon receiving the output data capturing trigger. The format of the captured data conforms to the format of the output data register of the selected channel.

37.2.13 Buffer A Circuit

The buffer A circuit, incorporated in the FIFO macro, selects up to six channels from the DFE0 and DFE1 channels and stores the values of the output data registers in the FIFO, in response to the FIFO capture triggers. The format of the output data stored in buffer A conforms to the format of the output data register of the selected channel.

37.2.14 Buffer B Circuit

The buffer B circuit, incorporated in the FIFO macro, selects one channel from the DFE0 and DFE1 channels and automatically stores the values of the output data registers in the FIFO, each time the output data register is updated. The format of the output data stored in buffer B conforms to the format of the output data register of the selected channel.

37.2.15 Terms

Table 37.10 provides definitions of terms used in this section.

Table 37.10 List of Terms (1/3)

Term	Definition in this Section
DFE operating clock	160 MHz clock to be input to the DFE
Filtering circuit	A circuit to perform FIR filtering and IIR filtering
Data to be processed	Data for which the DFE performs filter processing, including data to be input by the AD and peripheral bus and filter processing result re-entry (cascade) data
Software input	Activation of the DFE by writing data to be processed to a register from the CPU or DMA through the peripheral bus
Input data register	A register for writing data to be processed, which is provided for 16 channels
DFE activation signal	DFE processing start signal to be input to the DFE together with data to be processed
AD tag	A tag value to be input to the DFE by the AD together with data to be processed
AD data	Data to be processed and input to the DFE by the AD
AD input	AD tag or AD data or DFE activation signal to be input by the AD In figures shown in this section, AD input shows DFE activation signal to be input by the AD.
Channel tag	A tag signal specified by the DFE's control register A, which is provided for each channel
Dispatch	Writing data to be processed to an input data register Dispatch means writing data to the input data register of the matched channel through comparison between channel tag and AD tag signal after processing is started by the DFE activation signal.
Dispatch circuit	A circuit to output filtering target data to the dispatch processing and filtering circuit
Accumulation	Accumulation of filter processing result data in each channel for the count specified in the control register
Decimation	Decimation (thinning) of filter processing result data for the count specified in the control register
Accumulation circuit	A circuit to perform absolute value calculation, accumulation, and decimation for filter processing result data
Subtraction	Subtraction on two arbitrary channels
Subtraction circuit	Circuit that performs subtraction on the filtering result data of two channels for subtraction
Subtraction result register	Register that holds subtraction result values
Subtrahend flag	Subtrahend channel data input enabling bit (SUBF)
Minuend flag	Minuend channel data input enabling bit (MINF)
Peak-hold processing	Processing to hold the maximum or minimum of output data from the accumulation circuit
Peak-hold circuit	A circuit to perform peak-hold processing and comparison processing
Peak-hold result register	A register that retains the peak-hold processing result value
Peak-hold index register	A register that indicates the number of times from the beginning of peak-hold processing when the peak-hold result register value is input to the peak-hold circuit
Output circuit	A circuit to write accumulation circuit output data to the output data register and to convert accumulation circuit output data to floating-point format data and write the data to the output data register
Output data register	A register used to write data after DFE processing is completed, which is provided for 16 channels
Intermediate value output circuit	Circuit that writes the intermediate values output from the filtering circuit to the intermediate value output data register when FIR is selected.
Intermediate value output register	Register to which the intermediate value data is written.
Capture circuit	Circuit that writes the value of the output data register of the specified channel to the capture result register when a capture trigger is input.
Capture result register	Register to which the value of the specified channel output data register is written.
Buffer A circuit	Circuit that selects up to six channels from the DFE0 and DFE1 channels and stores the values of the output data registers in the FIFO, in response to the FIFO capture triggers.
Buffer B circuit	Circuit that selects one channel from the DFE0 and DFE1 channels and automatically stores the values of the output data registers in the FIFO, each time the output data is updated.

Table 37.10 List of Terms (2/3)

Term	Definition in this Section
Input data overwrite error (DIOW)	An error that occurs when new data is written to the input data register while valid data is contained in the input data register (filter processing wait state in the DFE, during filter processing, or during accumulation circuit processing)
Output data overwrite error (DOOW)	An error that occurs when new data is written to the output data register while valid data is contained in the output data register (after the DFE has written filter processing result or accumulation processing result to the output data register)
Multiplication error	An error that occurs during execution of $8000_H \times 8000_H$ in the filtering circuit
Guard error	A guard bit overflow or underflow error that occurs during execution of accumulation in the filtering circuit
Cascade error	An overflow error that occurs when the 32-bit accumulation circuit output data is rounded to 16-bit data when cascade processing is enabled
Absolute value calculation error	An error generated when $8000\ 0000_H$ is input during calculation of absolute values in the accumulation circuit
Timer trigger	Triggers to which timer trigger A/B from the timer (GTM or ATU) were allotted as the function of "PH initialization", "PH Mask Start/End", "Addition/Decimation Initialization/Prohibition" or "Subtraction Start/End". Please see section PIC or ATU.
Software trigger	A trigger generated by writing 1 to the software trigger register from the CPU or DMA
Trigger flag	There are nine flags: accumulation/decimation initialization flag, accumulation/decimation disable flag, peak-hold initialization flag, peak-hold end flag, peak-hold mask start flag, peak-hold mask end flag, subtraction starting flag, subtraction ending flag, and filter initialization flag, which are generated by the DFE at a timer trigger, software trigger, or trigger setting register (control register in the DFE)
Accumulation/decimation initialization flag	A trigger flag that functions as an accumulation initialization flag when accumulation is specified or as a decimation initialization flag when decimation is specified, which is used to initialize the counter in the accumulation circuit.
Accumulation/decimation disable flag	A trigger flag that functions as an accumulation disable flag when accumulation is specified or as a decimation disable flag when decimation is specified, which stops the counter in the accumulation circuit and disables accumulation processing and decimation processing without updating output data after execution
Peak-hold(PH) initialization flag	A trigger flag that is enabled when peak-hold (PH) operation is specified, which is used to initialize the counter in the peak-hold (PH) circuit.
Peak-hold(PH) end flag	A trigger flag that is enabled when peak-hold (PH) operation is specified, which is a command to terminate peak-hold (PH) processing
Peak-hold (PH) mask starting flag	One of the trigger flags. Enabled when PH is specified. Stops a PH process when a PH operation is in progress.
Peak-hold (PH) mask ending flag	One of the trigger flags. Enabled when PH is specified. Restarts the PH process that has been stopped by the PH mask starting flag.
Filter initialization flag	A trigger flag to performed adding the initialization sequence to the initialization for the address pointer used in filter processing and normal filter processing.
Subtraction starting flag	One of the trigger flags. Enabled when the subtraction circuit is enabled, and starts a subtraction process.
Subtraction ending flag	One of the trigger flags. Enabled when the subtraction circuit is enabled, and ends a subtraction process.
Interrupt request	An interrupt to request activation of the interrupt controller, CPU, or DMA.
Buffer A	Buffer A circuit has 6 number of 8-stage FIFOs. For each of the six inputs, data of different DFE channels can be separately captured.
Buffer B	Incorporates one 8-stage FIFO. Data of one DFE channel can be selected.
FIFO channels	The name each peculiar to up to 6 FIFO which buffer A circuit has is defined. This is called FIFO channels.
FIFO empty	State in which no data is stored in the FIFO buffer. Specifically, the FIFO empty state is entered immediately after the FIFO is reset, and when data having been captured in the FIFO buffer is read from all the stages.
FIFO full	State in which data is stored in all of the eight stages of the FIFO buffer.
SCH	Subtraction channels used for the subtraction circuit, three channels are supported.
CCH	Capture channels used for the capture circuit. Three channels are supported.

Table 37.10 List of Terms (3/3)

Term	Definition in this Section
FCH	FIFO channels used for the buffer A circuit six channels are supported.
Timer trigger A	Triggers that are input from the timer. In this section, these are called "Timer triggerA", "compare A" or "compare match A".
Timer trigger B	Triggers that are input from the timer. In this section, these are called "Timer triggerB", "compare B" or "compare match B".
Timer DFE Capture trigger	Triggers that are input from the timer. In this section, these are called "Timer DFE Capture trigger" or "DFE Timer capture trigger".
Timer FIFO Capture trigger	A trigger that is input from the timer.
PH update notification	Triggers that are output to the timer. In this section, these are called "PH update trigger" or "PH update notification".

37.3 Control Registers

37.3.1 List of Registers for DFE Macro

Table 37.11 lists registers and memory addresses of the DFE macro. 8, 16 or 32-bit write access to all registers is possible. 32-bit read access to all registers is possible. 32-bit write access to the coefficient memory is possible, and upper 16-bit or lower 16-bit or 32-bit write access to the data memory is possible.

Table 37.11 List of Registers for DFE Macro (1/8)

Module Name	Register Name	Symbol	Address	Access Size	Access Protect
DFE0					
DFE0	Control Register A (Channel 0)	DFE0CTLACH0	<DFE0_base> + 0000 _H	8, 16, 32	—
DFE0	Control Register A (Channel 1)	DFE0CTLACH1	<DFE0_base> + 0004 _H	8, 16, 32	—
DFE0	Control Register A (Channel 2)	DFE0CTLACH2	<DFE0_base> + 0008 _H	8, 16, 32	—
DFE0	Control Register A (Channel 3)	DFE0CTLACH3	<DFE0_base> + 000C _H	8, 16, 32	—
DFE0	Control Register A (Channel 4)	DFE0CTLACH4	<DFE0_base> + 0010 _H	8, 16, 32	—
DFE0	Control Register A (Channel 5)	DFE0CTLACH5	<DFE0_base> + 0014 _H	8, 16, 32	—
DFE0	Control Register A (Channel 6)	DFE0CTLACH6	<DFE0_base> + 0018 _H	8, 16, 32	—
DFE0	Control Register A (Channel 7)	DFE0CTLACH7	<DFE0_base> + 001C _H	8, 16, 32	—
DFE0	Control Register A (Channel 8)	DFE0CTLACH8	<DFE0_base> + 0020 _H	8, 16, 32	—
DFE0	Control Register A (Channel 9)	DFE0CTLACH9	<DFE0_base> + 0024 _H	8, 16, 32	—
DFE0	Control Register A (Channel 10)	DFE0CTLACH10	<DFE0_base> + 0028 _H	8, 16, 32	—
DFE0	Control Register A (Channel 11)	DFE0CTLACH11	<DFE0_base> + 002C _H	8, 16, 32	—
DFE0	Control Register B (Channel 0)	DFE0CTLBCH0	<DFE0_base> + 0040 _H	8, 16, 32	—
DFE0	Control Register B (Channel 1)	DFE0CTLBCH1	<DFE0_base> + 0044 _H	8, 16, 32	—
DFE0	Control Register B (Channel 2)	DFE0CTLBCH2	<DFE0_base> + 0048 _H	8, 16, 32	—
DFE0	Control Register B (Channel 3)	DFE0CTLBCH3	<DFE0_base> + 004C _H	8, 16, 32	—
DFE0	Control Register B (Channel 4)	DFE0CTLBCH4	<DFE0_base> + 0050 _H	8, 16, 32	—
DFE0	Control Register B (Channel 5)	DFE0CTLBCH5	<DFE0_base> + 0054 _H	8, 16, 32	—
DFE0	Control Register B (Channel 6)	DFE0CTLBCH6	<DFE0_base> + 0058 _H	8, 16, 32	—
DFE0	Control Register B (Channel 7)	DFE0CTLBCH7	<DFE0_base> + 005C _H	8, 16, 32	—
DFE0	Control Register B (Channel 8)	DFE0CTLBCH8	<DFE0_base> + 0060 _H	8, 16, 32	—
DFE0	Control Register B (Channel 9)	DFE0CTLBCH9	<DFE0_base> + 0064 _H	8, 16, 32	—
DFE0	Control Register B (Channel 10)	DFE0CTLBCH10	<DFE0_base> + 0068 _H	8, 16, 32	—
DFE0	Control Register B (Channel 11)	DFE0CTLBCH11	<DFE0_base> + 006C _H	8, 16, 32	—
DFE0	Output Data Register (Channel 0)	DFE0DOCH0	<DFE0_base> + 0080 _H	32	—
DFE0	Output Data Register (Channel 1)	DFE0DOCH1	<DFE0_base> + 0084 _H	32	—
DFE0	Output Data Register (Channel 2)	DFE0DOCH2	<DFE0_base> + 0088 _H	32	—
DFE0	Output Data Register (Channel 3)	DFE0DOCH3	<DFE0_base> + 008C _H	32	—
DFE0	Output Data Register (Channel 4)	DFE0DOCH4	<DFE0_base> + 0090 _H	32	—
DFE0	Output Data Register (Channel 5)	DFE0DOCH5	<DFE0_base> + 0094 _H	32	—
DFE0	Output Data Register (Channel 6)	DFE0DOCH6	<DFE0_base> + 0098 _H	32	—
DFE0	Output Data Register (Channel 7)	DFE0DOCH7	<DFE0_base> + 009C _H	32	—
DFE0	Output Data Register (Channel 8)	DFE0DOCH8	<DFE0_base> + 00A0 _H	32	—
DFE0	Output Data Register (Channel 9)	DFE0DOCH9	<DFE0_base> + 00A4 _H	32	—
DFE0	Output Data Register (Channel 10)	DFE0DOCH10	<DFE0_base> + 00A8 _H	32	—
DFE0	Output Data Register (Channel 11)	DFE0DOCH11	<DFE0_base> + 00AC _H	32	—
DFE0	Peak-Hold Result Register (Channel 0)	DFE0PHCH0	<DFE0_base> + 00C0 _H	32	—
DFE0	Peak-Hold Result Register (Channel 1)	DFE0PHCH1	<DFE0_base> + 00C4 _H	32	—
DFE0	Peak-Hold Result Register (Channel 2)	DFE0PHCH2	<DFE0_base> + 00C8 _H	32	—
DFE0	Peak-Hold Result Register (Channel 3)	DFE0PHCH3	<DFE0_base> + 00CC _H	32	—
DFE0	Peak-Hold Result Register (Channel 4)	DFE0PHCH4	<DFE0_base> + 00D0 _H	32	—
DFE0	Peak-Hold Result Register (Channel 5)	DFE0PHCH5	<DFE0_base> + 00D4 _H	32	—
DFE0	Peak-Hold Result Register (Channel 6)	DFE0PHCH6	<DFE0_base> + 00D8 _H	32	—

Table 37.11 List of Registers for DFE Macro (2/8)

Module Name	Register Name	Symbol	Address	Access Size	Access Protect
DFE0	Peak-Hold Result Register (Channel 7)	DFE0PHCH7	<DFE0_base> + 00DC _H	32	—
DFE0	Peak-Hold Result Register (Channel 8)	DFE0PHCH8	<DFE0_base> + 00E0 _H	32	—
DFE0	Peak-Hold Result Register (Channel 9)	DFE0PHCH9	<DFE0_base> + 00E4 _H	32	—
DFE0	Peak-Hold Result Register (Channel 10)	DFE0PHCH10	<DFE0_base> + 00E8 _H	32	—
DFE0	Peak-Hold Result Register (Channel 11)	DFE0PHCH11	<DFE0_base> + 00EC _H	32	—
DFE0	Peak-Hold Index Register (Channel 0)	DFE0PHINDCH0	<DFE0_base> + 0100 _H	32	—
DFE0	Peak-Hold Index Register (Channel 1)	DFE0PHINDCH1	<DFE0_base> + 0104 _H	32	—
DFE0	Peak-Hold Index Register (Channel 2)	DFE0PHINDCH2	<DFE0_base> + 0108 _H	32	—
DFE0	Peak-Hold Index Register (Channel 3)	DFE0PHINDCH3	<DFE0_base> + 010C _H	32	—
DFE0	Peak-Hold Index Register (Channel 4)	DFE0PHINDCH4	<DFE0_base> + 0110 _H	32	—
DFE0	Peak-Hold Index Register (Channel 5)	DFE0PHINDCH5	<DFE0_base> + 0114 _H	32	—
DFE0	Peak-Hold Index Register (Channel 6)	DFE0PHINDCH6	<DFE0_base> + 0118 _H	32	—
DFE0	Peak-Hold Index Register (Channel 7)	DFE0PHINDCH7	<DFE0_base> + 011C _H	32	—
DFE0	Peak-Hold Index Register (Channel 8)	DFE0PHINDCH8	<DFE0_base> + 0120 _H	32	—
DFE0	Peak-Hold Index Register (Channel 9)	DFE0PHINDCH9	<DFE0_base> + 0124 _H	32	—
DFE0	Peak-Hold Index Register (Channel 10)	DFE0PHINDCH10	<DFE0_base> + 0128 _H	32	—
DFE0	Peak-Hold Index Register (Channel 11)	DFE0PHINDCH11	<DFE0_base> + 012C _H	32	—
DFE0	Status Register (Channel 0)	DFE0STCH0	<DFE0_base> + 0140 _H	8, 16, 32	—
DFE0	Status Register (Channel 1)	DFE0STCH1	<DFE0_base> + 0144 _H	8, 16, 32	—
DFE0	Status Register (Channel 2)	DFE0STCH2	<DFE0_base> + 0148 _H	8, 16, 32	—
DFE0	Status Register (Channel 3)	DFE0STCH3	<DFE0_base> + 014C _H	8, 16, 32	—
DFE0	Status Register (Channel 4)	DFE0STCH4	<DFE0_base> + 0150 _H	8, 16, 32	—
DFE0	Status Register (Channel 5)	DFE0STCH5	<DFE0_base> + 0154 _H	8, 16, 32	—
DFE0	Status Register (Channel 6)	DFE0STCH6	<DFE0_base> + 0158 _H	8, 16, 32	—
DFE0	Status Register (Channel 7)	DFE0STCH7	<DFE0_base> + 015C _H	8, 16, 32	—
DFE0	Status Register (Channel 8)	DFE0STCH8	<DFE0_base> + 0160 _H	8, 16, 32	—
DFE0	Status Register (Channel 9)	DFE0STCH9	<DFE0_base> + 0164 _H	8, 16, 32	—
DFE0	Status Register (Channel 10)	DFE0STCH10	<DFE0_base> + 0168 _H	8, 16, 32	—
DFE0	Status Register (Channel 11)	DFE0STCH11	<DFE0_base> + 016C _H	8, 16, 32	—
DFE0	Clear Status Register (Channel 0)	DFE0CLRSTCH0	<DFE0_base> + 0180 _H	8, 16	—
DFE0	Clear Status Register (Channel 1)	DFE0CLRSTCH1	<DFE0_base> + 0184 _H	8, 16	—
DFE0	Clear Status Register (Channel 2)	DFE0CLRSTCH2	<DFE0_base> + 0188 _H	8, 16	—
DFE0	Clear Status Register (Channel 3)	DFE0CLRSTCH3	<DFE0_base> + 018C _H	8, 16	—
DFE0	Clear Status Register (Channel 4)	DFE0CLRSTCH4	<DFE0_base> + 0190 _H	8, 16	—
DFE0	Clear Status Register (Channel 5)	DFE0CLRSTCH5	<DFE0_base> + 0194 _H	8, 16	—
DFE0	Clear Status Register (Channel 6)	DFE0CLRSTCH6	<DFE0_base> + 0198 _H	8, 16	—
DFE0	Clear Status Register (Channel 7)	DFE0CLRSTCH7	<DFE0_base> + 019C _H	8, 16	—
DFE0	Clear Status Register (Channel 8)	DFE0CLRSTCH8	<DFE0_base> + 01A0 _H	8, 16	—
DFE0	Clear Status Register (Channel 9)	DFE0CLRSTCH9	<DFE0_base> + 01A4 _H	8, 16	—
DFE0	Clear Status Register (Channel 10)	DFE0CLRSTCH10	<DFE0_base> + 01A8 _H	8, 16	—
DFE0	Clear Status Register (Channel 11)	DFE0CLRSTCH11	<DFE0_base> + 01AC _H	8, 16	—
DFE0	Error Mask Register (Channel 0)	DFE0ERMCH0	<DFE0_base> + 01C0 _H	8	—
DFE0	Error Mask Register (Channel 1)	DFE0ERMCH1	<DFE0_base> + 01C4 _H	8	—
DFE0	Error Mask Register (Channel 2)	DFE0ERMCH2	<DFE0_base> + 01C8 _H	8	—
DFE0	Error Mask Register (Channel 3)	DFE0ERMCH3	<DFE0_base> + 01CC _H	8	—
DFE0	Error Mask Register (Channel 4)	DFE0ERMCH4	<DFE0_base> + 01D0 _H	8	—
DFE0	Error Mask Register (Channel 5)	DFE0ERMCH5	<DFE0_base> + 01D4 _H	8	—
DFE0	Error Mask Register (Channel 6)	DFE0ERMCH6	<DFE0_base> + 01D8 _H	8	—
DFE0	Error Mask Register (Channel 7)	DFE0ERMCH7	<DFE0_base> + 01DC _H	8	—
DFE0	Error Mask Register (Channel 8)	DFE0ERMCH8	<DFE0_base> + 01E0 _H	8	—
DFE0	Error Mask Register (Channel 9)	DFE0ERMCH9	<DFE0_base> + 01E4 _H	8	—
DFE0	Error Mask Register (Channel 10)	DFE0ERMCH10	<DFE0_base> + 01E8 _H	8	—

Table 37.11 List of Registers for DFE Macro (3/8)

Module Name	Register Name	Symbol	Address	Access Size	Access Protect
DFE0	Error Mask Register (Channel 11)	DFE0ERMCH11	<DFE0_base> + 01EC _H	8	—
DFE0	Trigger Setting Register (Channel 0)	DFE0TRGCH0	<DFE0_base> + 0200 _H	16, 32	—
DFE0	Trigger Setting Register (Channel 1)	DFE0TRGCH1	<DFE0_base> + 0204 _H	16, 32	—
DFE0	Trigger Setting Register (Channel 2)	DFE0TRGCH2	<DFE0_base> + 0208 _H	16, 32	—
DFE0	Trigger Setting Register (Channel 3)	DFE0TRGCH3	<DFE0_base> + 020C _H	16, 32	—
DFE0	Trigger Setting Register (Channel 4)	DFE0TRGCH4	<DFE0_base> + 0210 _H	16, 32	—
DFE0	Trigger Setting Register (Channel 5)	DFE0TRGCH5	<DFE0_base> + 0214 _H	16, 32	—
DFE0	Trigger Setting Register (Channel 6)	DFE0TRGCH6	<DFE0_base> + 0218 _H	16, 32	—
DFE0	Trigger Setting Register (Channel 7)	DFE0TRGCH7	<DFE0_base> + 021C _H	16, 32	—
DFE0	Trigger Setting Register (Channel 8)	DFE0TRGCH8	<DFE0_base> + 0220 _H	16, 32	—
DFE0	Trigger Setting Register (Channel 9)	DFE0TRGCH9	<DFE0_base> + 0224 _H	16, 32	—
DFE0	Trigger Setting Register (Channel 10)	DFE0TRGCH10	<DFE0_base> + 0228 _H	16, 32	—
DFE0	Trigger Setting Register (Channel 11)	DFE0TRGCH11	<DFE0_base> + 022C _H	16, 32	—
DFE0	Trigger History Register (Channel 0)	DFE0TRHCH0	<DFE0_base> + 0240 _H	8	—
DFE0	Trigger History Register (Channel 1)	DFE0TRHCH1	<DFE0_base> + 0244 _H	8	—
DFE0	Trigger History Register (Channel 2)	DFE0TRHCH2	<DFE0_base> + 0248 _H	8	—
DFE0	Trigger History Register (Channel 3)	DFE0TRHCH3	<DFE0_base> + 024C _H	8	—
DFE0	Trigger History Register (Channel 4)	DFE0TRHCH4	<DFE0_base> + 0250 _H	8	—
DFE0	Trigger History Register (Channel 5)	DFE0TRHCH5	<DFE0_base> + 0254 _H	8	—
DFE0	Trigger History Register (Channel 6)	DFE0TRHCH6	<DFE0_base> + 0258 _H	8	—
DFE0	Trigger History Register (Channel 7)	DFE0TRHCH7	<DFE0_base> + 025C _H	8	—
DFE0	Trigger History Register (Channel 8)	DFE0TRHCH8	<DFE0_base> + 0260 _H	8	—
DFE0	Trigger History Register (Channel 9)	DFE0TRHCH9	<DFE0_base> + 0264 _H	8	—
DFE0	Trigger History Register (Channel 10)	DFE0TRHCH10	<DFE0_base> + 0268 _H	8	—
DFE0	Trigger History Register (Channel 11)	DFE0TRHCH11	<DFE0_base> + 026C _H	8	—
DFE0	Comparison Value Setting Register A	DFE0CPA	<DFE0_base> + 0280 _H	32	—
DFE0	Comparison Value Setting Register B	DFE0CPB	<DFE0_base> + 0284 _H	32	—
DFE0	Comparison Value Setting Register C	DFE0CPC	<DFE0_base> + 0288 _H	32	—
DFE0	Comparison Value Setting Register D	DFE0CPD	<DFE0_base> + 028C _H	32	—
DFE0	Peak-Hold Initial Value Setting Register A	DFE0PHIA	<DFE0_base> + 0290 _H	32	—
DFE0	Peak-Hold Initial Value Setting Register B	DFE0PHIB	<DFE0_base> + 0294 _H	32	—
DFE0	Peak-Hold Initial Value Setting Register C	DFE0PHIC	<DFE0_base> + 0298 _H	32	—
DFE0	Peak-Hold Initial Value Setting Register D	DFE0PHID	<DFE0_base> + 029C _H	32	—
DFE0	Peak-Hold Mask-Start/End Timer Trigger Select Register 0	DFE0PMITRG0	<DFE0_base> + 02A8 _H	8,16,32	—
DFE0	Peak-Hold Mask-Start/End Timer Trigger Select Register 1	DFE0PMITRG1	<DFE0_base> + 02AC _H	8,16,32	—
DFE0	Accumulation/Decimation Count Setting Register A	DFE0ACA	<DFE0_base> + 02B0 _H	16	—
DFE0	Accumulation/Decimation Count Setting Register B	DFE0ACB	<DFE0_base> + 02B4 _H	16	—
DFE0	Accumulation/Decimation Count Setting Register C	DFE0ACC	<DFE0_base> + 02B8 _H	16	—
DFE0	Accumulation/Decimation Count Setting Register D	DFE0ACD	<DFE0_base> + 02BC _H	16	—
DFE0	Software Input Data Register	DFE0DI	<DFE0_base> + 02C0 _H	32	—
DFE0	Software Trigger Register	DFE0TRG	<DFE0_base> + 02C4 _H	8	—
DFE0	DFE Error Status Register	DFE0EST	<DFE0_base> + 02C8 _H	32	—
DFE0	DFE Status Register	DFE0ST	<DFE0_base> + 02CC _H	8, 16, 32	—
DFE0	Peak-Hold Initialization/End Timer Trigger Select Register 1	DFE0PITRG1	<DFE0_base> + 02D4 _H	8, 16, 32	—
DFE0	Accumulation/Decimation Initialization/Prohibition Timer Trigger Select Register 1	DFE0MITRG1	<DFE0_base> + 02D8 _H	8, 16, 32	—
DFE0	Filter Initialization Timer Trigger Select Register 1	DFE0FITRG1	<DFE0_base> + 02DC _H	8, 16, 32	—
DFE0	Peak-Hold Initialization/End Timer Trigger Select Register	DFE0PITRG	<DFE0_base> + 0300 _H	8, 16, 32	—
DFE0	Accumulation/Decimation Initialization/Prohibition Timer Trigger Select Register	DFE0MITRG	<DFE0_base> + 0304 _H	8, 16, 32	—
DFE0	Filter Initialization Timer Trigger Select Register	DFE0FITRG	<DFE0_base> + 0308 _H	8, 16, 32	—
DFE0	Peak-Hold Update Notification Setting Register 0	DFE0PHUPDC0	<DFE0_base> + 0310 _H	8	—
DFE0	Peak-Hold Update Notification Setting Register 1	DFE0PHUPDC1	<DFE0_base> + 0314 _H	8	—

Table 37.11 List of Registers for DFE Macro (4/8)

Module Name	Register Name	Symbol	Address	Access Size	Access Protect
DFE0	Peak-Hold Update Notification Setting Register 2	DFE0PHUPDC2	<DFE0_base> + 0318 _H	8	—
DFE0	Peak-Hold Update Notification Setting Register 3	DFE0PHUPDC3	<DFE0_base> + 031C _H	8	—
DFE0	Intermediate Value Output Register L (Channel 0)	DFE0HOLCH0	<DFE0_base> + 0320 _H	32	—
DFE0	Intermediate Value Output Register H (Channel 0)	DFE0HOHCH0	<DFE0_base> + 0324 _H	32	—
DFE0	Intermediate Value Output Register L (Channel 2)	DFE0HOLCH2	<DFE0_base> + 0328 _H	32	—
DFE0	Intermediate Value Output Register H (Channel 2)	DFE0HOHCH2	<DFE0_base> + 032C _H	32	—
DFE0	Intermediate Value Output Register L (Channel 4)	DFE0HOLCH4	<DFE0_base> + 0330 _H	32	—
DFE0	Intermediate Value Output Register H (Channel 4)	DFE0HOHCH4	<DFE0_base> + 0334 _H	32	—
DFE0	Intermediate Value Output Register L (Channel 6)	DFE0HOLCH6	<DFE0_base> + 0338 _H	32	—
DFE0	Intermediate Value Output Register H (Channel 6)	DFE0HOHCH6	<DFE0_base> + 033C _H	32	—
DFE0	Intermediate Value Output Register L (Channel 8)	DFE0HOLCH8	<DFE0_base> + 0340 _H	32	—
DFE0	Intermediate Value Output Register H (Channel 8)	DFE0HOHCH8	<DFE0_base> + 0344 _H	32	—
DFE0	Intermediate Value Output Register L (Channel 10)	DFE0HOLCH10	<DFE0_base> + 0348 _H	32	—
DFE0	Intermediate Value Output Register H (Channel 10)	DFE0HOHCH10	<DFE0_base> + 034C _H	32	—
DFE0	Intermediate Value Output Mirror Register L (Channel 0)	DFE0HOLMCH0	<DFE0_base> + 0360 _H	32	—
DFE0	Intermediate Value Output Mirror Register L (Channel 2)	DFE0HOLMCH2	<DFE0_base> + 0364 _H	32	—
DFE0	Intermediate Value Output Mirror Register L (Channel 4)	DFE0HOLMCH4	<DFE0_base> + 0368 _H	32	—
DFE0	Intermediate Value Output Mirror Register L (Channel 6)	DFE0HOLMCH6	<DFE0_base> + 036C _H	32	—
DFE0	Intermediate Value Output Mirror Register L (Channel 8)	DFE0HOLMCH8	<DFE0_base> + 0370 _H	32	—
DFE0	Intermediate Value Output Mirror Register L (Channel 10)	DFE0HOLMCH10	<DFE0_base> + 0374 _H	32	—
DFE0	IIR Filter Gain Setting Register (Channel 0)	DFE0GAINCH0	<DFE0_base> + 0380 _H	8,16,32	—
DFE0	IIR Filter Gain Setting Register (Channel 1)	DFE0GAINCH1	<DFE0_base> + 0384 _H	8,16,32	—
DFE0	IIR Filter Gain Setting Register (Channel 2)	DFE0GAINCH2	<DFE0_base> + 0388 _H	8,16,32	—
DFE0	IIR Filter Gain Setting Register (Channel 3)	DFE0GAINCH3	<DFE0_base> + 038C _H	8,16,32	—
DFE0	IIR Filter Gain Setting Register (Channel 4)	DFE0GAINCH4	<DFE0_base> + 0390 _H	8,16,32	—
DFE0	IIR Filter Gain Setting Register (Channel 5)	DFE0GAINCH5	<DFE0_base> + 0394 _H	8,16,32	—
DFE0	IIR Filter Gain Setting Register (Channel 6)	DFE0GAINCH6	<DFE0_base> + 0398 _H	8,16,32	—
DFE0	IIR Filter Gain Setting Register (Channel 7)	DFE0GAINCH7	<DFE0_base> + 039C _H	8,16,32	—
DFE0	IIR Filter Gain Setting Register (Channel 8)	DFE0GAINCH8	<DFE0_base> + 03A0 _H	8,16,32	—
DFE0	IIR Filter Gain Setting Register (Channel 9)	DFE0GAINCH9	<DFE0_base> + 03A4 _H	8,16,32	—
DFE0	IIR Filter Gain Setting Register (Channel 10)	DFE0GAINCH10	<DFE0_base> + 03A8 _H	8,16,32	—
DFE0	IIR Filter Gain Setting Register (Channel 11)	DFE0GAINCH11	<DFE0_base> + 03AC _H	8,16,32	—
DFE0	Timer Trigger Setting Register (Channel 0)	DFE0TMTRGCH0	<DFE0_base> + 03C0 _H	8,16,32	—
DFE0	Timer Trigger Setting Register (Channel 1)	DFE0TMTRGCH1	<DFE0_base> + 03C4 _H	8,16,32	—
DFE0	Timer Trigger Setting Register (Channel 2)	DFE0TMTRGCH2	<DFE0_base> + 03C8 _H	8,16,32	—
DFE0	Timer Trigger Setting Register (Channel 3)	DFE0TMTRGCH3	<DFE0_base> + 03CC _H	8,16,32	—
DFE0	Timer Trigger Setting Register (Channel 4)	DFE0TMTRGCH4	<DFE0_base> + 03D0 _H	8,16,32	—
DFE0	Timer Trigger Setting Register (Channel 5)	DFE0TMTRGCH5	<DFE0_base> + 03D4 _H	8,16,32	—
DFE0	Timer Trigger Setting Register (Channel 6)	DFE0TMTRGCH6	<DFE0_base> + 03D8 _H	8,16,32	—
DFE0	Timer Trigger Setting Register (Channel 7)	DFE0TMTRGCH7	<DFE0_base> + 03DC _H	8,16,32	—
DFE0	Timer Trigger Setting Register (Channel 8)	DFE0TMTRGCH8	<DFE0_base> + 03E0 _H	8,16,32	—
DFE0	Timer Trigger Setting Register (Channel 9)	DFE0TMTRGCH9	<DFE0_base> + 03E4 _H	8,16,32	—
DFE0	Timer Trigger Setting Register (Channel 10)	DFE0TMTRGCH10	<DFE0_base> + 03E8 _H	8,16,32	—
DFE0	Timer Trigger Setting Register (Channel 11)	DFE0TMTRGCH11	<DFE0_base> + 03EC _H	8,16,32	—
DFE0	Comparison Offset Value Setting Register 0	DFE0CPOFST0	<DFE0_base> + 0440 _H	8,16,32	—
DFE0	Comparison Offset Value Setting Register 1	DFE0CPOFST1	<DFE0_base> + 0444 _H	8,16,32	—
DFE0	Comparison Offset Value Setting Register 2	DFE0CPOFST2	<DFE0_base> + 0448 _H	8,16,32	—
DFE0	Peak-Hold 23 Common Control Register 0	DFE0PH23CCTL0	<DFE0_base> + 0700 _H	8,16,32	—
DFE0	Peak-Hold 2 Control Register 0	DFE0PH2CTL0	<DFE0_base> + 0704 _H	8,16,32	—
DFE0	Peak-Hold 3 Control Register 0	DFE0PH3CTL0	<DFE0_base> + 0708 _H	8,16,32	—
DFE0	Peak-Hold 2 Result Register 0	DFE0PH20	<DFE0_base> + 0780 _H	32	—
DFE0	Peak-Hold 3 Result Register 0	DFE0PH30	<DFE0_base> + 0784 _H	32	—

Table 37.11 List of Registers for DFE Macro (5/8)

Module Name	Register Name	Symbol	Address	Access Size	Access Protect
DFE0	Peak-Hold 2 Index Register 0	DFE0PH2IND0	<DFE0_base> + 07E0 _H	32	—
DFE0	Peak-Hold 3 Index Register 0	DFE0PH3IND0	<DFE0_base> + 07E4 _H	32	—
DFE0	Subtraction Control Register (SCH0)	DFE0SUBCTLCH0	<DFE0_base> + 0840 _H	8,16,32	—
DFE0	Subtraction Control Register (SCH1)	DFE0SUBCTLCH1	<DFE0_base> + 0844 _H	8,16,32	—
DFE0	Subtraction Control Register (SCH2)	DFE0SUBCTLCH2	<DFE0_base> + 0848 _H	8,16,32	—
DFE0	Subtraction Result Register (SCH0)	DFE0SUBDOCH0	<DFE0_base> + 0860 _H	32	—
DFE0	Subtraction Result Register (SCH1)	DFE0SUBDOCH1	<DFE0_base> + 0864 _H	32	—
DFE0	Subtraction Result Register (SCH2)	DFE0SUBDOCH2	<DFE0_base> + 0868 _H	32	—
DFE0	Subtraction Status Register (SCH0)	DFE0SUBSTCH0	<DFE0_base> + 0880 _H	32	—
DFE0	Subtraction Status Register (SCH1)	DFE0SUBSTCH1	<DFE0_base> + 0884 _H	32	—
DFE0	Subtraction Status Register (SCH2)	DFE0SUBSTCH2	<DFE0_base> + 0888 _H	32	—
DFE0	Subtraction Clear Status Register (SCH0)	DFE0SUBCLRSTCH0	<DFE0_base> + 08A0 _H	8,16,32	—
DFE0	Subtraction Clear Status Register (SCH1)	DFE0SUBCLRSTCH1	<DFE0_base> + 08A4 _H	8,16,32	—
DFE0	Subtraction Clear Status Register (SCH2)	DFE0SUBCLRSTCH2	<DFE0_base> + 08A8 _H	8,16,32	—
DFE0	Subtraction Error Mask Register (SCH0)	DFE0SUBERMCH0	<DFE0_base> + 08C0 _H	8,16,32	—
DFE0	Subtraction Error Mask Register (SCH1)	DFE0SUBERMCH1	<DFE0_base> + 08C4 _H	8,16,32	—
DFE0	Subtraction Error Mask Register (SCH2)	DFE0SUBERMCH2	<DFE0_base> + 08C8 _H	8,16,32	—
DFE0	Subtraction Trigger Setting Register (SCH0)	DFE0SUBTRGCH0	<DFE0_base> + 08E0 _H	8,16,32	—
DFE0	Subtraction Trigger Setting Register (SCH1)	DFE0SUBTRGCH1	<DFE0_base> + 08E4 _H	8,16,32	—
DFE0	Subtraction Trigger Setting Register (SCH2)	DFE0SUBTRGCH2	<DFE0_base> + 08E8 _H	8,16,32	—
DFE0	Subtraction Trigger History Register (SCH0)	DFE0SUBTRHCH0	<DFE0_base> + 0900 _H	32	—
DFE0	Subtraction Trigger History Register (SCH1)	DFE0SUBTRHCH1	<DFE0_base> + 0904 _H	32	—
DFE0	Subtraction Trigger History Register (SCH2)	DFE0SUBTRHCH2	<DFE0_base> + 0908 _H	32	—
DFE0	Subtraction Start/End Timer Trigger Select Register 0	DFE0SUBTRG0	<DFE0_base> + 0920 _H	8,16,32	—
DFE0	Capture Control Register (CCH0)	DFE0CAPCTLCH0	<DFE0_base> + 0940 _H	32	—
DFE0	Capture Control Register (CCH1)	DFE0CAPCTLCH1	<DFE0_base> + 0944 _H	32	—
DFE0	Capture Control Register (CCH2)	DFE0CAPCTLCH2	<DFE0_base> + 0948 _H	32	—
DFE0	Capture Result Register (CCH0)	DFE0CAPDOCH0	<DFE0_base> + 0960 _H	32	—
DFE0	Capture Result Register (CCH1)	DFE0CAPDOCH1	<DFE0_base> + 0964 _H	32	—
DFE0	Capture Result Register (CCH2)	DFE0CAPDOCH2	<DFE0_base> + 0968 _H	32	—
DFE0	Capture Status Register (CCH0)	DFE0CAPSTCH0	<DFE0_base> + 0980 _H	32	—
DFE0	Capture Status Register (CCH1)	DFE0CAPSTCH1	<DFE0_base> + 0984 _H	32	—
DFE0	Capture Status Register (CCH2)	DFE0CAPSTCH2	<DFE0_base> + 0988 _H	32	—
DFE0	Capture Clear Status Register (CCH0)	DFE0CAPCLRSTCH0	<DFE0_base> + 09A0 _H	8,16,32	—
DFE0	Capture Clear Status Register (CCH1)	DFE0CAPCLRSTCH1	<DFE0_base> + 09A4 _H	8,16,32	—
DFE0	Capture Clear Status Register (CCH2)	DFE0CAPCLRSTCH2	<DFE0_base> + 09A8 _H	8,16,32	—
DFE0	Capture Error Mask Register (CCH0)	DFE0CAPERMCH0	<DFE0_base> + 09C0 _H	8,16,32	—
DFE0	Capture Error Mask Register (CCH1)	DFE0CAPERMCH1	<DFE0_base> + 09C4 _H	8,16,32	—
DFE0	Capture Error Mask Register (CCH2)	DFE0CAPERMCH2	<DFE0_base> + 09C8 _H	8,16,32	—
DFE0	Coefficient Memory Area (0.75 KB)	—	<DFE0_base> + 1000 _H to 12FF _H	32	—
DFE0	Data Memory Area (1.5 KB)	—	<DFE0_base> + 2000 _H to 25FF _H	16, 32	—
DFE1	Control Register A (Channel 0)	DFE1CTLACH0	<DFE1_base> + 0000 _H	8, 16, 32	—
DFE1	Control Register A (Channel 1)	DFE1CTLACH1	<DFE1_base> + 0004 _H	8, 16, 32	—
DFE1	Control Register A (Channel 2)	DFE1CTLACH2	<DFE1_base> + 0008 _H	8, 16, 32	—
DFE1	Control Register A (Channel 3)	DFE1CTLACH3	<DFE1_base> + 000C _H	8, 16, 32	—
DFE1	Control Register B (Channel 0)	DFE1CTLBCH0	<DFE1_base> + 0040 _H	8, 16, 32	—
DFE1	Control Register B (Channel 1)	DFE1CTLBCH1	<DFE1_base> + 0044 _H	8, 16, 32	—
DFE1	Control Register B (Channel 2)	DFE1CTLBCH2	<DFE1_base> + 0048 _H	8, 16, 32	—
DFE1	Control Register B (Channel 3)	DFE1CTLBCH3	<DFE1_base> + 004C _H	8, 16, 32	—
DFE1	Output Data Register (Channel 0)	DFE1DOCH0	<DFE1_base> + 0080 _H	32	—
DFE1	Output Data Register (Channel 1)	DFE1DOCH1	<DFE1_base> + 0084 _H	32	—

Table 37.11 List of Registers for DFE Macro (6/8)

Module Name	Register Name	Symbol	Address	Access Size	Access Protect
DFE1	Output Data Register (Channel 2)	DFE1DOCH2	<DFE1_base> + 0088 _H	32	—
DFE1	Output Data Register (Channel 3)	DFE1DOCH3	<DFE1_base> + 008C _H	32	—
DFE1	Peak-Hold Result Register (Channel 0)	DFE1PHCH0	<DFE1_base> + 00C0 _H	32	—
DFE1	Peak-Hold Result Register (Channel 1)	DFE1PHCH1	<DFE1_base> + 00C4 _H	32	—
DFE1	Peak-Hold Result Register (Channel 2)	DFE1PHCH2	<DFE1_base> + 00C8 _H	32	—
DFE1	Peak-Hold Result Register (Channel 3)	DFE1PHCH3	<DFE1_base> + 00CC _H	32	—
DFE1	Peak-Hold Index Register (Channel 0)	DFE1PHINDCH0	<DFE1_base> + 0100 _H	32	—
DFE1	Peak-Hold Index Register (Channel 1)	DFE1PHINDCH1	<DFE1_base> + 0104 _H	32	—
DFE1	Peak-Hold Index Register (Channel 2)	DFE1PHINDCH2	<DFE1_base> + 0108 _H	32	—
DFE1	Peak-Hold Index Register (Channel 3)	DFE1PHINDCH3	<DFE1_base> + 010C _H	32	—
DFE1	Status Register (Channel 0)	DFE1STCH0	<DFE1_base> + 0140 _H	8, 16, 32	—
DFE1	Status Register (Channel 1)	DFE1STCH1	<DFE1_base> + 0144 _H	8, 16, 32	—
DFE1	Status Register (Channel 2)	DFE1STCH2	<DFE1_base> + 0148 _H	8, 16, 32	—
DFE1	Status Register (Channel 3)	DFE1STCH3	<DFE1_base> + 014C _H	8, 16, 32	—
DFE1	Clear Status Register (Channel 0)	DFE1CLRSTC0	<DFE1_base> + 0180 _H	8, 16	—
DFE1	Clear Status Register (Channel 1)	DFE1CLRSTC1	<DFE1_base> + 0184 _H	8, 16	—
DFE1	Clear Status Register (Channel 2)	DFE1CLRSTC2	<DFE1_base> + 0188 _H	8, 16	—
DFE1	Clear Status Register (Channel 3)	DFE1CLRSTC3	<DFE1_base> + 018C _H	8, 16	—
DFE1	Error Mask Register (Channel 0)	DFE1ERMCH0	<DFE1_base> + 01C0 _H	8	—
DFE1	Error Mask Register (Channel 1)	DFE1ERMCH1	<DFE1_base> + 01C4 _H	8	—
DFE1	Error Mask Register (Channel 2)	DFE1ERMCH2	<DFE1_base> + 01C8 _H	8	—
DFE1	Error Mask Register (Channel 3)	DFE1ERMCH3	<DFE1_base> + 01CC _H	8	—
DFE1	Trigger Setting Register (Channel 0)	DFE1TRGCH0	<DFE1_base> + 0200 _H	16, 32	—
DFE1	Trigger Setting Register (Channel 1)	DFE1TRGCH1	<DFE1_base> + 0204 _H	16, 32	—
DFE1	Trigger Setting Register (Channel 2)	DFE1TRGCH2	<DFE1_base> + 0208 _H	16, 32	—
DFE1	Trigger Setting Register (Channel 3)	DFE1TRGCH3	<DFE1_base> + 020C _H	16, 32	—
DFE1	Trigger History Register (Channel 0)	DFE1TRHCH0	<DFE1_base> + 0240 _H	8	—
DFE1	Trigger History Register (Channel 1)	DFE1TRHCH1	<DFE1_base> + 0244 _H	8	—
DFE1	Trigger History Register (Channel 2)	DFE1TRHCH2	<DFE1_base> + 0248 _H	8	—
DFE1	Trigger History Register (Channel 3)	DFE1TRHCH3	<DFE1_base> + 024C _H	8	—
DFE1	Comparison Value Setting Register A	DFE1CPA	<DFE1_base> + 0280 _H	32	—
DFE1	Comparison Value Setting Register B	DFE1CPB	<DFE1_base> + 0284 _H	32	—
DFE1	Comparison Value Setting Register C	DFE1CPC	<DFE1_base> + 0288 _H	32	—
DFE1	Comparison Value Setting Register D	DFE1CPD	<DFE1_base> + 028C _H	32	—
DFE1	Peak-Hold Initial Value Setting Register A	DFE1PHIA	<DFE1_base> + 0290 _H	32	—
DFE1	Peak-Hold Initial Value Setting Register B	DFE1PHIB	<DFE1_base> + 0294 _H	32	—
DFE1	Peak-Hold Initial Value Setting Register C	DFE1PHIC	<DFE1_base> + 0298 _H	32	—
DFE1	Peak-Hold Initial Value Setting Register D	DFE1PHID	<DFE1_base> + 029C _H	32	—
DFE1	Peak-Hold Mask-Start/End Timer Trigger Select Register 0	DFE1PMITRG0	<DFE1_base> + 02A8 _H	8,16,32	—
DFE1	Peak-Hold Mask-Start/End Timer Trigger Select Register 1	DFE1PMITRG1	<DFE1_base> + 02AC _H	8,16,32	—
DFE1	Accumulation/Decimation Count Setting Register A	DFE1ACA	<DFE1_base> + 02B0 _H	16	—
DFE1	Accumulation/Decimation Count Setting Register B	DFE1ACB	<DFE1_base> + 02B4 _H	16	—
DFE1	Accumulation/Decimation Count Setting Register C	DFE1ACC	<DFE1_base> + 02B8 _H	16	—
DFE1	Accumulation/Decimation Count Setting Register D	DFE1ACD	<DFE1_base> + 02BC _H	16	—
DFE1	Software Input Data Register	DFE1DI	<DFE1_base> + 02C0 _H	32	—
DFE1	Software Trigger Register	DFE1TRG	<DFE1_base> + 02C4 _H	8	—
DFE1	DFE Error Status Register	DFE1EST	<DFE1_base> + 02C8 _H	32	—
DFE1	DFE Status Register	DFE1ST	<DFE1_base> + 02CC _H	8, 16, 32	—
DFE1	Peak-Hold Initialization/End Timer Trigger Select Register 1	DFE1PITRG1	<DFE1_base> + 02D4 _H	8, 16, 32	—
DFE1	Accumulation/Decimation Initialization/Prohibition Timer Trigger Select Register 1	DFE1MITRG1	<DFE1_base> + 02D8 _H	8, 16, 32	—
DFE1	Filter Initialization Timer Trigger Select Register 1	DFE1FITRG1	<DFE1_base> + 02DC _H	8, 16, 32	—

Table 37.11 List of Registers for DFE Macro (7/8)

Module Name	Register Name	Symbol	Address	Access Size	Access Protect
DFE1	Peak-Hold Initialization/End Timer Trigger Select Register	DFE1PITRG	<DFE1_base> + 0300 _H	8, 16, 32	—
DFE1	Accumulation/Decimation Initialization/Prohibition Timer Trigger Select Register	DFE1MITRG	<DFE1_base> + 0304 _H	8, 16, 32	—
DFE1	Filter Initialization Timer Trigger Select Register	DFE1FITRG	<DFE1_base> + 0308 _H	8, 16, 32	—
DFE1	Peak-Hold Update Notification Setting Register 0	DFE1PHUPDC0	<DFE1_base> + 0310 _H	8	—
DFE1	Peak-Hold Update Notification Setting Register 1	DFE1PHUPDC1	<DFE1_base> + 0314 _H	8	—
DFE1	Peak-Hold Update Notification Setting Register 2	DFE1PHUPDC2	<DFE1_base> + 0318 _H	8	—
DFE1	Peak-Hold Update Notification Setting Register 3	DFE1PHUPDC3	<DFE1_base> + 031C _H	8	—
DFE1	Intermediate Value Output Register L (Channel 0)	DFE1HOLCH0	<DFE1_base> + 0320 _H	32	—
DFE1	Intermediate Value Output Register H (Channel 0)	DFE1HOHCH0	<DFE1_base> + 0324 _H	32	—
DFE1	Intermediate Value Output Register L (Channel 2)	DFE1HOLCH2	<DFE1_base> + 0328 _H	32	—
DFE1	Intermediate Value Output Register H (Channel 2)	DFE1HOHCH2	<DFE1_base> + 032C _H	32	—
DFE1	Intermediate Value Output Mirror Register L (Channel 0)	DFE1HOLMCH0	<DFE1_base> + 0360 _H	32	—
DFE1	Intermediate Value Output Mirror Register L (Channel 2)	DFE1HOLMCH2	<DFE1_base> + 0364 _H	32	—
DFE1	IIR Filter Gain Setting Register (Channel 0)	DFE1GAINCH0	<DFE1_base> + 0380 _H	8,16,32	—
DFE1	IIR Filter Gain Setting Register (Channel 1)	DFE1GAINCH1	<DFE1_base> + 0384 _H	8,16,32	—
DFE1	IIR Filter Gain Setting Register (Channel 2)	DFE1GAINCH2	<DFE1_base> + 0388 _H	8,16,32	—
DFE1	IIR Filter Gain Setting Register (Channel 3)	DFE1GAINCH3	<DFE1_base> + 038C _H	8,16,32	—
DFE1	Timer Trigger Setting Register (Channel 0)	DFE1TMTRGCH0	<DFE1_base> + 03C0 _H	8,16,32	—
DFE1	Timer Trigger Setting Register (Channel 1)	DFE1TMTRGCH1	<DFE1_base> + 03C4 _H	8,16,32	—
DFE1	Timer Trigger Setting Register (Channel 2)	DFE1TMTRGCH2	<DFE1_base> + 03C8 _H	8,16,32	—
DFE1	Timer Trigger Setting Register (Channel 3)	DFE1TMTRGCH3	<DFE1_base> + 03CC _H	8,16,32	—
DFE1	Comparison Offset Value Setting Register 0	DFE1CPOFST0	<DFE1_base> + 0440 _H	8,16,32	—
DFE1	Comparison Offset Value Setting Register 1	DFE1CPOFST1	<DFE1_base> + 0444 _H	8,16,32	—
DFE1	Comparison Offset Value Setting Register 2	DFE1CPOFST2	<DFE1_base> + 0448 _H	8,16,32	—
DFE1	Peak-Hold 23 Common Control Register 0	DFE1PH23CCTL0	<DFE1_base> + 0700 _H	8,16,32	—
DFE1	Peak-Hold 2 Control Register 0	DFE1PH2CTL0	<DFE1_base> + 0704 _H	8,16,32	—
DFE1	Peak-Hold 3 Control Register 0	DFE1PH3CTL0	<DFE1_base> + 0708 _H	8,16,32	—
DFE1	Peak-Hold 2 Result Register 0	DFE1PH20	<DFE1_base> + 0780 _H	32	—
DFE1	Peak-Hold 3 Result Register 0	DFE1PH30	<DFE1_base> + 0784 _H	32	—
DFE1	Peak-Hold 2 Index Register 0	DFE1PH2IND0	<DFE1_base> + 07E0 _H	32	—
DFE1	Peak-Hold 3 Index Register 0	DFE1PH3IND0	<DFE1_base> + 07E4 _H	32	—
DFE1	Subtraction Control Register (SCH0)	DFE1SUBCTLCH0	<DFE1_base> + 0840 _H	8,16,32	—
DFE1	Subtraction Control Register (SCH1)	DFE1SUBCTLCH1	<DFE1_base> + 0844 _H	8,16,32	—
DFE1	Subtraction Control Register (SCH2)	DFE1SUBCTLCH2	<DFE1_base> + 0848 _H	8,16,32	—
DFE1	Subtraction Result Register (SCH0)	DFE1SUBDOCH0	<DFE1_base> + 0860 _H	32	—
DFE1	Subtraction Result Register (SCH1)	DFE1SUBDOCH1	<DFE1_base> + 0864 _H	32	—
DFE1	Subtraction Result Register (SCH2)	DFE1SUBDOCH2	<DFE1_base> + 0868 _H	32	—
DFE1	Subtraction Status Register (SCH0)	DFE1SUBSTCH0	<DFE1_base> + 0880 _H	32	—
DFE1	Subtraction Status Register (SCH1)	DFE1SUBSTCH1	<DFE1_base> + 0884 _H	32	—
DFE1	Subtraction Status Register (SCH2)	DFE1SUBSTCH2	<DFE1_base> + 0888 _H	32	—
DFE1	Subtraction Clear Status Register (SCH0)	DFE1SUBCLRSTCH0	<DFE1_base> + 08A0 _H	8,16,32	—
DFE1	Subtraction Clear Status Register (SCH1)	DFE1SUBCLRSTCH1	<DFE1_base> + 08A4 _H	8,16,32	—
DFE1	Subtraction Clear Status Register (SCH2)	DFE1SUBCLRSTCH2	<DFE1_base> + 08A8 _H	8,16,32	—
DFE1	Subtraction Error Mask Register (SCH0)	DFE1SUBERMCH0	<DFE1_base> + 08C0 _H	8,16,32	—
DFE1	Subtraction Error Mask Register (SCH1)	DFE1SUBERMCH1	<DFE1_base> + 08C4 _H	8,16,32	—
DFE1	Subtraction Error Mask Register (SCH2)	DFE1SUBERMCH2	<DFE1_base> + 08C8 _H	8,16,32	—
DFE1	Subtraction Trigger Setting Register (SCH0)	DFE1SUBTRGCH0	<DFE1_base> + 08E0 _H	8,16,32	—
DFE1	Subtraction Trigger Setting Register (SCH1)	DFE1SUBTRGCH1	<DFE1_base> + 08E4 _H	8,16,32	—
DFE1	Subtraction Trigger Setting Register (SCH2)	DFE1SUBTRGCH2	<DFE1_base> + 08E8 _H	8,16,32	—
DFE1	Subtraction Trigger History Register (SCH0)	DFE1SUBTRHCH0	<DFE1_base> + 0900 _H	32	—
DFE1	Subtraction Trigger History Register (SCH1)	DFE1SUBTRHCH1	<DFE1_base> + 0904 _H	32	—

Table 37.11 List of Registers for DFE Macro (8/8)

Module Name	Register Name	Symbol	Address	Access Size	Access Protect
DFE1	Subtraction Trigger History Register (SCH2)	DFE1SUBTRHCH2	<DFE1_base> + 0908 _H	32	—
DFE1	Subtraction Start/End Timer Trigger Select Register 0	DFE1SUBTRG0	<DFE1_base> + 0920 _H	8,16,32	—
DFE1	Capture Control Register (CCH0)	DFE1CAPCTLCH0	<DFE1_base> + 0940 _H	32	—
DFE1	Capture Control Register (CCH1)	DFE1CAPCTLCH1	<DFE1_base> + 0944 _H	32	—
DFE1	Capture Control Register (CCH2)	DFE1CAPCTLCH2	<DFE1_base> + 0948 _H	32	—
DFE1	Capture Result Register (CCH0)	DFE1CAPDOCH0	<DFE1_base> + 0960 _H	32	—
DFE1	Capture Result Register (CCH1)	DFE1CAPDOCH1	<DFE1_base> + 0964 _H	32	—
DFE1	Capture Result Register (CCH2)	DFE1CAPDOCH2	<DFE1_base> + 0968 _H	32	—
DFE1	Capture Status Register (CCH0)	DFE1CAPSTCH0	<DFE1_base> + 0980 _H	32	—
DFE1	Capture Status Register (CCH1)	DFE1CAPSTCH1	<DFE1_base> + 0984 _H	32	—
DFE1	Capture Status Register (CCH2)	DFE1CAPSTCH2	<DFE1_base> + 0988 _H	32	—
DFE1	Capture Clear Status Register (CCH0)	DFE1CAPCLRSTCH0	<DFE1_base> + 09A0 _H	8,16,32	—
DFE1	Capture Clear Status Register (CCH1)	DFE1CAPCLRSTCH1	<DFE1_base> + 09A4 _H	8,16,32	—
DFE1	Capture Clear Status Register (CCH2)	DFE1CAPCLRSTCH2	<DFE1_base> + 09A8 _H	8,16,32	—
DFE1	Capture Error Mask Register (CCH0)	DFE1CAPERMCH0	<DFE1_base> + 09C0 _H	8,16,32	—
DFE1	Capture Error Mask Register (CCH1)	DFE1CAPERMCH1	<DFE1_base> + 09C4 _H	8,16,32	—
DFE1	Capture Error Mask Register (CCH2)	DFE1CAPERMCH2	<DFE1_base> + 09C8 _H	8,16,32	—
DFE1	Coefficient Memory Area (0.25 KB)	—	<DFE1_base> + 1000 _H to 10FF _H	32	—
DFE1	Data Memory Area (0.5 KB)	—	<DFE1_base> + 2000 _H to 21FF _H	16, 32	—

37.3.2 List of Registers for the FIFO Macro

Table 37.12 lists the registers and memory addresses of the FIFO macro. 8, 16, or 32-bit write access to all registers is possible. 32-bit read access to all registers is possible.

Table 37.12 List of the Control Registers for the FIFO Macro

Module Name	Register Name	Symbol	Address	Access Size	Access Protect
DFE FIFO	Buffer A Common Control Register	DFBFACCTL	< DFE_FIFO_base > + 0000 _H	8, 16, 32	—
DFE FIFO	Buffer A Control Register (FCH0)	DFBFACTLCH0	< DFE_FIFO_base > + 0010 _H	8, 16, 32	—
DFE FIFO	Buffer A Control Register (FCH1)	DFBFACTLCH1	< DFE_FIFO_base > + 0014 _H	8, 16, 32	—
DFE FIFO	Buffer A Control Register (FCH2)	DFBFACTLCH2	< DFE_FIFO_base > + 0018 _H	8, 16, 32	—
DFE FIFO	Buffer A Control Register (FCH3)	DFBFACTLCH3	< DFE_FIFO_base > + 001C _H	8, 16, 32	—
DFE FIFO	Buffer A Control Register (FCH4)	DFBFACTLCH4	< DFE_FIFO_base > + 0020 _H	8, 16, 32	—
DFE FIFO	Buffer A Control Register (FCH5)	DFBFACTLCH5	< DFE_FIFO_base > + 0024 _H	8, 16, 32	—
DFE FIFO	Buffer A Output Register (FCH0)	DFBFADOCH0	< DFE_FIFO_base > + 0050 _H	32	—
DFE FIFO	Buffer A Output Register (FCH1)	DFBFADOCH1	< DFE_FIFO_base > + 0054 _H	32	—
DFE FIFO	Buffer A Output Register (FCH2)	DFBFADOCH2	< DFE_FIFO_base > + 0058 _H	32	—
DFE FIFO	Buffer A Output Register (FCH3)	DFBFADOCH3	< DFE_FIFO_base > + 005C _H	32	—
DFE FIFO	Buffer A Output Register (FCH4)	DFBFADOCH4	< DFE_FIFO_base > + 0060 _H	32	—
DFE FIFO	Buffer A Output Register (FCH5)	DFBFADOCH5	< DFE_FIFO_base > + 0064 _H	32	—
DFE FIFO	Buffer A Clear Register	DFBFACLR	< DFE_FIFO_base > + 0090 _H	8, 16, 32	—
DFE FIFO	Buffer A Common Status Register	DFBFACST	< DFE_FIFO_base > + 00A0 _H	32	—
DFE FIFO	Buffer B Control Register	DFBFBCTL	< DFE_FIFO_base > + 00B0 _H	8, 16, 32	—
DFE FIFO	Buffer B Output Register	DFBFBDO	< DFE_FIFO_base > + 00B4 _H	32	—
DFE FIFO	Buffer B Clear Register	DFBFBCLR	< DFE_FIFO_base > + 00B8 _H	8, 16, 32	—
DFE FIFO	Buffer B Status Register	DFBFBST	< DFE_FIFO_base > + 00BC _H	32	—

37.3.3 DFEjCTLACHn — Control Register A n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1)

This register is used to execute filter processing, control interrupt requests, and set channel tag and cascade tag for each channel.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CATAG					CAEN		—	—	—	TAG				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMD				—	—	—	FMT	IEP	IEE	IEC	IEO	CNSL	CNSLE	AIME	EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: While the setting of DFEjCTLACHn.EN or DFEjSTCHn.VALID is 1, writing to CATAG, CAEN, TAG, CMD, FMT, IEP, IEE, IEC, IEO, CNSL, CNSLE and AIME is prohibited.

Table 37.13 DFEjCTLACHn Register Contents (1/3)

Bit Position	Bit Name	Function										
31	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.										
30 to 26	CATAG	Tag Value (Cascade Tag) for Reentering the Filter Result (Cascade Input) 0000 _B to 1111 _B : Output data of the accumulation circuit is reentered (cascaded) during execution (CAEN = 10 _B or 11 _B). A channel tag value (DFEjCTLACHn.TAG) is specified. When cascading is off (CAEN = 00 _B or 01 _B), all CATAG settings are ignored.										
25, 24	CAEN	Cascade Enable 00 _B : Cascade is disabled. Output data of the accumulation circuit is written to the output data register. 01 _B : Setting prohibited 10 _B : Cascade is enabled. Output data of the accumulation circuit is written to the output data register. 11 _B : Cascade is enabled. Output data of the accumulation circuit is not written to the output data register.										
23 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.										
20 to 16	TAG	Channel Tag 0000 _B to 1111 _B : The following four types of filtering process are enabled when the TAG for this channel and the tags to be filtered are set to the same value and a request is issued:										
		<table border="1"> <thead> <tr> <th>Objects of Filtering Process</th> <th>TAG Settings</th> </tr> </thead> <tbody> <tr> <td>AD input data</td> <td>Same value as the tag value [3:0] for AD*1</td> </tr> <tr> <td>Software input data</td> <td>Same value as DFEjDI.TAG [4:0]</td> </tr> <tr> <td>Cascading input on other channels</td> <td>Same value as DFEjCTLACHn.CATAG [4:0]</td> </tr> <tr> <td>Cascading input on the subtraction channel</td> <td>Same value as DFEjSUBCTLACHnCATAG [4:0]</td> </tr> </tbody> </table>	Objects of Filtering Process	TAG Settings	AD input data	Same value as the tag value [3:0] for AD*1	Software input data	Same value as DFEjDI.TAG [4:0]	Cascading input on other channels	Same value as DFEjCTLACHn.CATAG [4:0]	Cascading input on the subtraction channel	Same value as DFEjSUBCTLACHnCATAG [4:0]
Objects of Filtering Process	TAG Settings											
AD input data	Same value as the tag value [3:0] for AD*1											
Software input data	Same value as DFEjDI.TAG [4:0]											
Cascading input on other channels	Same value as DFEjCTLACHn.CATAG [4:0]											
Cascading input on the subtraction channel	Same value as DFEjSUBCTLACHnCATAG [4:0]											
		Note 1. Set the TAG [4] to 0, to correspond to the AD TAG value.										

Table 37.13 DFEjCTLACHn Register Contents (2/3)

Bit Position	Bit Name	Function
15 to 12	CMD	Filter Processing Select 0000 _B : FIR 8TAP 0001 _B : FIR 16TAP 0010 _B : FIR 24TAP 0011 _B : FIR 32TAP 0100 _B : Setting prohibited 0101 _B : Setting prohibited 0110 _B : Setting prohibited 0111 _B : FIR 64TAP*1 1000 _B : IIR secondary biquad 1 stage 1001 _B : IIR secondary biquad 2 stages 1010 _B : IIR secondary biquad 3 stages 1011 _B to 1100 _B : Setting prohibited 1101 _B : IIR secondary biquad 2 stages with a gain adjustment function 1110 _B : IIR secondary biquad 3 stages with a gain adjustment function 1111 _B : Setting prohibited Note 1. FIR64TAP can be selected only on even-numbered channels. If FIR64TAP is selected on an even-numbered channel n, the use of the next odd-numbered channel n + 1 is prohibited.
11 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	FMT	FIR Input Data Format Select 0: Input data format = 16-bit fixed-point decimal number (value after a reset) 1: Input data format = 16-bit integer To execute accumulation processing or floating-point processing, specify 16-bit fixed-point.
7	IEP	PH End Interrupt Request Enable 0: Disables interrupt requests at the end of PH. (Value after a reset) 1: Enables interrupt requests at the end of PH. CAUTION: Enabled when PH and comparison are used simultaneously (DFEjTLBCHn.PRC5B = 11 _B).
6	IEE	Error Interrupt Request Enable 0: An error interrupt request is disabled: Value after a reset. 1: An error interrupt request is enabled.
5	IEC	Condition Match Interrupt Request Enable 0: A peak-hold end interrupt request is disabled when peak-hold operation is specified: value after a reset. When comparison calculation is specified, an interrupt request when the comparison calculation result is true is disabled: Value after a reset. 1: A peak-hold end interrupt request is enabled when peak-hold operation is specified. When a comparison calculation is specified, an interrupt request when the comparison calculation result is true is enabled.
4	IEO	Output Data Interrupt Request Enable 0: An output data interrupt when the result of calculation is written to the output data register (and the intermediate values output register) is disabled: value after a reset. 1: An output data interrupt when the result of calculation is written to the output data register (and the intermediate values output register) is enabled.

Table 37.13 DFEjCTLACHn Register Contents (3/3)

Bit Position	Bit Name	Function
3	CNSL	Condition Match Interrupt 0/1 Function Selection 0: Value after a reset INT_DFE_CND0: Selects a comparison match interrupt request. INT_DFE_CND1: Selects a PH end interrupt request. 1: INT_DFE_CND0: Selects a PH end interrupt request. INT_DFE_CND1: Selects a comparison match interrupt request. CAUTION: Enabled when PH and comparison are used simultaneously (DFEjCTLBCHn.PRC SB = 11 _B).
2	CNSLE	CNSL Enable 0: CNSL bit settings disabled. (Value after a reset) When PH and comparison are used simultaneously (DFEjCTLBCHn.PRC SB = 11 _B), the following functions are enabled: INT_DFE_CND0: Comparison match interrupt request INT_DFE_CND1: PH end interrupt request 1: CNSL bit settings enabled. Note: When either PH or comparison is specified (DFEjTLBCHn.PRC SB = 01 _B , 10 _B), INT_DFE_CND0/1 act as the following functions: INT_DFE_CND0: Comparison match interrupt request INT_DFE_CND1: No functions CAUTION: Enabled when PH and comparison are used simultaneously (DFEjCTLBCHn.PRC SB = 11 _B).
1	AIME	Automatic Initialization Enable 0: Automatic initialization function disabled. (Value after a reset) 1: Automatic initialization function enabled.
0	EN	Channel Enable 0: Channel disabled No processing is executed for this channel: Value after a reset. 1: Channel enabled Processing is executed for this channel.

When the automatic initialization function is enabled (DFEjCTLACHn.AIME = 1), on this channel the automatic initialization flag is enabled upon the rise of this bit (0 → 1).

Even when the automatic initialization function is disabled (DFEjCTLACHn.AIME = 0), the rise of this bit (0 → 1) is detected, and the automatic initialization flag is also enabled when AIME = 1 and EN = 1 are written simultaneously.

When the filtering request that is input is enabled after the automatic initialization flag is enabled, automatic initialization is executed during the pertinent filtering process.

CAUTION

If FIR64TAP is selected on an even-numbered channel n, it is prohibited to set the EN bit to 1 on the next odd-numbered channel n + 1 (even if the bit is set to 1, no trigger input will be accepted).

37.3.4 DFEjCTLBCHn — Control Register B n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1)

This register is used to set floating-point conversion, absolute value conversion, accumulation/decimation processing, and peak-hold/comparison processing.

All the bits can be set on even-numbered channels. However, the DFEjCTLBCHn.HOFS bit is not implemented on odd-numbered channels since those channels do not have an intermediate value output register.

Value after reset: 0101 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	—	—	OFSL		DISB	PHPS	CPCS	—	—	PHSLB2			DISA	
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R/W	R/W	R/W*	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	PRCSC	SELB2		SELB1		PRCSB		HOFS	PICS	SELA		PFMT	ABS	PRCSA			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Note: While the setting of DFEjCTLACHn.EN or DFEjSTCHn.VALID is 1, writing to any bit in the control register B is prohibited.

Table 37.14 DFEjCTLBCHn Register Contents (1/5)

Bit Position	Bit Name	Function
31 to 27	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
26, 25	OFSL	Comparison Offset Value α Selection 00 _B : Value after a reset DFEjCPOFST0 register value is selected for the comparison offset value α . 01 _B : DFEjCPOFST1 register value is selected for the comparison offset value α . 10 _B : DFEjCPOFST2 register value is selected for the comparison offset value α . 11 _B : Setting prohibited CAUTION: Enabled when PH and comparison are used simultaneously (DFEjCTLBCHn.PRCSB = 11 _B) and DFEjCTLBCHn.CPCS = 1

Table 37.14 DFEjCTLBCHn Register Contents (2/5)

Bit Position	Bit Name	Function
24	DISB	<p>Peak-Hold Processing Disable</p> <p>0: Indicates that peak-hold processing is in progress or peak-hold processing is enabled: DISB is cleared to 0 when the following conditions are satisfied while DISB is 1:</p> <ul style="list-style-type: none"> - The peak-hold initialization flag is enabled by a timer trigger or a software trigger. At this time, DISB is automatically cleared to 0. - 0 is written to DISB by the software. <p>When DISB is 0, peak-hold processing is started from the data that is input after DFEjCTLBCHn.PRC SB is set to 01_B (peak-hold processing executed) and DFEjCTLACHn.EN is set to 1.</p> <p>1: Indicates that peak-hold processing is disabled: Value after a reset. DISB is set to 1 when the following conditions are satisfied while DISB is 0:</p> <ul style="list-style-type: none"> - Reset input - The peak-hold end flag is enabled by a timer trigger or a software trigger. At this time, DISB is automatically set to 1. - 1 is written to DISB by setting the register by the software. <p>When DISB is 1, peak-hold processing is not started until the peak-hold initialization flag is enabled after DFEjCTLBCHn.PRC SB is set to 01_B (peak-hold processing executed) and DFEjCTLACHn.EN is set to 1.</p>
23	PHPS	<p>Peak-Hold Peak Type Select</p> <p>0: The PH processing detects upper-limit peaks (value after a reset).</p> <p>1: The PH processing detects lower-limit peaks.</p> <p>CAUTION: Enabled when PH is specified (DFEjCTLBCHn.PRC SB = 01_B or 11_B).</p>
22	CPCS	<p>Comparison Value Type Select</p> <p>0: Value after a reset DFEjC PA to DFEjC PD is selected by DFEjCTLBCHn.SELB1.</p> <p>1: DFEjPHCHn is used for calculating the comparison target value. For details, see SELB1.</p> <p>CAUTION: Enabled when PH and comparison are used simultaneously (DFEjCTLBCHn.PRC SB = 11_B).</p>
21, 20	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
19 to 17	PHSLB2	<p>PH Initial Value Register Select</p> <p>000_B: Value after a reset DFEjPHIA register value is selected for the PH processing initial value.</p> <p>001_B: DFEjPHIB register value is selected for the PH processing initial value.</p> <p>010_B: DFEjPHIC register value is selected for the PH processing initial value.</p> <p>011_B: DFEjPHID register value is selected for the PH processing initial value.</p> <p>100_B to 111_B: Setting prohibited</p> <p>CAUTION: Enabled when PH and comparison are used simultaneously (DFEjCTLBCHn.PRC SB = 11_B).</p>

Table 37.14 DFEjCTLBCHn Register Contents (3/5)

Bit Position	Bit Name	Function
16	DISA	<p>Accumulation/Decimation Processing Disable</p> <p>0: Indicates that accumulation/decimation processing is in progress or accumulation/decimation processing is enabled DISA is cleared to 0 when the following condition is satisfied while DISA is 1:</p> <ul style="list-style-type: none"> - The accumulation/decimation initialization flag is enabled by a timer trigger or a software trigger. At this time, DISA is automatically cleared to 0. - 0 is written to DISA by the software. <p>When DISA is 0, accumulation/decimation processing is started from the data that is input after DFEjCTLBCHn.PRCSA is set to 01_B (accumulation/decimation processing executed) and DFEjCTLACHn.EN is set to 1.</p> <p>1: Indicates that accumulation/decimation processing is disabled: Value after a reset. DISA is set to 1 when the following condition is satisfied while DISA is 0:</p> <ul style="list-style-type: none"> - Reset input - The accumulation/decimation disable flag is enabled by a timer trigger or a software trigger. At this time, DISA is automatically set to 1. - 1 is written to DISA by setting the register by the software. <p>When DISA is 1, accumulation/decimation processing is not started until the accumulation/decimation initialization flag is enabled after DFEjCTLBCHn.PRCSA is set to 01_B (accumulation or decimation processing executed) and DFEjCTLACHn.EN is set to 1.</p>
15	PRCSC	<p>Output Data Register Floating-Point Conversion</p> <p>0: Floating-point conversion for DFEjDOCHn is not executed: Value after a reset.</p> <p>1: Floating-point conversion for DFEjDOCHn is executed. Fixed-point format (1.31/10.22) is converted to floating-point (IEEE754). This bit must not be set to 1 if the setting of DFEjCTLACHn.FMT = 1 (integer mode) while FIR operation is selected.</p>
14 to 12	SELB2	<p>Peak-Hold Initialization Value Register Select</p> <p>When peak-hold operation is specified exclusively (DFEjCTLBCHn.PRCSB = 01_B)</p> <p>000_B: DFEjPHIA register value is selected for the initial value of peak-hold processing: Value after a reset.</p> <p>001_B: DFEjPHIB register value is selected for the initial value of peak-hold processing.</p> <p>010_B: DFEjPHIC register value is selected for the initial value of peak-hold processing.</p> <p>011_B: DFEjPHID register value is selected for the initial value of peak-hold processing.</p> <p>100_B to 111_B: Setting prohibited</p> <p>Comparison Calculation Select</p> <p>When comparison is specified exclusively (DFEjCTLBCHn.PRCSB = 10_B) or when PH and comparison are used simultaneously (DFEjCTLBCHn.PRCSB = 11_B).</p> <p>000_B: "Equal to (==)" is selected. Calculation of "equal to (==)" for the comparison target register selected by DFEjCTLBCHn.SELB1 and peak-hold circuit input data is performed: Value after a reset.</p> <p>001_B: "Equal to or less than (≤)" is selected. Calculation of "equal to or less than (≤)" for the comparison target register selected by DFEjCTLBCHn.SELB1 and peak-hold circuit input data is performed.</p> <p>010_B: "Equal to or more than (≥)" is selected. Calculation of "equal to or more than (≥)" for the comparison target register selected by DFEjCTLBCHn.SELB1 and peak-hold circuit input data is performed.</p> <p>011_B: "Less than (<)" is selected. Calculation of "less than (<)" for the comparison target register selected by DFEjCTLBCHn.SELB1 and peak-hold circuit input data is performed.</p> <p>100_B: "More than (>)" is selected. Calculation of "more than (>)" for the comparison target register selected by DFEjCTLBCHn.SELB1 and peak-hold circuit input data is performed.</p> <p>101_B to 111_B: Setting prohibited</p>

Table 37.14 DFEjCTLBCHn Register Contents (4/5)

Bit Position	Bit Name	Function
11, 10	SELB1	<p>Comparison Target Register Select</p> <p>When comparison is specified exclusively (DFEjCTLBCHn.PRCSB = 2b10) or when PH and comparison are used simultaneously (DFEjCTLBCHn.PRCSB = 2b11) and DFEjCTLBCHn.CPCS = 0.</p> <p>00_B: DFEjCPA register value is selected for the comparison calculation target value. 01_B: DFEjCPB register value is selected for the comparison calculation target value. 10_B: DFEjCPC register value is selected for the comparison calculation target value. 11_B: DFEjCPD register value is selected for the comparison calculation target value.</p> <p>Enabled when PH and comparison are used simultaneously (DFEjCTLBCHn.PRCSB = 2b11) and DFEjCTLBCHn.CPCS = 1</p> <p>00_B: The following value is selected for the comparison operation [Peak-Hold Result Register (DFEjPHCHn) + Comparison Offset Value Setting Register (DFEjCPOFSTn)]: Value after a reset. 01_B- 11_B: Setting prohibited.</p> <p>Note: Comparison offset value α is selected by Comparison offset value α Select (DFEjCTLBCHn.OFSL).</p>
9, 8	PRCSB	<p>Peak-Hold Circuit Processing Select</p> <p>00_B: Neither peak-hold processing nor comparison calculation processing is executed. 01_B: Peak-hold processing is executed. 10_B: Comparison calculation processing is executed. 11_B: Peak-Hold and comparison are executed simultaneously</p>
7	HOFS	<p>Intermediate Values Output Register Floating-Point Conversion</p> <p>0: Floating-point conversion for DFEjHOHCH and DFEjHOLCH is not executed: Value after a reset. 1: Floating-point conversion for DFEjMOHCH and DFEjMOLCH is executed. The setting of this bit is effective while DFEjCTLBCHn.PRCSC = 0. Fixed-point format is converted to floating-point (IEEE754). This bit must not be set to 1 if the setting of DFEjCTLACHn.FMT = 1 (integer mode) and DFEjCTLBCHn.PRCSC = 1 (output data register floating-point converted) when FIR is selected.</p> <p>Note: This bit is not implemented on odd-numbered channels.</p>
6	PICS	<p>Peak-Hold Index Register Control Select (not supported by channels 10 to 15)</p> <p>0: Peak-hold index updating mode (value after a reset) Following the completion of peak-hold processing, the peak-hold index register is cleared before the start of a next round of peak-hold processing. 1: Peak-hold index-retention mode After the completion of peak-hold processing, the peak-hold index register retains its value until it is read (clearing of the index register proceeds when the next round of peak-hold processing is started).</p>
5, 4	SELA	<p>Accumulation/Decimation Count Register Select</p> <p>00_B: DFEjACA value is selected for the accumulation/decimation count. 01_B: DFEjACB value is selected for the accumulation/decimation count. 10_B: DFEjACC value is selected for the accumulation/decimation count. 11_B: DFEjACD value is selected for the accumulation/decimation count.</p>

Table 37.14 DFEjCTLBCHn Register Contents (5/5)

Bit Position	Bit Name	Function
3	PFMT	<p>PH Result Register Floating-Point Conversion Execution</p> <p>0: Does not execute floating-point conversion on DFEjPHCH.</p> <p>1: Executes floating-point conversion on DFEjPHCH.</p> <p>Converts the fixed-point format (S.31/S9.22) to the floating-point format (IEEE754).</p> <p>When FIR is selected and DFEjTLACHn.FMT = 1 (integer mode), it is prohibited to set PFMT to 1.</p> <p>CAUTION: Enabled when PH is specified (DFEjCTLBCHn.PRC SB = 01_B or 11_B).</p>
2	ABS	<p>Absolute Value Calculation</p> <p>0: Absolute value calculation for the filter result is not performed.</p> <p>1: Absolute value calculation for the filter result is performed.</p>
1, 0	PRCSA	<p>Accumulation Circuit Processing Select</p> <p>00_B: Neither accumulation processing nor decimation processing is executed.</p> <p>01_B: Accumulation processing is executed.</p> <p>10_B: Decimation processing is executed.</p> <p>11_B: Setting prohibited.</p>

37.3.5 DFEjDOCHn — Output Data Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1)

This register is used to store the calculation results for each channel after DFE processing has been completed.

Value after reset: 0000 0000_H

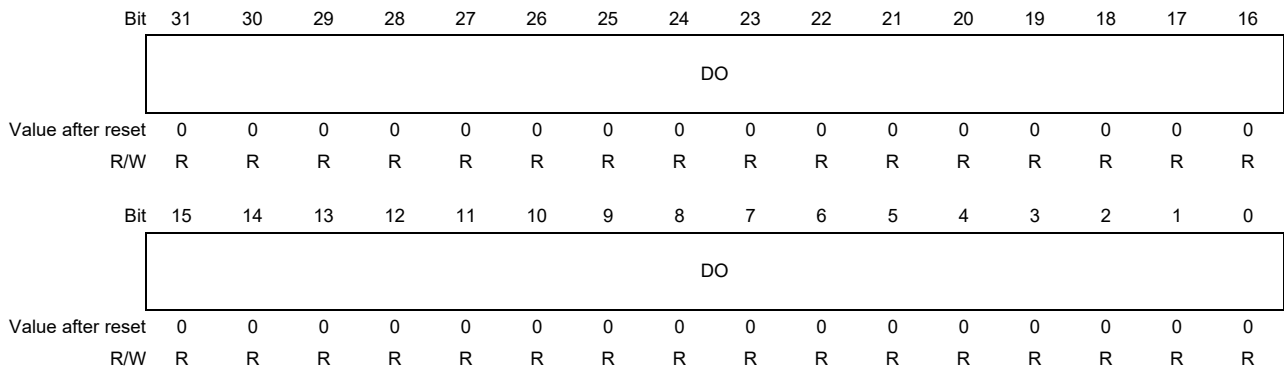


Table 37.15 DFEjDOCHn Register Contents

Bit Position	Bit Name	Function
31 to 0	DO	Output Data These bits show the DFE processing calculation results.

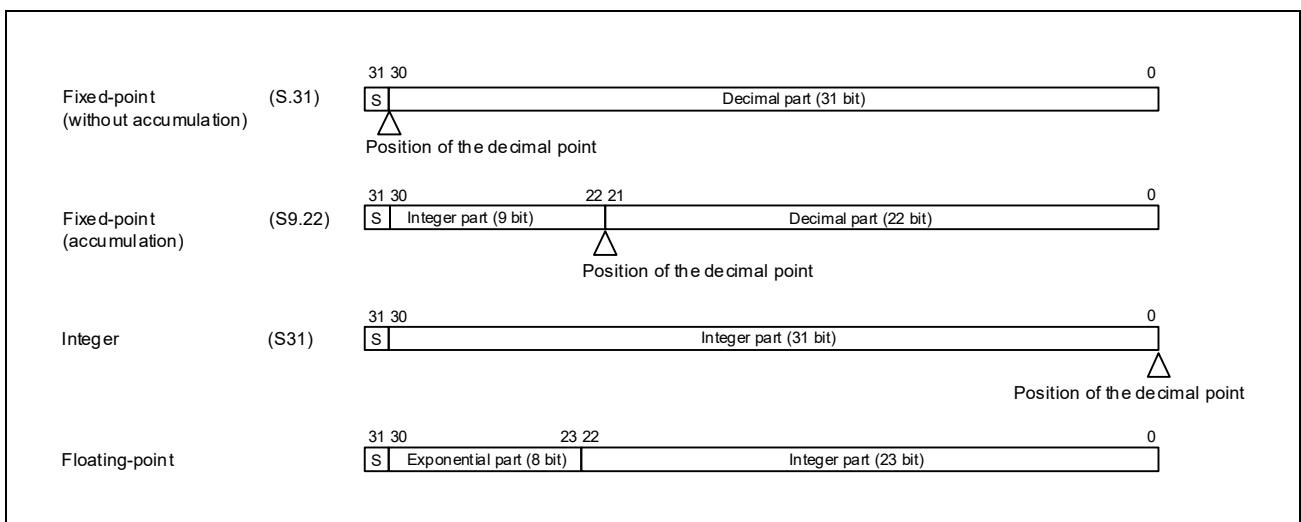


Figure 37.4 Format of Output Data Register Value

37.3.6 DFEjPHCHn — Peak-Hold Result Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1)

This register is used to store the peak-hold processing result value.

Value after reset: 8000 0000_H

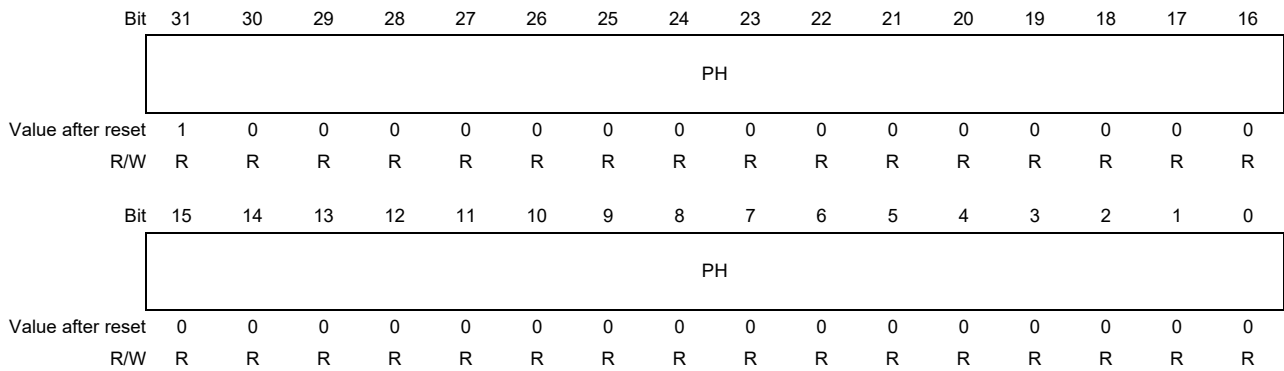


Table 37.16 DFEjPHCHn Register Contents

Bit Position	Bit Name	Function
31 to 0	PH	Peak-Hold Result Data These bits show the peak-hold processing calculation results.

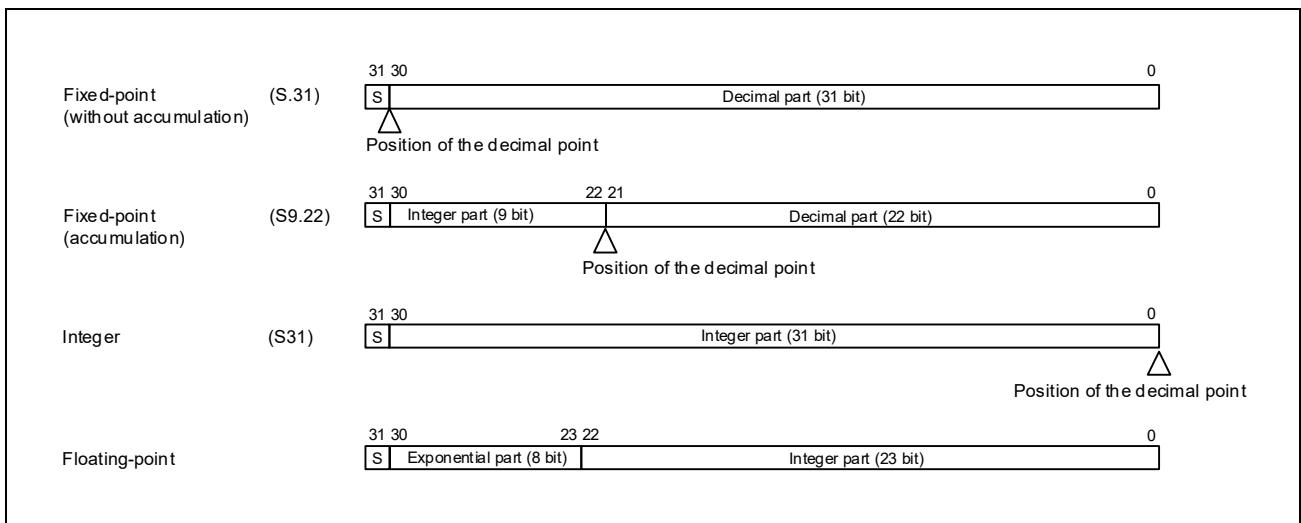


Figure 37.5 Format of Peak-Hold Result Register Value

37.3.7 DFEjPHINDCHn — Peak-Hold Index Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1)

These registers hold the number of rounds of peak-hold processing that have occurred when the peak-hold result register is updated.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PHIOW	PHIOF	PHIND													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.17 DFEjPHINDCHn Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15	PHIOW	Peak-Hold Index Overwrite Flag 0: PHIND has not been overwritten: Value after a reset. This flag becomes 0 when 1 is written to DFEjCLRSTCHn.CLRPHIOW. 1: PHIND has been overwritten in peak-hold index updating mode. This flag is valid in peak-hold index updating mode. It is always 0 in peak-hold index retention mode.
14	PHIOF	Peak-Hold Index Overflow Flag 0: The number of rounds of peak-hold processing has not overflowed (≤ 16383 times): Value after a reset. • If the peak-hold initialization flag is raised in peak-hold index updating mode, peak-hold processing clears it to 0. • In peak-hold index-retention mode, the peak-hold initialization flag is raised after the peak-hold index register is read, and processing of the first value for peak-hold processing clears it to 0. 1: The number of rounds of peak-hold processing has overflowed (> 16383 times). The number of rounds of peak-hold processing is counted from the time of input to the peak-hold initialization flag to the time of input to the peak-hold end flag.

Table 37.17 DFEjPHINDCHn Register Contents (2/2)

Bit Position	Bit Name	Function
13 to 0	PHIND	<p>Peak-Hold Index</p> <p>0_H: Value after a reset</p> <ul style="list-style-type: none"> In peak-hold index-updating mode: Indicates that the first condition for peak-hold updating is not satisfied when the peak-hold initialization flag has been raised. In peak-hold index-retention mode: Indicates that the first condition for peak-hold updating is not satisfied when the peak-hold initialization flag has been raised after the peak-hold index register has been read. <p>1_H: Indicates that the value for the first input to the peak-hold circuit has been written to the peak-hold result register.</p> <p>2_H: Indicates that the value for the second input to the peak-hold circuit has been written to the peak-hold result register.</p> <p>:</p> <p>3FFF_H: Indicates that the value for the 16383rd input to the peak-hold circuit has been written to the peak-hold result register.</p> <p>When the condition for peak-hold updating has been satisfied and the value is stored in the peak-hold result register, the DFEjPHIND register captures the number of values to which peak-hold processing has been applied (the value of the peak-hold counter within the digital filter).</p> <p>After the value of PHIOF becomes "1" when the number of rounds of peak-hold processing reaches and exceeds 16384, operation of the peak-hold counter continues and PHIND continues to be updated.</p>

37.3.8 DFEjSTCHn — Status Register n (n = 0 to 11, for DFE0, n = 0 to 3 for DFE1)

This register is used to indicate channel status.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	PH23ST		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	VALID	—	PHE	CND	DOEN	—	—	CER	AER	MER	GER	DOOW	DIOW
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.18 DFEjSTCHn Register Contents (1/4)

Bit Position	Bit Name	Function
31 to 19	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
18 to 16	PH23ST	<p>PH23 Processing Status</p> <p>000_B: This indicates that PH23 function is in an invalid state (value after a reset). When 0 is written to DFEjPH23CCTL0.PEN, PH23ST whose target channel of DFEjPH23CCTL0.CHS is set to 000_B. PH23ST[2:0] is set to 000_B, when the filter processing is executed after the auto initialization flag which is PH23 processing target channel is valid.</p> <p>001_B: This indicates that PH1 processing is in a valid state. This indicates that PH2 processing is in an invalid state. This indicates that PH3 processing is in an invalid state. PH23ST[2:0] is set to 001_B, when the filter processing whose PH1 initialization flag is valid is executed.</p> <p>010_B: This indicates that PH1 processing is in a finished state. This indicates that PH2 processing is in a running state. This indicates that PH3 processing is in an invalid state. PH23ST[2:0] is set to 010_B, when the comparison calculation result of PH1 is true.</p> <p>011_B: This indicates that PH1 processing is in a finished state. This indicates that PH2 processing is in a finished state. This indicates that PH3 processing is in a running state. PH23ST[2:0] is set to 011_B, when the comparison calculation result of PH2 is true.</p> <p>100_B: This indicates that PH1 processing is in a finished state. This indicates that PH2 processing is in a finished state. This indicates that PH3 processing is in a finished state. PH23ST[2:0] is set to 100_B, when the comparison calculation result of PH3 is true.</p> <p>101_B to 111_B: Other than above: Setting prohibited.</p> <p>CAUTION: The PH23ST is enabled only when the PH23 function is enabled (DFEjPH23CCTL0.PEN = 1 and DFEjPH23CCTL0.CHS is set to the target channel)</p>

Table 37.18 DFEjSTCHn Register Contents (2/4)

Bit Position	Bit Name	Function
15 to 13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12	VALID	Input Data Register Valid 0: Input data register value is invalid. This indicates that there is no data to be processed: Value after a reset. 1: Input data register value is valid. This bit is set to 1 when data to be processed is input to the input data register, and is cleared to 0 when the data has been processed in the accumulation circuit.
11	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
10	PHE	Peak-Hold End 0: Value after a reset This indicates that the peak-hold end flag is invalid. This is set to 0 when PHE = 1 and DFEjCLRSTCHn.CLRPHE is written to 1. 1: This indicates that the peak-hold end flag was valid and the filter processing was finished. This is set to 1 when PHE = 0 and the filter processing for the peak-hold is finished. CAUTION: Enabled when PH and comparison are used simultaneously (DFEjCTLBCHn.PRCBSB = 11 _B).
9	CND	Condition Match When peak-hold operation is specified exclusively (DFEjCTLBCHn.PRCBSB = 01 _B) 0: Value after a reset This indicates that the peak-hold end flag is invalid. This is set to 0 when CND = 1 and DFEjCLRSTCHn.CLRCOND is written to 1. 1: This indicates that that the peak-hold end flag was valid and the filter processing was finished. This is set to 1 when CND = 0 and the filter processing for the peak-hold was finished. Note: When PH and comparison are used simultaneously (DFEjCTLBCHn.PRCBSB = 11 _B), the PHE bit performs this function. When comparison is specified exclusively (DFEjCTLBCHn.PRCBSB = 10 _B) or when PH and comparison are used simultaneously (DFEjCTLBCHn.PRCBSB = 11 _B) 0: Value after a reset This indicates that the condition of a comparison calculation does not match. This is set to 0 when CND = 1 and DFEjCLRSTCHn.CLRCOND is written to 1. 1: This indicates that the condition of a comparison calculation is a match. This is set to 1 when CND = 0 and the comparison calculation result is true.

Table 37.18 DFEjSTCHn Register Contents (3/4)

Bit Position	Bit Name	Function
8	DOEN	<p>Output Data Register Valid</p> <ul style="list-style-type: none"> • Even-numbered channels <ul style="list-style-type: none"> 0: Indicates that the value of the output data register is invalid: Value after a reset. <ul style="list-style-type: none"> While DOEN = 1, writing 1 to DFEjCLRSTCHn.CLRDOEN clears this bit to 0. While DOEN = 1 and the IIR filter is selected, reading the DFEjDOCHn register clears this bit to 0. While DOEN = 1 and the FIR filter is selected, this bit is cleared in the following conditions. <ul style="list-style-type: none"> While DFEjTLBCHn.HOFS = 0, reading the DFEjDOCHn register or reading both the DFEjHOLCHn and DFEjHOHCHn registers clears this bit to 0. While DFEjTLBCHn.HOFS = 1, reading the DFEjDOCHn register or the DFEjHOLCHn register clears this bit to 0. This is set to 0 when DOEN = 1 and filter result data is stored in the buffer A circuit or buffer B circuit in the FIFO macro. 1: Indicates that the value of the output data register is valid. <ul style="list-style-type: none"> This bit is set to 1 when the DFE stores the results of a calculation in the output data register (and the intermediate values output register) while DFEjTLBCHn.PRCSA = 01_B (accumulation) or DFEjTLBCHn.PRCSA = 10_B (decimation). • Odd-numbered channels <ul style="list-style-type: none"> 0: Indicates that the output data register value is invalid: Value after a reset. <ul style="list-style-type: none"> Setting DFEjCLRSTCHn.CDOEN to 1 when DFEjCLRSTCHn.CLRDOEN = 1 clears this bit to 0. Reading the DFEjDOCHn register when DOEN = 1 clears this bit to 0. This is set to 0 when DOEN = 1 and filter result data is stored in the buffer A circuit or buffer B circuit in the FIFO macro. 1: Indicates that the output data register value is valid. <ul style="list-style-type: none"> This bit is set to 1 when the DFE writes the calculation result data to the output data register while DFEjTLBCHn.PRCSA is 01_B (accumulation) or DFEjTLBCHn.PRCSA is 10_B (decimation).
7, 6	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
5	CER	<p>Cascade Rounding Error</p> <ul style="list-style-type: none"> 0: No cascade rounding error is present: Value after a reset. <ul style="list-style-type: none"> Setting DFEjCLRSTCHn.CLR CER = 1 clears this bit to 0. 1: A cascade rounding error is present. <ul style="list-style-type: none"> This bit is set to 1 if an overflow occurs when the 32-bit accumulation circuit output data is rounded to 16-bit data and cascade is specified (DFEjTLACHn.CAEN = 10_B or 11_B).
4	AER	<p>Absolute Value Calculation Error</p> <ul style="list-style-type: none"> 0: No absolute value calculation error is present: Value after a reset. <ul style="list-style-type: none"> Setting DFEjCLRSTCHn.CLR AER = 1 clears this bit to 0. 1: An absolute value calculation error is present. <ul style="list-style-type: none"> This bit is set to 1 when the accumulation circuit input data is 8000 0000_H and absolute value calculation is specified (DFEjTLBCH.ABS = 1).
3	MER	<p>Multiplication Error</p> <ul style="list-style-type: none"> 0: No multiplication error is present: Value after a reset. <ul style="list-style-type: none"> Setting DFEjCLRSTCHn.CLR MER = 1 clears this bit to 0. 1: A multiplication error is present. <ul style="list-style-type: none"> This bit is set to 1 when multiplication 8000_H × 8000_H is performed in the filtering circuit.

Table 37.18 DFEjSTCHn Register Contents (4/4)

Bit Position	Bit Name	Function
2	GER	<p>Guard Error</p> <p>0: No guard error is present: Value after a reset. Setting DFEjCLRSTCHn.CLRGER = 1 clears this bit to 0.</p> <p>1: A guard error is present. This bit is set to 1 if an overflow or underflow occurs when the accumulation result is rounded to 32 bits in the filter circuit.</p>
1	DOOW	<p>Output Data Overwrite Error</p> <p>0: No output data overwrite error is present: Value after a reset. Setting DFEjCLRSTCHn.CLRDOOW = 1 clears this bit to 0.</p> <p>1: An output data overwrite error is present. This bit is set to 1 when the DFE stores the result of calculation in the output data register (and the intermediate values output register) while the output data register has a valid value (DFEjSTCHn.DOEN = 1) and DFEjCTLBCHn.PRCSA is 01_B (accumulation) or DFEjCTLBCHn.PRCSA is 10_B (decimation).</p>
0	DIOW	<p>Input Data Overwrite Error</p> <p>0: No input data overwrite is present: Value after a reset. Setting DFEjCLRSTCHn.CLRDIOW = 1 clears this bit to 0.</p> <p>1: An input data overwrite error is present. This bit is set to 1 when the value is stored again while the input data register has a valid value (VALID = 1).</p>

37.3.9 DFEjCLRSTCHn — Clear Status Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1)

This register is used to clear the status register. Writing 1 to each clear bit clears the corresponding bits in DFEjSTCHn and DFEjPHINDCHn to 0.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLR PHIOW	—	—	—	—	CLR PHE	CLR CND	CLR DOEN	—	—	CLR CER	CLR AER	CLR MER	CLR GER	CLR DOOW	CLR DIOW
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.19 DFEjCLRSTCHn Register Contents (1/2)

Bit Position	Bit Name	Function
15	CLRPHIOW	<p>When the PH23 function is enabled PH index overwrite flag clear</p> <p>0: Value after a reset Does not clear PHIOW in DFEjPHINDCHn, DFEjPH2IND, and DFEjPH3IND. 1: Clears PHIOW in DFEjPHINDCHn, DFEjPH2IND, and DFEjPH3IND. Clearing PHIOW in DFEjPHINDCHn, DFEjPH2IND, and DFEjPH3IND automatically sets CLRPHIOW to 0.</p> <p>When the PH23 function is disabled PH index overwrite flag clear</p> <p>0: Value after a reset Does not clear PHIOW in DFEjPHINDCHn. 1: Clears PHIOW in DFEjPHINDCHn. Clearing PHIOW in DFEjPHINDCHn automatically sets CLRPHIOW to 0. This bit is always read as 0.</p>
14 to 11	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
10	CLRPHE	<p>Enabled when PH and comparison are enabled simultaneously (DFEjCTLBCHn.PRC5B = 11_B).</p> <p>Peak-Hold end bit clear</p> <p>0: Value after a reset. Does not clear DFEjSTCHn.PHE. 1: Clears DFEjSTCHn.PHE. Clearing DFEjSTCHn.PHE automatically sets CLRPHE to 0. This bit is always read as 0.</p>

Table 37.19 DFEjCLRSTCHn Register Contents (2/2)

Bit Position	Bit Name	Function
9	CLRCND	Condition Match Bit Clear 0: DFEjSTCHn.CND is not cleared: Value after a reset. 1: DFEjSTCHn.CND is cleared. When DFEjSTCHn.CND is cleared, this bit is automatically cleared to 0. This bit is always read as 0.
8	CLRDOEN	Output Data Register Enable Bit Clear 0: DFEjSTCHn.DOEN is not cleared: Value after a reset. 1: DFEjSTCHn.DOEN is cleared. When DFEjSTCHn.DOEN is cleared, this bit is automatically cleared to 0. This bit is always read as 0.
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	CLRCER	Cascade Rounding Error Bit Clear 0: DFEjSTCHn.CER is not cleared: Value after a reset. 1: DFEjSTCHn.CER is cleared. This bit is always read as 0.
4	CLRAER	Absolute Value Calculation Error Bit Clear 0: DFEjSTCHn.AER is not cleared: Value after a reset. 1: DFEjSTCHn.AER is cleared. When DFEjSTCHn.AER is cleared, this bit is automatically cleared to 0. This bit is always read as 0.
3	CLRMER	Multiplication Error Bit Clear 0: DFEjSTCHn.MER is not cleared: Value after a reset. 1: DFEjSTCHn.MER is cleared. When DFEjSTCHn.MER is cleared, this bit is automatically cleared to 0. This bit is always read as 0.
2	CLRGER	Guard Error Bit Clear 0: DFEjSTCHn.GER is not cleared: Value after a reset. 1: DFEjSTCHn.GER is cleared. When DFEjSTCHn.GER is cleared, this bit is automatically cleared to 0. This bit is always read as 0.
1	CLRDOOW	Output Data Register Overwrite Error Bit Clear 0: DFEjSTCHn.DOOW is not cleared: Value after a reset. 1: DFEjSTCHn.DOOW is cleared. This bit is always read as 0.
0	CLRDIOW	Input Data Register Overwrite Error Bit Clear 0: DFEjSTCHn.DIOW is not cleared: Value after a reset. 1: DFEjSTCHn.DIOW is cleared. When DFEjSTCHn.DIOW is cleared, this bit is automatically cleared to 0. This bit is always read as 0.

37.3.10 DFEjERMCHn — Error Mask Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1)

This register is used to mask errors.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	MSK CER	MSK AER	MSK MER	MSK GER	MSK DOOW	MSK DIOW
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.20 DFEjERMCHn Register Contents

Bit Position	Bit Name	Function
7, 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	MSKCER	Cascade Rounding Error Bit (DFEjSTCHn.CER) Mask 0: The cascade rounding error bit (DFEjSTCHn.CER) is not masked: Value after a reset. 1: The cascade rounding error bit (DFEjSTCHn.CER) is masked.
4	MSKAER	Absolute Value Calculation Error Bit (DFEjSTCHn.AER) Mask 0: The absolute value calculation error bit (DFEjSTCHn.AER) is not masked: Value after a reset. 1: The absolute value calculation error bit (DFEjSTCHn.AER) is masked.
3	MSKMER	Multiplication Error Bit (DFEjSTCHn.MER) Mask 0: The multiplication error bit (DFEjSTCHn.MER) is not masked: Value after a reset. 1: The multiplication error bit (DFEjSTCHn.MER) is masked.
2	MSKGER	Guard Error Bit (DFEjSTCHn.GER) Mask 0: The guard error bit (DFEjSTCHn.GER) is not masked: Value after a reset. 1: The guard error bit (DFEjSTCHn.GER) is masked.
1	MSKDOOW	Output Data Overwrite Error Bit (DFEjSTCHn.DOOW) Mask 0: The output data overwrite error bit (DFEjSTCHn.DOOW) is not masked: Value after a reset. 1: The output data overwrite error bit (DFEjSTCHn.DOOW) is masked.
0	MSKDIOW	Input Data Overwrite Error Bit (DFEjSTCHn.DIOW) Mask 0: The input data overwrite error bit (DFEjSTCHn.DIOW) is not masked: Value after a reset. 1: The input data overwrite error bit (DFEjSTCHn.DIOW) is masked.

37.3.11 DFEjTRGCHn — Trigger Setting Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1)

This register is used to set the generation of trigger flags by using software triggers or timer triggers. It is prohibited to set both the initialization flag and end flag as a software trigger for the peak-hold processing. Furthermore, it is prohibited to set both the initialization flag and the disable flag as a software trigger for the accumulation/decimation processing. Any operation under prohibited settings is not guaranteed.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PMFE	PME	PFE	—	—			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AFE	—	—	PE	PT	AE	AT	FE	FT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.21 DFEjTRGCHn Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 24	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
23, 22	PMFE	Peak-Hold Mask End Flag Trigger Setting 00 _B : Value after a reset The peak-hold mask end flag is not generated. 01 _B : The peak-hold end flag is generated by a timer trigger. (timer triggers 0 to 5) 10 _B : The peak-hold end flag is generated by a software trigger. 11 _B : Setting prohibited It is prohibited to set both the Peak-Hold mask start flag and the Peak-Hold mask end flag to a software trigger. (It is prohibited to set DFEjTRGCHn.PMFE = 10 _B and DFEjTRGCHn.PME = 10 _B). When the peak-hold mask end flag is generated by a timer trigger (DFEjTRGCHn.PMFE = 01 _B), the peak-hold mask end flag is automatically selected from among compare signals B0 to B19 from the timer. Whichever signal corresponds to (has the same number as) the timer trigger (one of compare signals A0 to A19) selected by the peak-hold mask start flag and peak-hold mask end flag timer trigger select bit (DFEjTMTRGCHn.PMT) is selected.
21, 20	PME	Peak-Hold Mask Start Flag Trigger Setting 00 _B : Value after a reset The peak-hold mask start flag is not generated. 01 _B : The peak-hold start flag is generated by a timer trigger. (timer triggers 0 to 5) 10 _B : The peak-hold start flag is generated by a software trigger. 11 _B : Setting prohibited

Table 37.21 DFEjTRGCHn Register Contents (2/3)

Bit Position	Bit Name	Function
19, 18	PFE	<p>Peak-Hold End Flag Trigger Setting</p> <p>00_B: The peak-hold end flag is not generated: Value after a reset. 01_B: The peak-hold end flag is generated by a timer trigger. 10_B: The peak-hold end flag is generated by a software trigger. 11_B: Setting prohibited</p> <p>It is prohibited to set both the Peak-Hold initialization flag and the Peak-Hold end flag to a software trigger. (It is prohibited to set DFEjTRGCHn.PFE = 10_B and DFEjTRGCHn.PE = 10_B).</p> <p>When the peak-hold end flag is generated by a timer trigger (DFEjTRGCHn.PE = 01_B), the peak-hold end flag is automatically selected from among compare signals B0 to B19 from the timer. Whichever signal corresponds to (has the same number as) the timer trigger (one of compare signals A0 to A19) selected by the peak-hold initialization flag and peak-hold end flag timer trigger select bits (DFEjTRGCHn.PT and DFEjTMTRGCHn.PT) is selected.</p>
17, 16	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
15, 14	AFE	<p>Accumulation/Decimation Disable Flag Trigger Setting</p> <p>00_B: The accumulation/decimation disable flag is not generated: Value after a reset. 01_B: The accumulation/decimation disable flag is generated by a timer trigger. 10_B: The accumulation/decimation disable flag is generated by a software trigger. 11_B: Setting prohibited</p> <p>These bits disable both the accumulation/decimation initialization flag and the accumulation/decimation disable flag to be set as a software trigger. (Setting both AE and AFE bits in DFEjTRGCHn to 10_B is prohibited.)</p> <p>When the accumulation/decimation disable flag is generated by a timer trigger (DFEjTRGCHn.AFE = 01_B), the accumulation/decimation disable flag is automatically selected from among compare signals B0 to B19 from the timer. Whichever signal corresponds to (has the same number as) the timer trigger (one of compare signals A0 to A19) selected by the accumulation/decimation initialization flag and accumulation/decimation prohibition flag timer trigger select bits (DFEjTRGCHn.AT and DFEjTMTRGCHn.AT) is selected.</p>
13, 12	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
11, 10	PE	<p>Peak-Hold Initialization Flag Trigger Setting</p> <p>00_B: The peak-hold initialization flag is not generated: Value after a reset. 01_B: The peak-hold initialization flag is generated by a timer trigger. (Timer triggers 0 to 3) A timer trigger is selected from timer trigger 0-3 by setting DFEjTRGCHn.PT. 10_B: The peak-hold initialization flag is generated by a software trigger. 11_B: The peak-hold initialization flag is generated by a timer trigger. (Timer triggers 0 to 5) A timer trigger is selected from timer trigger 0-5 by setting DFEjTMTRGCHn.PT.</p>
9, 8	PT	<p>Peak-Hold Initialization Flag and Peak-Hold End Flag Timer Trigger Select</p> <p>00_B: Timer trigger 0 is selected for generating the peak-hold initialization flag and peak-hold end flag: Value after a reset. 01_B: Timer trigger 1 is selected for generating the peak-hold initialization flag and peak-hold end flag. 10_B: Timer trigger 2 is selected for generating the peak-hold initialization flag and peak-hold end flag. 11_B: Timer trigger 3 is selected for generating the peak-hold initialization flag and peak-hold end flag.</p> <p>These bits are valid when the peak-hold initialization flag trigger setting bit (DFEjTRGCHn.PE) or the peak-hold end flag trigger setting bit (DFEjTRGCHn.PFE) is 01_B. Timer triggers 0 to 3 can be selected by using PITMTRG0 to PITMTRG3 in DFEjPITRG.</p>

Table 37.21 DFEjTRGCHn Register Contents (3/3)

Bit Position	Bit Name	Function
7, 6	AE	<p>Accumulation/Decimation Initialization Flag Trigger Setting</p> <p>00_B: The accumulation/decimation initialization flag is not generated: Value after a reset.</p> <p>01_B: The accumulation/decimation initialization flag is generated by a timer trigger. (Timer triggers 0 to 3) A timer trigger is selected from timer trigger 0-3 by setting DFEjTRGCHn.AT.</p> <p>10_B: The accumulation/decimation initialization flag is generated by a software trigger.</p> <p>11_B: The accumulation/decimation initialization flag is generated by a timer trigger. (Timer triggers 0 to 5) A timer trigger is selected from timer trigger 0-5 by setting DFEjTMTRGCHn.AT.</p>
5, 4	AT	<p>Accumulation/Decimation Initialization Flag and Accumulation/Decimation Prohibition Flag Timer Trigger Select</p> <p>00_B: Timer trigger 0 is used for generating the accumulation/decimation initialization flag and accumulation/decimation prohibition flag: Value after a reset.</p> <p>01_B: Timer trigger 1 is used for generating the accumulation/decimation initialization flag and accumulation/decimation prohibition flag.</p> <p>10_B: Timer trigger 2 is used for generating the accumulation/decimation initialization flag and accumulation/decimation prohibition flag.</p> <p>11_B: Timer trigger 3 is used for generating the accumulation/decimation initialization flag and accumulation/decimation prohibition flag.</p> <p>These bits are valid when the accumulation/decimation initialization flag trigger setting bit (DFEjTRGCHn.AE) or the accumulation/decimation prohibition flag trigger setting bit (DFEjTRGCHn.AFE) is 01_B. Timer triggers 0 to 3 can be selected by using MITMTRG0 to MITMTRG3 in DFEjMITRG.</p>
3, 2	FE	<p>Filter Initialization Flag Trigger Setting</p> <p>00_B: The filter initialization flag is not generated: Value after a reset.</p> <p>01_B: The filter initialization flag is generated by a timer trigger. (Timer triggers 0 to 3) A timer trigger is selected from timer trigger 0-3 by setting DFEjTRGCHn.AT.</p> <p>10_B: The filter initialization flag is generated by a software trigger.</p> <p>11_B: The filter initialization flag is generated by a timer trigger (Timer triggers 0 to 5) A timer trigger is selected from timer trigger 0-5 by setting DFEjTMTRGCHn.AT.</p>
1, 0	FT	<p>Filter Initialization Flag Timer Trigger Select</p> <p>00_B: Timer trigger 0 is used for generating the filter initialization flag: Value after a reset.</p> <p>01_B: Timer trigger 1 is used for generating the filter initialization flag.</p> <p>10_B: Timer trigger 2 is used for generating the filter initialization flag.</p> <p>11_B: Timer trigger 3 is used for generating the filter initialization flag. Timer triggers 0 to 3 can be selected by using FITMTRG0 to FITMTRG3 in DFEjFITRG.</p> <p>Note: The filter initialization flag can be enabled when DFEjTRGCHn.FE is 01_B.</p>

37.3.12 DFEjTMTRGCHn — Timer Trigger Setting Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1)

This register is used to select timer triggers. The bits in this register represent a functional expansion of the same bits of DFEjTRGCHn.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PMT		—	PT		—	AT		—	FT		—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 37.22 DFEjTMTRGCHn Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 15	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
14 to 12	PMT	Peak-Hold Mask Start Flag and Peak-Hold Mask End Flag timer trigger Select 000 _B : Value after a reset Timer trigger 0 is used for generating the peak-hold mask start flag and the peak-hold mask end flag. 001 _B : Timer trigger 1 is used for generating the peak-hold mask start flag and the peak-hold mask end flag. 010 _B : Timer trigger 2 is used for generating the peak-hold mask start flag and the peak-hold mask end flag. 011 _B : Timer trigger 3 is used for generating the peak-hold mask start flag and the peak-hold mask end flag. 100 _B : Timer trigger 4 is used for generating the peak-hold mask start flag and the peak-hold mask end flag. 101 _B : Timer trigger 5 is used for generating the peak-hold mask start flag and the peak-hold mask end flag. 110 _B to 111 _B : Setting prohibited. CAUTION: The peak-hold mask start flag can be enabled when DFEjTRGCHn.PME is 01 _B . The peak-hold mask ending flag can be enabled when DFEjTRGCHn.PMFE is 01 _B . Timer triggers 0 to 3 can be selected by using DFEjPMITRG0.PMITMTRG0 to PMITMTRG3. Timer triggers 4 to 5 can be selected by using DFEjPMITRG1.PMITMTRG4 to PMITMTRG5
11	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 37.22 DFEjTMTRGCHn Register Contents (2/3)

Bit Position	Bit Name	Function
10 to 8	PT	<p>Peak-Hold Initialization Flag and Peak-Hold End Flag Timer Trigger Select</p> <p>000_B: Timer trigger 0 is selected for generating the peak-hold initialization flag and peak-hold end flag.</p> <p>001_B: Timer trigger 1 is selected for generating the peak-hold initialization flag and peak-hold end flag.</p> <p>010_B: Timer trigger 2 is selected for generating the peak-hold initialization flag and peak-hold end flag.</p> <p>011_B: Timer trigger 3 is selected for generating the peak-hold initialization flag and peak-hold end flag.</p> <p>100_B: Timer trigger 4 is selected for generating the peak-hold initialization flag and peak-hold end flag.</p> <p>101_B: Timer trigger 5 is selected for generating the peak-hold initialization flag and peak-hold end flag.</p> <p>110_B to 111_B: Setting prohibited</p> <p>CAUTION: The peak-hold initialization flag is enabled when DFEjTRGCHn.PE is 11_B. The peak-hold end flag is enabled when DFEjTRGCHn.PFE is 01_B. Timer triggers 0 to 3 can be selected by using DFEjPITRG.PITMTRG0 to PITMTRG3. Timer triggers 4 to 5 can be selected by using DFEjPITRG1.PITMTRG4 to PITMTRG5.</p>
7	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
6 to 4	AT	<p>Accumulation/Decimation Initialization Flag and Accumulation/Decimation Prohibition Flag Timer Trigger Select</p> <p>000_B: Value after a reset Timer trigger 0 is used for generating the accumulation/decimation initialization flag and accumulation/decimation prohibition flag.</p> <p>001_B: Timer trigger 1 is used for generating the accumulation/decimation initialization flag and accumulation/decimation prohibition flag.</p> <p>010_B: Timer trigger 2 is used for generating the accumulation/decimation initialization flag and accumulation/decimation prohibition flag.</p> <p>011_B: Timer trigger 3 is used for generating the accumulation/decimation initialization flag and accumulation/decimation prohibition flag.</p> <p>100_B: Timer trigger 4 is used for generating the accumulation/decimation initialization flag and accumulation/decimation prohibition flag.</p> <p>101_B: Timer trigger 5 is used for generating the accumulation/decimation initialization flag and accumulation/decimation prohibition flag.</p> <p>110_B to 111_B: Setting prohibited</p> <p>CAUTION: The accumulation/decimation initialization flag can be enabled when DFEjTRGCHn.AE is 11_B. The accumulation/decimation disable flag can be enabled when DFEjTRGCHn.AFE is 01_B. Timer triggers 0 to 3 can be selected by using DFEjMITRG.MITMTRG0 to MITMTRG3. Timer triggers 4 to 5 can be selected by using DFEjMITRG1.MITMTRG4 to MITMTRG5.</p>
3	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>

Table 37.22 DFEjTMTRGCHn Register Contents (3/3)

Bit Position	Bit Name	Function
2 to 0	FT	<p>Filter Initialization Flag Timer Trigger Select</p> <p>000_B: Value after a reset</p> <p>Timer trigger 0 is used for generating the filter initialization flag.</p> <p>001_B: Timer trigger 1 is used for generating the filter initialization flag.</p> <p>010_B: Timer trigger 2 is used for generating the filter initialization flag.</p> <p>011_B: Timer trigger 3 is used for generating the filter initialization flag.</p> <p>100_B: Timer trigger 4 is used for generating the filter initialization flag.</p> <p>101_B: Timer trigger 5 is used for generating the filter initialization flag.</p> <p>110_B to 111_B: Setting prohibited</p> <p>CAUTION: The filter initialization flag can be enabled when DFEjTRGCHn.FE is 11_B. Timer triggers 0 to 3 can be selected by using DFEjFITRG.FITMTRG0 to FITMTRG3. Timer triggers 4 to 5 can be selected by using DFEjFITRG1.FITMTRG4 to FITMTRG5.</p>

37.3.13 DFEjTRHCHn — Trigger History Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1)

This register is used to monitor the execution history of software triggers, timer triggers, and trigger flags.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	PMITS	PMETS	PITS	PETS	MITS	METS	FITS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 37.23 DFEjTRHCHn Register Contents (1/2)

Bit Position	Bit Name	Function
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	PMITS	Peak-Hold Mask Start flag trigger History 0: Value after a reset The peak-hold mask start flag trigger is not present. This bit is cleared to 0 when execution of the peak-hold mask start flag has been completed while PMITS = 1. This bit is also cleared to 0 if DFEjCTLACHn.EN is set to 0 and there is no input data which is undergoing DFE processing. 1: This bit is set to 1 when a peak-hold mask start flag trigger occurs.
5	PMETS	Peak-Hold Mask End flag trigger History 0: Value after a reset The peak-hold mask end flag trigger is not present. This bit is cleared to 0 when execution of the peak-hold mask end flag has been completed while PMETS = 1. This bit is also cleared to 0 if DFEjCTLACHn.EN is set to 0 and there is no input data which is undergoing DFE processing. 1: This bit is set to 1 when a peak-hold mask end flag trigger occurs.
4	PITS	Peak-Hold Initialization Flag Trigger History 0: Peak-hold initialization flag trigger is not present: Value after a reset. This bit is cleared to 0 when execution of the peak-hold initialization flag has been completed while PITS = 1. This bit is also cleared to 0 if EN in DFEjCTLACHn is set to 0 and there is no input data which is undergoing DFE processing. 1: This bit is set to 1 when a peak-hold initialization flag trigger occurs.
3	PETS	Peak-Hold End Flag Trigger History 0: Peak-hold end flag trigger is not present: Value after a reset. This bit is cleared to 0 when execution of the peak-hold end flag has been completed while PETS = 1. This bit is also cleared to 0 if EN in DFEjCTLACHn is set to 0 and there is no input data which is undergoing DFE processing. 1: This bit is set to 1 when a peak-hold end flag trigger occurs.
2	MITS	Accumulation/Decimation Initialization Flag Trigger History 0: Accumulation/decimation initialization flag trigger is not present: value after a reset. This bit is cleared to 0 when execution of the accumulation/decimation initialization flag has been completed while MITS = 1. This bit is also cleared to 0 if EN in DFEjCTLACHn is set to 0 and there is no input data which is in DFE processing. 1: This bit is set to 1 when an accumulation/decimation initialization flag trigger occurs.

Table 37.23 DFEjTRHCHn Register Contents (2/2)

Bit Position	Bit Name	Function
1	METS	<p>Accumulation/Decimation Disable Flag Trigger History</p> <p>0: Accumulation/decimation disable flag trigger is not present: Value after a reset. This bit is cleared to 0 when execution of the accumulation/decimation disable flag has been completed while METS = 1. This bit is also cleared to 0 if EN in DFEjCTLACHn is set to 0 and there is no input data which is undergoing DFE processing.</p> <p>1: This bit is set to 1 when an accumulation/decimation disable flag trigger occurs.</p>
0	FITS	<p>Filter Initialization Flag Trigger History</p> <p>0: Filter initialization flag trigger is not present: Value after a reset. This bit is cleared to 0 when execution of the filter initialization flag has been completed while FITS = 1. This bit is also cleared to 0 if EN in DFEjCTLACHn is set to 0 and there is no input data which is undergoing DFE processing.</p> <p>1: This bit is set to 1 when a filter initialization flag trigger occurs.</p>

37.3.14 DFEjCPA to DFEjCPD — Comparison Value Setting Register

These registers are used to set a comparison calculation target value when comparison calculation is specified.

Value after reset: 0000 0000_H

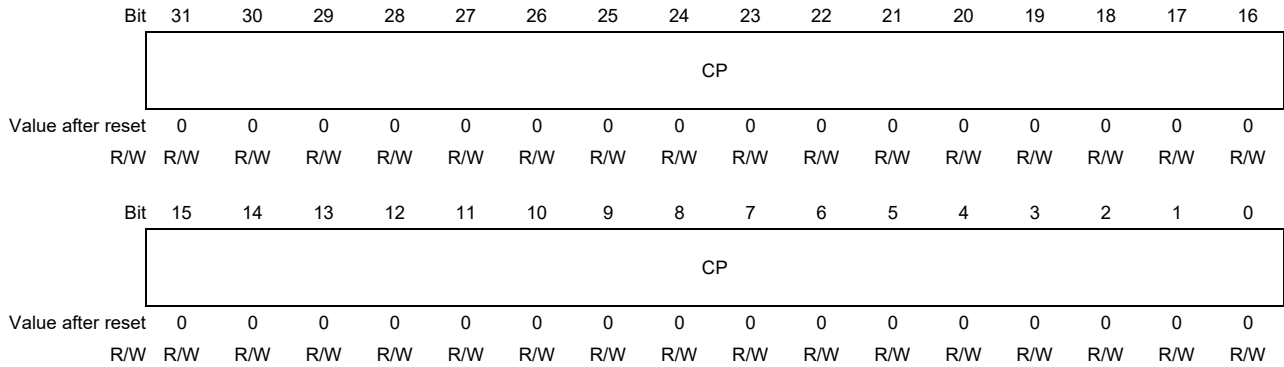


Table 37.24 DFEjCPA to DFEjCPD Register Contents

Bit Position	Bit Name	Function
31 to 0	CP	Comparison Value These bits set the comparison value when comparison calculation is specified.

The data format of the comparison value setting registers is as follows.

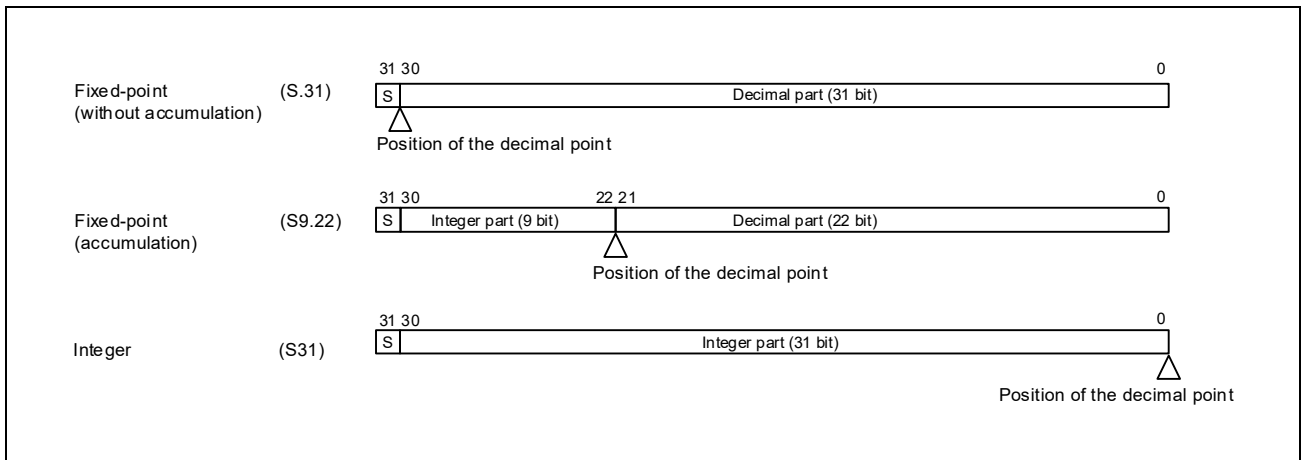


Figure 37.6 Data Format of Comparison Value Setting Registers

37.3.15 DFEjPHIA to DFEjPHID — Peak-Hold Initial Value Setting Register

These registers are used to set the initial peak-hold value when peak-hold calculation is specified.

Value after reset: 8000 0000_H

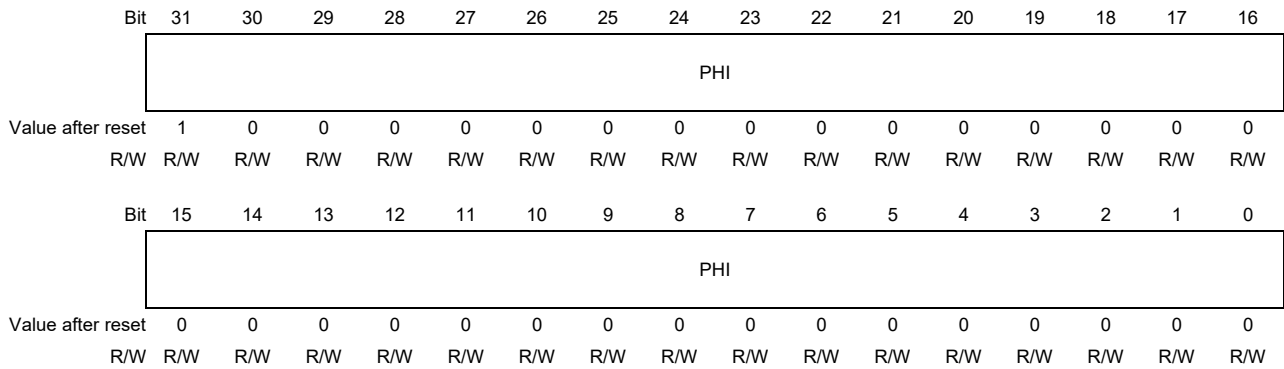


Table 37.25 DFEjPHIA to DFEjPHID Register Contents

Bit Position	Bit Name	Function
31 to 0	PHI	Initial Peak-Hold Value These bits set the initial peak-hold value when Peak-hold calculation is specified. Initial value: Negative minimum value (8000 0000 _H)

The data format of the peak-hold initial value setting registers is as follows.

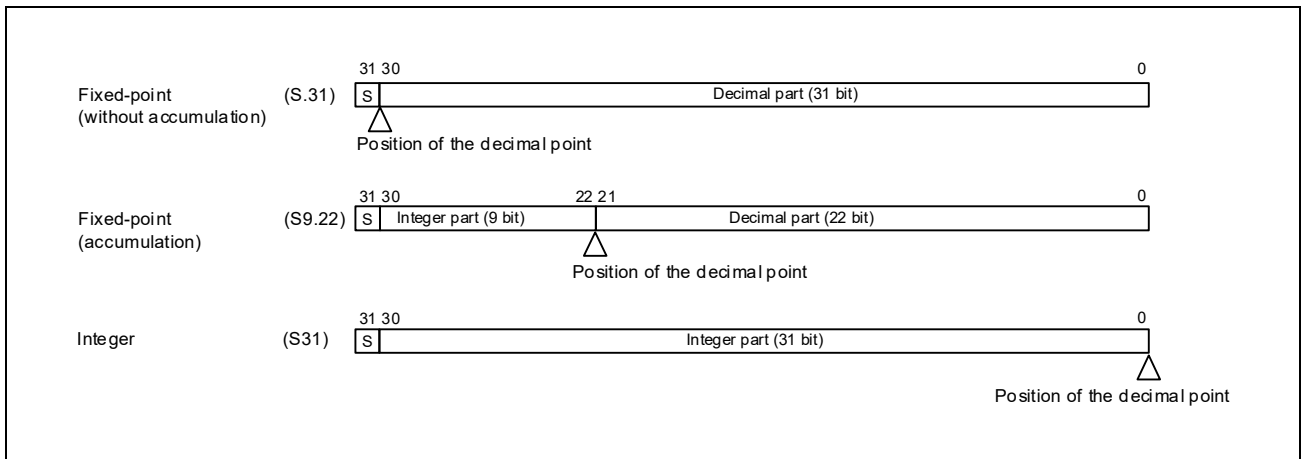


Figure 37.7 Data Format of Peak-Hold Initial Value Setting Registers

37.3.16 DFEjACA to DFEjACD — Accumulation/Decimation Count Setting Register

These registers are used to set accumulation count or decimation count when accumulation/decimation is specified. A value of 0 to 511 can be set in these registers.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	AC								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.26 DFEjACA to DFEjACD Register Contents

Bit Position	Bit Name	Function
15 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	AC	<p>In the case of accumulation (DFEjCTLACHn.PRCSA = 01_B) Accumulation Count Setting Register</p> <p>000_H: Accumulation processing is not performed*1: Value after a reset. 001_H: Accumulation is performed once. Two filter processing results are added and output from the accumulation circuit. 002_H: Accumulation is performed twice. Three filter processing results are added and output from the accumulation circuit. : 1FF_H: Accumulation is performed 511 times. 512 filter processing results are added and output from the accumulation circuit.</p> <p>In the case of decimation (DFEjCTLACHn.PRCSA = 10_B) Decimation Count Setting Register</p> <p>000_H: Decimation processing is not performed: Value after a reset. 001_H: Decimation is performed once. Filter processing results are output once every two filter processing results. 002_H: Decimation is performed twice. Filter processing results are output once every three filter processing results. : 1FF_H: Decimation is performed 511 times. Filter processing results are output once every 512 filter processing results.</p>

Note 1. However, since arithmetic right shift is executed during accumulation, the format of data that is input to the accumulation circuit is (S.31) data format to (S9.22) data format.

37.3.17 DFEjDI — Software Input Data Register

This register is used to input data to process (software input) from the CPU or DMA to the DFE through the peripheral bus. A write access to this register activates the DFE.

Write access to the TAG[4:0] and DI[15:0] bits should be made at the same time.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	TAG				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DI															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.27 DFEjDI Register Contents

Bit Position	Bit Name	Function
31 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20 to 16	TAG	Software Input Tag 00 _H to 1F _H : Set the same value as the channel tag (DFEjCTLACHn.TAG) to be allocated.
15 to 0	DI	Software Input Data Data to be filtered is set with the input data format of the channel to be allocated. The fixed-point (S.31) format or the integer format is applicable. Set data to be filtered with the data format of the channel to be allocated. (If DFEjCTLACHn.FMT = 0 while FIR is selected, set data in fixed-point format; if DFEjCTLACHn.FMT = 1, set data to the integer format. While IIR is selected, the fixed-point format is fixed.)

37.3.18 DFEJTRG — Software Trigger Register

This register is used to generate a software trigger by writing a value to this register from the CPU or DMA through the peripheral bus.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TRGA
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 37.28 DFEJTRG Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	TRGA	Software Trigger Generation 0: No software trigger is generated. 1: A software trigger is generated. Writing 1 to this bit generates a software trigger. This bit is always read as 0.

37.3.19 DFEjST — DFE Status Register

This register is used to indicate the DFE status. This register consists of a logical OR of each VALID bit and each error bit in DFEjSTCHn. The CPU application may use this register to monitor error channels when the DFE outputs an error interrupt request.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—				—				CH11S*1		CH10S*1		CH9S*1		CH8S*1	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH7S*1		CH6S*1		CH5S*1		CH4S*1		CH3S		CH2S		CH1S		CH0S	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. DFE0 only. DFE1 is the reserved.

Table 37.29 DFEjST Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2×n+1 to 2×n	CHnS	Channel n Status 00 _B : Input data register disabled with no errors: Value after a reset 01 _B : Input data register disabled with an error 10 _B : Input data register enabled with no errors 11 _B : Input data register enabled with an error

37.3.20 DFEjEST — DFE Error Status Register

This register indicates the error status on the DFE subtraction channel and the capture channel. The register is intended for an application where the CPU uses the register to monitor the error-generating channel when the DFE issues an error interrupt request.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	CP2S	CP1S	CP0S
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SB2S	SB1S	SB0S
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.30 DFEjEST Register Contents

Bit Position	Bit Name	Function
31 to 19	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
18 to 16	CPnS	Capture Channel-n Status 0: There are no capture errors. Value after a reset Capture channel-n has no error. 1: Capture channel-n has an error.
15 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	SBnS	Subtraction Channel-n Status 0: There are no subtraction errors. Value after a reset Subtraction Channel-n has no error. 1: Subtraction Channel-n has an error.

37.3.21 DFEjPMITRG0 — Peak-Hold Mask-Start/End Timer Trigger Select Register 0

This register selects which of the 20 timer trigger compare interfaces--compare signals A0/B0 to A19/B19 in the timer--to use for peak-hold mask start/end timer triggers 0 to 3.

Timer triggers 4 and 5 are set using the peak-hold mask start/end timer trigger select register 1 (DFEjPMITRG1).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	PMITMTRG3						—	—	—	PMITMTRG2				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	—	PMITMTRG1						—	—	—	PMITMTRG0				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	

Table 37.31 DFEjPMITRG0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
28 to 24	PMITMTRG3	Peak-Hold Mask Start/End Flag Timer Trigger 3 00000 _B : Value after a reset Uses compare A0/B0 of the Timer for the peak-hold mask start/end timer trigger 3. 00001 _B : Uses compare A1/B1 of the Timer for the peak-hold mask start/end timer trigger 3. ... 10011 _B : Uses compare A19/B19 of the Timer for the peak-hold mask start/end timer trigger 3. 10100 _B to 11111 _B : Setting prohibited.
23 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20 to 16	PMITMTRG2	Peak-Hold Mask Start/End Flag Timer Trigger 2 00000 _B : Value after a reset Uses compare A0/B0 of the Timer for the peak-hold mask start/end timer trigger 2. 00001 _B : Uses compare A1/B1 of the Timer for the peak-hold mask start/end timer trigger 2. ... 10011 _B : Uses compare A19/B19 of the Timer for the peak-hold mask start/end timer trigger 2. 10100 _B to 11111 _B : Setting prohibited.
15 to 13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	PMITMTRG1	Peak-Hold Mask Start/End Flag Timer Trigger 1 00000 _B : Value after a reset Uses compare A0/B0 of the Timer for the peak-hold mask start/end timer trigger 1. 00001 _B : Uses compare A1/B1 of the Timer for the peak-hold mask start/end timer trigger 1. ... 10011 _B : Uses compare A19/B19 of the Timer for the peak-hold mask start/end timer trigger 1. 10100 _B to 11111 _B : Settings prohibited.
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 37.31 DFEjPMITRG0 Register Contents (2/2)

Bit Position	Bit Name	Function
4 to 0	PMITMTRG0	Peak-Hold Mask Start/End Flag Timer Trigger 0 00000 _B : Value after a reset Uses compare A0/B0 of the Timer for the peak-hold mask start/end timer trigger 0. 00001 _B : Uses compare A1/B1 of the Timer for the peak-hold mask start/end timer trigger 0. ... 10011 _B : Uses compare A19/B19 of the Timer for the peak-hold mask start/end timer trigger 0. 10100 _B to 11111 _B : Setting prohibited.

37.3.22 DFEjPMITRG1 — Peak-Hold Mask-Start/End Timer Trigger Select Register 1

This register selects which of the 20 timer trigger compare interfaces--compare signals A0/B0 to A19/B19 in the timer--to use for peak-hold mask start/end timer triggers 4 to 5.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PMITMTRG5					—	—	—	PMITMTRG4				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 37.32 DFEjPMITRG1 Register Contents

Bit Position	Bit Name	Function
31 to 13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	PMITMTRG5	Peak-Hold Mask Start/End Flag Timer Trigger 5 0000 _B : Value after a reset Uses compare A0/B0 of the Timer for the peak-hold mask start/end timer trigger 5. 00001 _B : Uses compare A1/B1 of the Timer for the peak-hold mask start/end timer trigger 5. ... 10011 _B : Uses compare A19/B19 of the Timer for the peak-hold mask start/end timer trigger 5. 10100 _B to 11111 _B : Setting prohibited.
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	PMITMTRG4	Peak-Hold Mask Start/End Flag Timer Trigger 4 0000 _B : Value after a reset Uses compare A0/B0 of the Timer for the peak-hold mask start/end timer trigger 4. 00001 _B : Uses compare A1/B1 of the Timer for the peak-hold mask start/end timer trigger 4. ... 10011 _B : Uses compare A19/B19 of the Timer for the peak-hold mask start/end timer trigger 4. 10100 _B to 11111 _B : Setting prohibited.

37.3.23 DFEjPITRG — Peak-Hold Initialization/End Timer Trigger Select Register

This register selects which of the 20 timer trigger compare interfaces--compare signals A0/B0 to A19/B19 in the timer--to use for peak-hold initialization/end timer triggers 0 to 3.

Timer triggers 4 and 5 are set using the peak-hold initialization/end timer trigger select register 1 (DFEjPITRG1).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	PITMTRG3						—	—	—	PITMTRG2				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	—	PITMTRG1						—	—	—	PITMTRG0				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	

Table 37.33 DFEjPITRG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
28 to 24	PITMTRG3	Peak-Hold(PH) Initialization/End Flag Timer Trigger 3 0000 _B : Uses compare A0/B0 of the Timer for the PH initialization/end timer trigger 3: Value after a reset. 00001 _B : Uses compare A1/B1 of the Timer for the PH initialization/end timer trigger 3. 00010 _B : Uses compare A2/B2 of the Timer for the PH initialization/end timer trigger 3. 00011 _B : Uses compare A3/B3 of the Timer for the PH initialization/end timer trigger 3. 00100 _B : Uses compare A4/B4 of the Timer for the PH initialization/end timer trigger 3. 00101 _B : Uses compare A5/B5 of the Timer for the PH initialization/end timer trigger 3. 00110 _B : Uses compare A6/B6 of the Timer for the PH initialization/end timer trigger 3. 00111 _B : Uses compare A7/B7 of the Timer for the PH initialization/end timer trigger 3. 01000 _B : Uses compare A8/B8 of the Timer for the PH initialization/end timer trigger 3. 01001 _B : Uses compare A9/B9 of the Timer for the PH initialization/end timer trigger 3. 01010 _B : Uses compare A10/B10 of the Timer for the PH initialization/end timer trigger 3. 01011 _B : Uses compare A11/B11 of the Timer for the PH initialization/end timer trigger 3. 01100 _B : Uses compare A12/B12 of the Timer for the PH initialization/end timer trigger 3. 01101 _B : Uses compare A13/B13 of the Timer for the PH initialization/end timer trigger 3. 01110 _B : Uses compare A14/B14 of the Timer for the PH initialization/end timer trigger 3. 01111 _B : Uses compare A15/B15 of the Timer for the PH initialization/end timer trigger 3. 10000 _B : Uses compare A16/B16 of the Timer for the PH initialization/end timer trigger 3. 10001 _B : Uses compare A17/B17 of the Timer for the PH initialization/end timer trigger 3. 10010 _B : Uses compare A18/B18 of the Timer for the PH initialization/end timer trigger 3. 10011 _B : Uses compare A19/B19 of the Timer for the PH initialization/end timer trigger 3. 10100 _B to 11111 _B : Setting prohibited
23 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 37.33 DFEjPITRG Register Contents (2/2)

Bit Position	Bit Name	Function
20 to 16	PITMTRG2	<p>Peak-Hold (PH) Initialization/End Flag Timer Trigger 2</p> <p>00000_B: Uses compare A0/B0 of the Timer for the PH initialization/end timer trigger 2: Value after a reset.</p> <p>00001_B: Uses compare A1/B1 of the Timer for the PH initialization/end timer trigger 2.</p> <p>00010_B: Uses compare A2/B2 of the Timer for the PH initialization/end timer trigger 2.</p> <p>:</p> <p>10011_B: Uses compare A19/B19 of the Timer for the PH initialization/end timer trigger 2.</p> <p>10100_B to 11111_B: Setting prohibited</p>
15 to 13	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
12 to 8	PITMTRG1	<p>Peak-Hold (PH) Initialization/End Flag Timer Trigger 1</p> <p>00000_B: Uses compare A0/B0 of the Timer for the PH initialization/end timer trigger 1: Value after a reset.</p> <p>00001_B: Uses compare A1/B1 of the Timer for the PH initialization/end timer trigger 1.</p> <p>00010_B: Uses compare A2/B2 of the Timer for the PH initialization/end timer trigger 1.</p> <p>:</p> <p>10011_B: Uses compare A19/B19 of the Timer for the PH initialization/end timer trigger 1.</p> <p>10100_B to 11111_B: Setting prohibited</p>
7 to 5	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
4 to 0	PITMTRG0	<p>Peak-Hold (PH) Initialization/End Flag Timer Trigger 0</p> <p>00000_B: Uses compare A0/B0 of the Timer for the PH initialization/end timer trigger 0: Value after a reset.</p> <p>00001_B: Uses compare A1/B1 of the Timer for the PH initialization/end timer trigger 0.</p> <p>00010_B: Uses compare A2/B2 of the Timer for the PH initialization/end timer trigger 0.</p> <p>:</p> <p>10011_B: Uses compare A19/B19 of the Timer for the PH initialization/end timer trigger 0.</p> <p>10100_B to 11111_B: Setting prohibited</p>

37.3.24 DFEjPITRG1 — Peak-Hold Initialization/End Timer Trigger Select Register 1

This register selects which of the 20 timer trigger compare interfaces--compare signals A0/B0 to A19/B19 in the timer--to use for peak-hold initialization/end timer triggers 4 to 5.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PITMTRG5					—	—	—	PITMTRG4				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 37.34 DFEjPITRG1 Register Contents

Bit Position	Bit Name	Function
31 to 13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	PITMTRG5	Peak-Hold (PH) Initialization/End Flag Timer Trigger 5 0000 _B : Uses compare A0/B0 of the Timer for the PH initialization/end timer trigger 5: Value after a reset. 00001 _B : Uses compare A1/B1 of the Timer for the PH initialization/end timer trigger 5 00010 _B : Uses compare A2/B2 of the Timer for the PH initialization/end timer trigger 5. : 10011 _B : Uses compare A19/B19 of the Timer for the PH initialization/end timer trigger 5. 10100 _B to 11111 _B : Setting prohibited.
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	PITMTRG4	Peak-Hold Initialization/End Flag Timer Trigger 4 00000 _B : Uses compare A0/B0 of the Timer for the PH initialization/end timer trigger 4: value after a reset. 00001 _B : Uses compare A1/B1 of the Timer for the PH initialization/end timer trigger 4 00010 _B : Uses compare A2/B2 of the Timer for the PH initialization/end timer trigger 4. : 10011 _B : Uses compare A19/B19 of the Timer for the PH initialization/end timer trigger 4. 10100 _B to 11111 _B : Setting prohibited.

37.3.25 DFEjMITRG — Accumulation/Decimation Initialization/Prohibition Timer Trigger Select Register

This register selects which of the 20 timer trigger compare interfaces--compare signals A0/B0 to A19/B19 in the timer--to use for accumulation/decimation initialization/prohibition timer triggers 0 to 3.

Timer triggers 4 and 5 are set using accumulation/decimation initialization/prohibition timer trigger select register 1 (DFEjMITRG1).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	MITMTRG3						—	—	—	MITMTRG2				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	—	MITMTRG1						—	—	—	MITMTRG0				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	

Table 37.35 DFEjMITRG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
28 to 24	MITMTRG3	Accumulation/Decimation Initialization/Prohibition Flag Timer Trigger 3 0000 _B : Uses compare A0/B0 of the Timer for timer trigger 3. Value after a reset. 0000 _B : Uses compare A1/B1 of the Timer for timer trigger 3. 0001 _B : Uses compare A2/B2 of the Timer for timer trigger 3. 0001 _B : Uses compare A3/B3 of the Timer for timer trigger 3. 0010 _B : Uses compare A4/B4 of the Timer for timer trigger 3. 0010 _B : Uses compare A5/B5 of the Timer for timer trigger 3. 0011 _B : Uses compare A6/B6 of the Timer for timer trigger 3. 0011 _B : Uses compare A7/B7 of the Timer for timer trigger 3. 0100 _B : Uses compare A8/B8 of the Timer for timer trigger 3. 0100 _B : Uses compare A9/B9 of the Timer for timer trigger 3. 0101 _B : Uses compare A10/B10 of the Timer for timer trigger 3. 0101 _B : Uses compare A11/B11 of the Timer for timer trigger 3. 0110 _B : Uses compare A12/B12 of the Timer for timer trigger 3. 0110 _B : Uses compare A13/B13 of the Timer for timer trigger 3. 0111 _B : Uses compare A14/B14 of the Timer for timer trigger 3. 0111 _B : Uses compare A15/B15 of the Timer for timer trigger 3. 1000 _B : Uses compare A16/B16 of the Timer for timer trigger 3. 1000 _B : Uses compare A17/B17 of the Timer for timer trigger 3. 1001 _B : Uses compare A18/B18 of the Timer for timer trigger 3. 1001 _B : Uses compare A19/B19 of the Timer for timer trigger 3. 1010 _B to 1111 _B : Setting prohibited
23 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 37.35 DFEjMITRG Register Contents (2/2)

Bit Position	Bit Name	Function
20 to 16	MITMTRG2	Accumulation/Decimation Initialization/Prohibition Flag Timer Trigger 2 00000 _B : Uses compare A0/B0 of the Timer for timer trigger 2: Value after a reset. 00001 _B : Uses compare A1/B1 of the Timer for timer trigger 2. 00010 _B : Uses compare A2/B2 of the Timer for timer trigger 2. : 10011 _B : Uses compare A19/B19 of the Timer for timer trigger 2. 10100 _B to 11111 _B : Setting prohibited
15 to 13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	MITMTRG1	Accumulation/Decimation Initialization/Prohibition Flag Timer Trigger 1 00000 _B : Uses compare A0/B0 of the Timer for timer trigger 1: Value after a reset. 00001 _B : Uses compare A1/B1 of the Timer for timer trigger 1. 00010 _B : Uses compare A2/B2 of the Timer for timer trigger 1. : 10011 _B : Uses compare A19/B19 of the Timer for timer trigger 1. 10100 _B to 11111 _B : Setting prohibited
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	MITMTRG0	Accumulation/Decimation Initialization/Prohibition Flag Timer Trigger 0 00000 _B : Uses compare A0/B0 of the Timer for timer trigger 0: Value after a reset. 00001 _B : Uses compare A1/B1 of the Timer for timer trigger 0. 00010 _B : Uses compare A2/B2 of the Timer for timer trigger 0. : 10011 _B : Uses compare A19/B19 of the Timer for timer trigger 0. 10100 _B to 11111 _B : Setting prohibited

37.3.26 DFEjMITRG1 — Accumulation/Decimation Initialization/Prohibition Timer Trigger Select Register 1

This register selects which of the 20 timer trigger compare interfaces--compare signals A0/B0 to A19/B19 in the timer--to use for accumulation/decimation initialization/prohibition timer triggers 4 to 5.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	MITMTRG5					—	—	—	MITMTRG4				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 37.36 DFEjMITRG1 Register Contents

Bit Position	Bit Name	Function
31 to 13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	MITMTRG5	Accumulation/Decimation Initialization/Prohibition Timer Trigger 5 00000 _B : Uses compare A0/B0 of the Timer for timer trigger 5: Value after a reset. 00001 _B : Uses compare A1/B1 of the Timer for timer trigger 5 00010 _B : Uses compare A2/B2 of the Timer for timer trigger 5 : 10011 _B : Uses compare A19/B19 of the Timer for timer trigger 5. 10100 _B to 11111 _B : Setting prohibited.
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	MITMTRG4	Accumulation/Decimation Initialization/Prohibition Timer Trigger 4 00000 _B : Uses compare A0/B0 of the Timer for timer trigger 4: Value after a reset. 00001 _B : Uses compare A1/B1 of the Timer for timer trigger 4 00010 _B : Uses compare A2/B2 of the Timer for timer trigger 4 : 10011 _B : Uses compare A19/B19 of the Timer for timer trigger 4. 10100 _B to 11111 _B : Setting prohibited.

37.3.27 DFEjFITRG — Filter Initialization Timer Trigger Select Register

This register selects which of the 20 timer trigger compare interfaces--compare signals A0/B0 to A19/B19 in the timer--to use for filter initialization timer triggers 0 to 3.

Timer triggers 4 and 5 are set using the filter initialization timer trigger select register 1 (DFEjFITRG1).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	FITMTRG3						—	—	—	FITMTRG2				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	—	FITMTRG1						—	—	—	FITMTRG0				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	

Table 37.37 DFEjFITRG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
28 to 24	FITMTRG3	Filter Initialization Flag Timer Trigger 3 0000 _B : Uses compare A0 of the Timer for the filter initialization flag timer trigger 3: value after a reset. 0000 _{1B} : Uses compare A1 of the Timer for the filter initialization flag timer trigger 3. 0001 _{0B} : Uses compare A2 of the Timer for the filter initialization flag timer trigger 3. 0001 _{1B} : Uses compare A3 of the Timer for the filter initialization flag timer trigger 3. 0010 _{0B} : Uses compare A4 of the Timer for the filter initialization flag timer trigger 3. 0010 _{1B} : Uses compare A5 of the Timer for the filter initialization flag timer trigger 3. 0011 _{0B} : Uses compare A6 of the Timer for the filter initialization flag timer trigger 3. 0011 _{1B} : Uses compare A7 of the Timer for the filter initialization flag timer trigger 3. 0100 _{0B} : Uses compare A8 of the Timer for the filter initialization flag timer trigger 3. 0100 _{1B} : Uses compare A9 of the Timer for the filter initialization flag timer trigger 3. 0101 _{0B} : Uses compare A10 of the Timer for the filter initialization flag timer trigger 3. 0101 _{1B} : Uses compare A11 of the Timer for the filter initialization flag timer trigger 3. 0110 _{0B} : Uses compare A12 of the Timer for the filter initialization flag timer trigger 3. 0110 _{1B} : Uses compare A13 of the Timer for the filter initialization flag timer trigger 3. 0111 _{0B} : Uses compare A14 of the Timer for the filter initialization flag timer trigger 3. 0111 _{1B} : Uses compare A15 of the Timer for the filter initialization flag timer trigger 3. 1000 _{0B} : Uses compare A16 of the Timer for the filter initialization flag timer trigger 3. 1000 _{1B} : Uses compare A17 of the Timer for the filter initialization flag timer trigger 3. 1001 _{0B} : Uses compare A18 of the Timer for the filter initialization flag timer trigger 3. 1001 _{1B} : Uses compare A19 of the Timer for the filter initialization flag timer trigger 3. 1010 _{0B} to 1111 _{1B} : Setting prohibited
23 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 37.37 DFEjFITRGR Register Contents (2/2)

Bit Position	Bit Name	Function
20 to 16	FITMTRG2	Filter Initialization Flag Timer Trigger 2 00000 _B : Uses compare A0 of the Timer for the filter initialization flag timer trigger 2: Value after a reset. 00001 _B : Uses compare A1 of the Timer for the filter initialization flag timer trigger 2. 00010 _B : Uses compare A2 of the Timer for the filter initialization flag timer trigger 2. : 10011 _B : Uses compare A19 of the Timer for the filter initialization flag timer trigger 2. 10100 _B to 11111 _B : Setting prohibited
15 to 13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	FITMTRG1	Filter Initialization Flag Timer Trigger 1 00000 _B : Uses compare A0 of the Timer for the filter initialization flag timer trigger 1: Value after a reset. 00001 _B : Uses compare A1 of the Timer for the filter initialization flag timer trigger 1. 00010 _B : Uses compare A2 of the Timer for the filter initialization flag timer trigger 1. : 10011 _B : Uses compare A19 of the Timer for the filter initialization flag timer trigger 1. 10100 _B to 11111 _B : Setting prohibited
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	FITMTRG0	Filter Initialization Flag Timer Trigger 0 00000 _B : Uses compare A0 of the Timer for the filter initialization flag timer trigger 0: Value after a reset. 00001 _B : Uses compare A1 of the Timer for the filter initialization flag timer trigger 0. 00010 _B : Uses compare A2 of the Timer for the filter initialization flag timer trigger 0. : 10011 _B : Uses compare A19 of the Timer for the filter initialization flag timer trigger 0. 10100 _B to 11111 _B : Setting prohibited

37.3.28 DFEjFITRG1— Filter Initialization Timer Trigger Select Register 1

This register selects which of the 20 timer trigger compare interfaces--compare signals A0/B0 to A19/B19 in the timer--to use for filter initialization timer triggers 4 to 5.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FITMTRG5				—	—	—	FITMTRG4					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 37.38 DFEjFITRG1 Register Contents

Bit Position	Bit Name	Function
31 to 13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	FITMTRG5	Filter Initialization Timer Trigger 5 0000 _B : Uses compare A0 of the Timer for the filter initialization flag timer trigger 5: Value after a reset. 00001 _B : Uses compare A1 of the Timer for the filter initialization flag timer trigger 5 00010 _B : Uses compare A2 of the Timer for the filter initialization flag timer trigger 5 : 10011 _B : Uses compare A19 of the Timer for the filter initialization flag timer trigger 5. 10100 _B to 11111 _B : Setting prohibited.
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	FITMTRG4	Filter Initialization Timer Trigger 4 00000 _B : Uses compare A0 of the Timer for the filter initialization flag timer trigger 4: Value after a reset. 00001 _B : Uses compare A1 of the Timer for the filter initialization flag timer trigger 4 00010 _B : Uses compare A2 of the Timer for the filter initialization flag timer trigger 4 : 10011 _B : Uses compare A19 of the Timer for the filter initialization flag timer trigger 4. 10100 _B to 11111 _B : Setting prohibited.

37.3.29 DFEjPHUPDCn — Peak-Hold Update Notification Setting Register n (n = 0 to 3)

These registers are used to set a peak-hold result register update notification during peak-hold processing. One of the channels 0 to 11 for DFE0, channels 0 to 3 for DFE1, or PH2/PH3 processing can be independently selected with DFEjPHUPDC0-3. The set update notification is output to the Timer.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	PHUPDCH				—	PH23SL	PH23E	OEPH UPD
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 37.39 DFEjPHUPDCn Register Contents

Bit Position	Bit Name	Function
7 to 4	PHUPDCH	<p>Peak-Hold Result Register Update Notification Channel Select</p> <p>0_H: Notify that the peak-hold result register has been updated on channel 0: Value after a reset.</p> <p>1_H: Notify that the peak-hold result register has been updated on channel 1.</p> <p>2_H: Notify that the peak-hold result register has been updated on channel 2.</p> <p>:</p> <p>9_H: Notify that the peak-hold result register has been updated on channel 9.</p> <p>A_H: Notify that the peak-hold result register has been updated on channel 10.</p> <p>B_H: Notify that the peak-hold result register has been updated on channel 11.</p> <p>C_H to F_H: Setting prohibited</p> <p>Note: DFE1 is prohibited from 4_H to F_H.</p>
3	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>
2	PH23SL	<p>PH2/PH3 Result Register Update Notification Select</p> <p>0: Value after a reset Notify when the PH2 result register has been updated.</p> <p>1: Notify when the PH3 result register has been updated.</p> <p>Note: This bit is enabled when the PH23 function is used. This bit is disabled in other cases.</p>
1	PH23E	<p>PH2/PH3 Result Register Update Notification Select Enable</p> <p>0: Value after a reset It is not a notification target that PH2/PH3 result register has been updated. It is a notification target that peak-hold result register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1) has been updated.</p> <p>1: It is a notification target that PH2/PH3 result register has been updated. It is not a notification target that peak-hold result register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1) has been updated.</p> <p>Note: This bit is enabled when the PH23 function is used. This bit is disabled in other cases.</p>
0	OEPHUPD	<p>Peak-Hold Result Register Update Notification Enable</p> <p>0_B: Notification of peak-hold result register update in the selected channel is disabled: Value after a reset.</p> <p>1_B: Notification of peak-hold result register update in the selected channel is enabled.</p>

37.3.30 DFEjHOLCHn — Intermediate Value Output Register L n (n = even channels)

While FIR operation is selected, these registers hold intermediate values for digital filter processing for the corresponding channels.

The output intermediate values are the results of calculations prior to saturation processing.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	HO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.40 DFEjHOLCHn Register Contents

Bit Position	Bit Name	Function
31 to 0	HO	Output for Intermediate Values Intermediate values in calculation for digital filter processing (in the output of FIR results, values output prior to saturation processing)

The format of the values is as shown below.

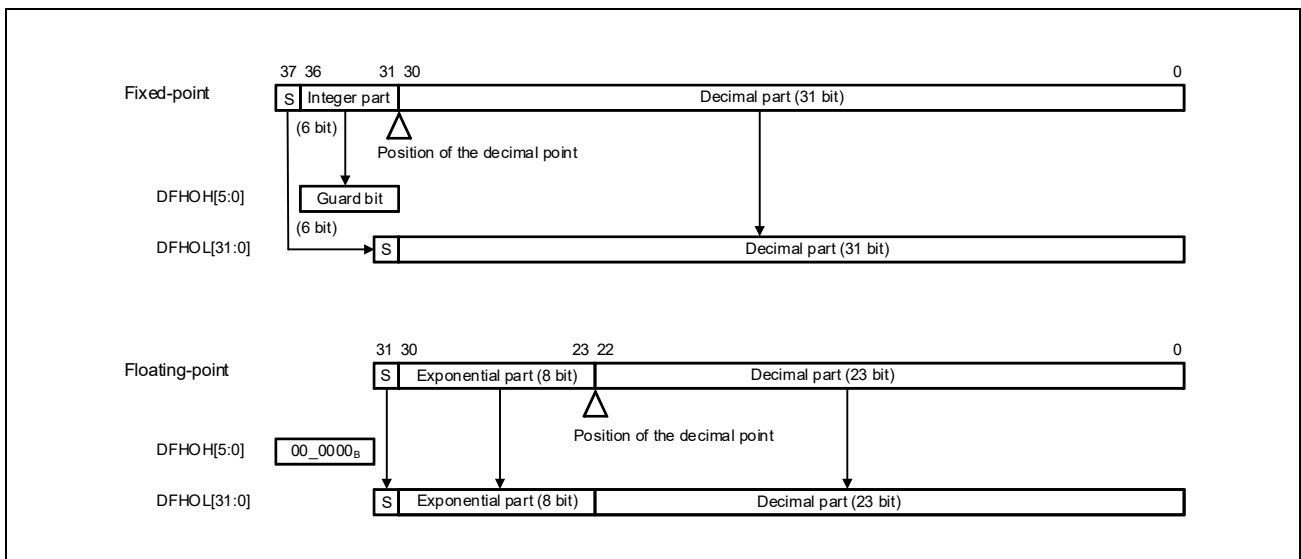


Figure 37.8 Format of Intermediate Value Output Register Values

37.3.31 DFEjHOHCHn — Intermediate Value Output Register H n (n = even channels)

While FIR operation is selected, these registers hold guard bits for intermediate values in processing of the digital filter for the value storage registers of the corresponding channels. The output intermediate values are the results of calculations prior to saturation processing.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	GURD					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.41 DFEjHOHCHn Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	GURD	Output Value Output Guard When processing is fixed-point These bits hold the integer portions (bits [36:31]) of intermediate values in processing for calculations by the digital filter. When conversion to floating point is being executed, 000000 _B is stored. For the format of the values, see Figure 37.8, Format of Intermediate Value Output Register Values.

37.3.32 DFEjHOLMCHn — Intermediate Value Output Mirror Register L n (n = even channels)

Although they are allocated to different locations, these registers are the same as the intermediate value output registers L (DFEjHOLCHn, n = even channels).

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	HO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.42 DFEjHOLMCHn Register Contents

Bit Position	Bit Name	Function
31 to 0	HO	Output for Intermediate Values Intermediate values in calculation for digital filter processing (in the output of FIR results, values output prior to saturation processing) For the specifications of the intermediate values output registers L, see Section 37.3.30, DFEjHOLCHn — Intermediate Value Output Register L n (n = even channels) .

37.3.33 DFEjGAINCHn — IIR Filter Gain Setting Register n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1)

This register is used to set the gain coefficient data GAIN2 and GAIN3 which are used when IIR with a gain adjustment function is selected.

Value after reset: 4000 4000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAIN3															
Value after reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAIN2															
Value after reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.43 DFEjGAINCHn Register Contents

Bit Position	Bit Name	Function
31 to 16	GAIN3	Gain 3 Coefficient Data GAIN3 setting bits are used for IIR secondary biquad 3-stage with a gain adjustment function.
15 to 0	GAIN2	Gain 2 Coefficient Data GAIN2 setting bits are used for IIR secondary biquad 2/3-stage with a gain adjustment function.

The data format of the IIR filter gain setting registers is as follows.

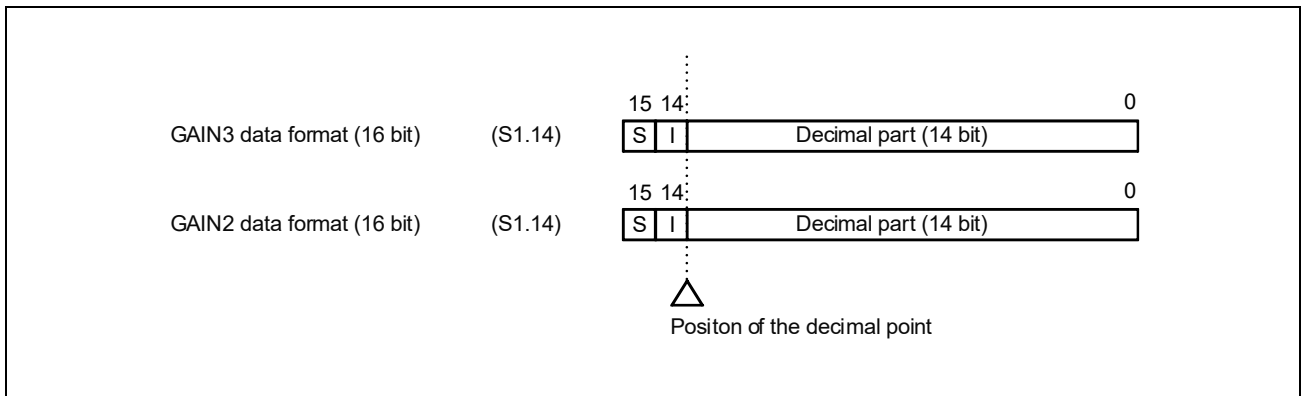


Figure 37.9 Data Format of IIR Filter Gain Setting Registers

37.3.34 DFEjCPOFSTn — Comparison Offset Value Setting Register n (n = 0 to 2)

This register is used to set the comparison offset value α .

Value after reset: 0000 0000_H

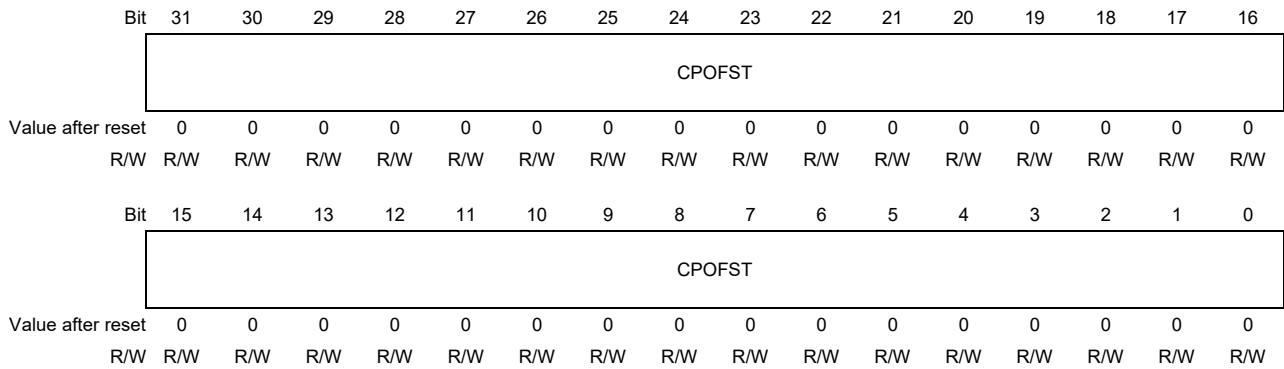


Table 37.44 DFEjCPOFSTn Register Contents

Bit Position	Bit Name	Function
31 to 0	CPOFST	Comparison Offset Value*1 These bits set the comparison offset value α .

Note 1. If the Comparison offset value is used, set it in a data format that is the same as the filter operation results. For details on the format of filter operation results, see **Section 37.5.8**.

When the comparison function is used, assign a negative number (2's complement) to the comparison offset value to make the comparison value equal to [Peak-Hold Result Register (DFEjPHCHn) – Comparison Offset Value Setting Register (DFEjCPOFSTn)].

The data format of the Comparison Offset Value Setting Register is as follows.

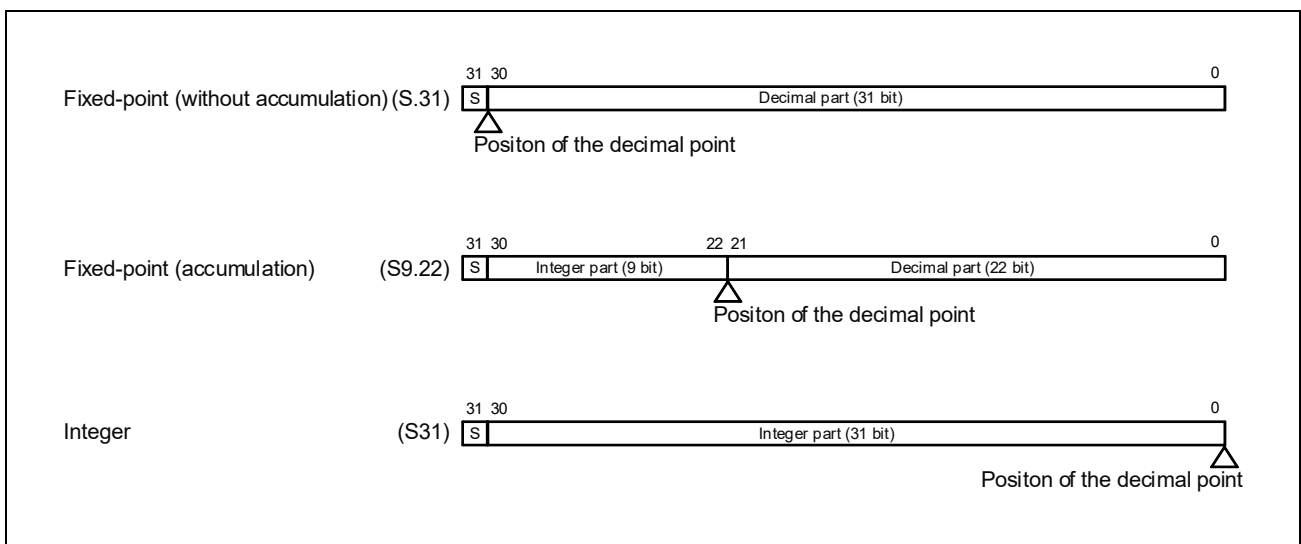


Figure 37.10 The Data Format of Comparison Offset Value Setting Register

37.3.35 DFEjPH23CCTL0 — Peak-Hold 23 Common Control Register 0

This register is used when the PH23 function is selected. The settings common to PH23 functions are specified using this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CHS			PFMT	—	—	PEN	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Note: When the target channel of the peak-hold 23 is being executed (DFEjPH23CCTL0.PEN = 1), it is prohibited to write to any bit other than the PEN bit.

Table 37.45 DFEjPH23CCTL0 Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7 to 4	CHS	Peak-Hold 23 Function Target Channel Select 0 _H : Value after a reset Channel 0 is the target of the peak-hold-23 function when the peak-hold processing is executed. 1 _H : Channel 1 is the target of the peak-hold-23 function when the peak-hold processing is executed. 2 _H : Channel 2 is the target of the peak-hold-23 function when the peak-hold processing is executed. ... B _H : Channel 11 is the target of the peak-hold-23 function when the peak-hold processing is executed. C _H to F _H : Setting prohibited Note: DFE1 is prohibited from 4 _H to F _H .
3	PFMT	Peak-Hold 23 Result Register Floating-Point Conversion 0: Value after a reset Floating-point conversion for neither DFPH20 nor DFPH30 is executed. 1: Floating-point conversion for both DFPH20 and DFPH30 is executed. Fixed-point format (S.31/S9.22) is converted to floating-point (IEEE754). This bit must not be set to 1 if the setting of DFEjCTLACHn.FMT = 1 (integer mode) while FIR operation is selected.
2, 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	PEN	Peak-Hold 23 Function Enable 0: Value after a reset The peak-hold 23 function is disabled. The peak-hold 23 function is not used. 1: The peak-hold 23 function is enabled. The peak-hold 23 function can be used.

Note: When 0 is written to DFEjPH23CCTL0.PEN, DFEjSTCHn.PH23ST of the channel set by DFEjPH23CCTL0.CHS is set to 0000_B.

37.3.36 DFEjPH2CTL0 — Peak-Hold 2 Control Register 0

This register is used when the peak-hold 23 function is selected. Any setting related to PH2 processing is specified using this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OFSL	—		PH2SLB2		PHPS		CN2SLB2		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: When the target channel of the peak-hold 23 is executed (DFEjPH23CCTL0.PEN = 1), it is prohibited to write to this register.

Table 37.46 DFEjPH2CTL0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 10	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
9, 8	OFSL	Peak-Hold 2 Processing Comparison Offset Value α Select 00 _B : Value after a reset DFEjCPOFST0 register value is selected for the comparison offset value α . 01 _B : DFEjCPOFST1 register value is selected for the comparison offset value α . 10 _B : DFEjCPOFST2 register value is selected for the comparison offset value α . 11 _B : Setting prohibited
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 4	PH2SLB2	Peak-Hold 2 Initial Value Register Select 000 _B : Value after a reset. DFEjPHIA register value is selected for the Peak-Hold 2 processing initial value. 001 _B : DFEjPHIB register value is selected for the Peak-Hold 2 processing initial value. 010 _B : DFEjPHIC register value is selected for the Peak-Hold 2 processing initial value. 011 _B : DFEjPHID register value is selected for the Peak-Hold 2 processing initial value. 100 _B to 111 _B : Setting prohibited Note: When using the peak-hold-23 function, set the peak-hold 2 initial value as follows: 1. When upper-limit peak detection (PHPS = 0) Select the negative minimum value (8000 0000 _H). 2. When lower-limit peak detection (PHPS = 1) Select the positive maximum value (7FFF FFFF _H).
3	PHPS	Peak-Hold 2 Peak Type Select 0: The peak-hold 2 processing detects upper-limit peaks (value after a reset). 1: The peak-hold 2 processing detects lower-limit peaks.

Table 37.46 DFEjPH2CTL0 Register Contents (2/2)

Bit Position	Bit Name	Function
2 to 0	CN2SLB2	<p>Peak-Hold 2 Comparison Calculation Select</p> <p>000_B: Value after a reset "Equal to (==)" is selected. Calculation of "equal to (==)" for the comparison the target data*¹ and the peak-hold circuit input data is performed.</p> <p>001_B: "Less than or equal to (\leq)" is selected. Calculation of "less than or equal to (\leq)" for the comparison the target data*¹ and the peak-hold circuit input data is performed.</p> <p>010_B: "More than or equal to (\geq)" is selected. Calculation of "more than or equal to (\geq)" for the comparison the target data*¹ and the peak-hold circuit input data is performed.</p> <p>011_B: "Less than (<)" is selected. Calculation of "less than (<)" for the comparison the target data*¹ and the peak-hold circuit input data is performed.</p> <p>100_B: "More than (>)" is selected. Calculation of "more than (>)" for the comparison the target data*¹ and the peak-hold circuit input data is performed.</p> <p>101_B to 111_B: Setting prohibited</p> <p>Note 1. The target data is "Peak-Hold 2 Result Register 0(DFEjPH20) + α(Comparison Offset Value)". α is Comparison Offset Value selected by DFEjPH2CTL0.OFSL.</p>

37.3.37 DFEjPH3CTL0 — Peak-Hold 3 Control Register 0

This register is used when the peak-hold 23 function is selected. Any setting related to PH3 processing is specified using this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OFSL	—		PH3SLB2		PHPS		CN3SLB2		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: When the target channel of the peak-hold 23 is being executed (DFEjPH23CCTL0.PEN = 1), it is prohibited to write to this register.

Table 37.47 DFEjPH3CTL0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 10	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
9, 8	OFSL	Peak-Hold 3 Processing Comparison Offset Value α Select 00 _B : Value after a reset DFEjCPOFST0 register value is selected for the comparison offset value α . 01 _B : DFEjCPOFST1 register value is selected for the comparison offset value α . 10 _B : DFEjCPOFST2 register value is selected for the comparison offset value α . 11 _B : Setting prohibited
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 4	PH3SLB2	Peak-Hold 3 Initial Value Register Select 000 _B : DFEjPHIA register value is selected for the Peak-Hold 3 processing initial value. Value after a reset. 001 _B : DFEjPHIB register value is selected for the Peak-Hold 3 processing initial value. 010 _B : DFEjPHIC register value is selected for the Peak-Hold 3 processing initial value. 011 _B : DFEjPHID register value is selected for the Peak-Hold 3 processing initial value. 100 _B to 111 _B : Setting prohibited Note: When using the peak-hold-23 function, set the peak-hold 3 initial value as follows: 1. When upper-limit peak detection (PHPS = 0) Select the negative minimum value (8000 0000 _H). 2. When lower-limit peak detection (PHPS = 1) Select the positive maximum value (7FFF FFFF _H).
3	PHPS	Peak-Hold 3 Peak Type Select 0: The peak-hold 3 processing detects upper-limit peaks (value after a reset) 1: The peak-hold 3 processing detects lower-limit peaks.

Table 37.47 DFEjPH3CTL0 Register Contents (2/2)

Bit Position	Bit Name	Function
2 to 0	CN3SLB2	<p>Peak-Hold 3 Comparison Calculation Select</p> <p>000_B: Value after a reset "Equal to (==)" is selected. Calculation of "equal to (==)" for the comparison the target data*¹ and the peak-hold circuit input data is performed.</p> <p>001_B: "Less than or equal to (\leq)" is selected. Calculation of "less than or equal to (\leq)" for the comparison the target data*¹ and the peak-hold circuit input data is performed.</p> <p>010_B: "More than or equal to (\geq)" is selected. Calculation of "more than or equal to (\geq)" for the comparison the target data*¹ and the peak-hold circuit input data is performed.</p> <p>011_B: "Less than (<)" is selected. Calculation of "less than (<)" for the comparison the target data*¹ and the peak-hold circuit input data is performed.</p> <p>100_B: "More than (>)" is selected. Calculation of "more than (>)" for the comparison the target data*¹ and the peak-hold circuit input data is performed.</p> <p>101_B to 111_B: Setting prohibited</p> <p>Note 1. The target data is "Peak-Hold 3 Result Register 0(DFEjPH30) + α(Comparison Offset Value)". α is Comparison Offset Value selected by DFEjPH3CTL0.OFSL.</p>

37.3.38 DFEjPH20 — Peak-Hold 2 Result Register 0

This register is used to store the peak-hold 2 processing result value.

Value after reset: 8000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PH2															
Value after reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.48 DFEjPH20 Register Contents

Bit Position	Bit Name	Function
31 to 0	PH2	Peak-Hold 2 Result Data These bits show the peak-hold 2 processing calculation results.

The data format for the peak-hold 2/3 result registers is the same as the data format of DFEjPHCHn.

37.3.39 DFEjPH30 — Peak-Hold 3 Result Register 0

This register is used to store the peak-hold 3 processing result value.

Value after reset: 8000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PH3															
Value after reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH3															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.49 DFEjPH30 Register Contents

Bit Position	Bit Name	Function
31 to 0	PH3	Peak-Hold 3 Result Data These bits show the peak-hold 3 processing calculation results.

37.3.40 DFEjPH2IND0 — Peak-Hold 2 Index Register 0

When the peak-hold 23 function is used and the peak-hold 2 result register is updated, this register holds for the peak-hold processing count.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH2 IOW	PH2 IOF	PH2IND													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.50 DFEjPH2IND0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15	PH2IOW	<p>0: Value after a reset PH2IND has not been overwritten: Value after a reset. This flag becomes 0 when 1 is written to DFEjCLRSTCHn.CLRPHIOW.</p> <p>1: PH2IND has been overwritten in peak-hold index updating mode.</p> <p>CAUTION: Enabled when the peak-hold 23 function is used, and disabled in all other cases. Peak-Hold 2 Index Overwrite Flag This flag is valid in peak-hold index updating mode. It is always 0 in peak-hold index retention mode.</p>
14	PH2IOF	<p>0: Value after a reset The number of rounds of peak-hold processing has not overflowed (≤ 16383 times).</p> <ul style="list-style-type: none"> • If the peak-hold initialization flag is raised in peak-hold index updating mode, peak-hold processing clears PH2IOF to 0. • In peak-hold index-retention mode, the peak-hold initialization flag is raised after the peak-hold 2 index register 0 is read, and processing of the first value for peak-hold processing clears PH2IOF to 0. <p>1: The number of rounds of peak hold processing has overflowed (> 16383 times).</p> <p>CAUTION: Enabled when the peak-hold 23 function is used, and disabled in all other cases. Peak-Hold 2 Index Overflow Flag The number of rounds of peak-hold processing is counted from the time of input to the peak-hold initialization flag to the time of input to the peak-hold end flag.</p>

Table 37.50 DFEjPH2IND0 Register Contents (2/2)

Bit Position	Bit Name	Function
13 to 0	PH2IND	<p>Peak-Hold 2 Index</p> <p>0000_H: Value after a reset</p> <ul style="list-style-type: none"> • In peak-hold index-updating mode: Indicates that the first condition for peak-hold updating is not satisfied when the peak-hold initialization flag has been raised. • In peak-hold index-retention mode: Indicates that the first condition for peak-hold updating is not satisfied when the peak-hold initialization flag has been raised after the peak-hold 2 index register 0 has been read. <p>0001_H: Indicates that the value for the first input to the peak-hold circuit has been written to the peak-hold 2 results register</p> <p>0002_H: Indicates that the value for the second input to the peak-hold circuit has been written to the peak-hold 2 results register</p> <p>.....</p> <p>3FFF_H: Indicates that the value for the 16383rd input to the peak-hold circuit has been written to the peak-hold 2 results register.</p> <p>CAUTION: When the peak-hold 23 function is used and the condition for updating peak-hold has been satisfied and the value is stored in the peak-hold 2 results register, the PH2IND register captures the number of values to which peak-hold 2 processing has been applied (the value of the peak-hold counter within the digital filter). After the value of PH2IOF becomes "1" when the number of rounds of peak-hold processing reaches and exceeds 16384, operation of the peak-hold counter continues and PH2IND continues to be updated.</p>

37.3.41 DFEjPH3IND0 — Peak-Hold 3 Index Register 0

When the peak-hold 23 function is used and the peak-hold 3 result register is updated, this register holds for the peak-hold processing count.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH3 IOW	PH3 IOF	PH3IND													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.51 DFEjPH3IND0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15	PH3IOW	0: Value after a reset PH3IND has not been overwritten: Value after a reset. This flag becomes 0 when 1 is written to DFEjCLRSTCHn.CLRPHIOW. 1: PH3IND has been overwritten in peak-hold index updating mode. CAUTION: Enabled when the peak-hold 23 function is used, and disabled all other cases. This flag is valid in peak-hold index updating mode. It is always 0 in peak-hold index retention mode.
14	PH3IOF	0: Value after a reset The number of rounds of peak-hold processing has not overflowed (≤ 16383 times). <ul style="list-style-type: none"> If the peak-hold initialization flag is raised in peak-hold index updating mode, peak-hold processing clears PH3IOF to 0. In peak-hold index-retention mode, the peak-hold initialization flag is raised after the peak-hold 3 index register 0 is read, and processing of the first value for peak-hold processing clears PH3IOF to 0. 1: The number of rounds of peak hold processing has overflowed (> 16383 times). CAUTION: Enabled when the peak-hold 23 function is used, and disabled all other cases. Peak-Hold 3 Index Overflow Flag The number of rounds of peak-hold processing is counted from the time of input to the peak-hold initialization flag to the time of input to the peak-hold end flag.

Table 37.51 DFEjPH3IND0 Register Contents (2/2)

Bit Position	Bit Name	Function
13 to 0	PH3IND	<p>Peak-Hold 3 Index</p> <p>0000_H: Value after a reset</p> <ul style="list-style-type: none"> • In peak-hold index-updating mode: Indicates that the first condition for peak-hold updating is not satisfied when the peak-hold initialization flag has been raised. • In peak-hold index-retention mode: Indicates that the first condition for peak-hold updating is not satisfied when the peak-hold initialization flag has been raised after the peak-hold 3 index register 0 has been read. <p>0001_H: Indicates that the value for the first input to the peak-hold circuit has been written to the peak-hold 3 results register</p> <p>0002_H: Indicates that the value for the second input to the peak-hold circuit has been written to the peak-hold 3 results register</p> <p>...: ...</p> <p>3FFF_H: Indicates that the value for the 16383th input to the peak-hold circuit has been written to the peak-hold 3 results register.</p> <p>CAUTION: When the peak-hold 23 function is used and the condition for updating peak-hold has been satisfied and the value is stored in the peak-hold 3 results register, the PH3IND register captures the number of values to which peak-hold 3 processing has been applied (the value of the peak-hold counter within the digital filter). After the value of PH3IOF becomes "1" when the number of rounds of peak-hold processing reaches and exceeds 16384, operation of the peak-hold counter continues and PH3IND continues to be updated.</p>

37.3.42 DFEjSUBCTLCHn — Subtraction Control Register n (n = 0 to 2)

This register is used when the subtraction function is selected. Set each subtraction channel.

Assign the same data format to both the subtrahend and minuend channels.

It is prohibited that the subtrahend and minuend channels have different data formats.

$$[\text{Subtraction output data}] = \text{Filter results of minuend channel} - \text{Filter results of subtrahend channel}$$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	CATAG					CAEN		—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	—	—	MINCH			SUBCH				SFMT	SIEE	SIEO	SEN		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Note: When the target channel of the subtrahend or minuend is being executed (DFEjCTLACHn.EN = 1), it is prohibited to write any bit other than the DFEjSUBCTLCHn.SEN.

Table 37.52 DFEjSUBCTLCHn Register Contents (1/2)

Bit Position	Bit Name	Function
31	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
30 to 26	CATAG	Tag Value for Reentering Subtraction Result (cascading tag) 00 _H to 1F _H : Subtraction result data of the subtraction circuit is reentered (cascaded) during execution of cascade. A channel tag value (DFEjCTLACHn.TAG) is specified.
25, 24	CAEN	Cascade Enable 00 _B : Value after a reset Cascade is disabled. Subtraction result data of the subtraction circuit is written to the output data register. 01 _B : Setting prohibited 10 _B : Cascade is enabled. Subtraction result data of the subtraction circuit is written to the output data register together with cascade. 11 _B : Cascade is enabled. Subtraction result data of the subtraction circuit is not written to the output data register together with cascade.
23 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 8	MINCH	Minuend Channel Select 0 _H : Channel 0 is set to the minuend channel. Value after a reset 1 _H : Channel 1 is set to the minuend channel. 2 _H : Channel 2 is set to the minuend channel. ... B _H : Channel 11 is set to the minuend channel. C _H to F _H : Setting prohibited Note: DFE1 is prohibited from 4 _H to F _H . CAUTION: It is prohibited to set the same channel to both MINCH and SUBCH.

Table 37.52 DFEjSUBCTLCHn Register Contents (2/2)

Bit Position	Bit Name	Function
7 to 4	SUBCH	<p>Subtrahend Channel Select</p> <p>0_H: Channel 0 is set to the subtrahend channel. Value after a reset</p> <p>1_H: Channel 1 is set to the subtrahend channel.</p> <p>2_H: Channel 2 is set to the subtrahend channel.</p> <p>...</p> <p>B_H: Channel 11 is set to the subtrahend channel.</p> <p>C_H to 000F_H: Setting prohibited</p> <p>Note: DFE1 is prohibited from 4_H to F_H.</p> <p>CAUTION: It is prohibited to set the same channel to both MINCH and SUBCH.</p>
3	SFMT	<p>Subtraction Result Register n Floating-Point Conversion</p> <p>0: Floating-point conversion for DFEjSUBDOCHn is not executed. Value after a reset</p> <p>1: Floating-point conversion for DFEjSUBDOCHn is executed.</p> <p>Floating-point conversion for both DFPH20 and DFPH30 is executed.</p> <p>Fixed-point format (S.31/S9.22) is converted to floating-point (IEEE754).</p> <p>This bit must not be set to 1 if the setting of DFEjCTLACHn.FMT = 1 (integer mode) while FIR operation is selected for both the subtrahend and minuend channels.</p>
2	SIEE	<p>Subtraction Error Interrupt Request Enable</p> <p>0: A subtraction error interrupt request is disabled. Value after a reset</p> <p>1: A subtraction error interrupt request is enabled.</p>
1	SIEO	<p>Subtraction Output Data Interrupt Request Enable</p> <p>0: A subtraction output data interrupt is disabled when the result of subtraction is written to the subtraction output data register. Value after a reset</p> <p>1: A subtraction error interrupt request is enabled.</p> <p>A subtraction output data interrupt is enabled when the result of subtraction is written to the subtraction output data register.</p>
0	SEN	<p>Subtraction Channel Enable</p> <p>0: Subtraction Channel disabled.</p> <p>No subtraction processing is executed in this channel. Value after a reset</p> <p>1: Subtraction Channel enabled.</p> <p>The subtraction processing is executed in this channel.</p>

The subtraction processing of the target subtraction channel is stopped by writing 0 to DFEjSUBCTLCHn.SEN.

37.3.43 DFEjSUBDOCHn — Subtraction Result Register n (n = 0 to 2)

This register stores operation results after the completion of subtraction operations for each subtraction channel.

Value after reset: 0000 0000_H

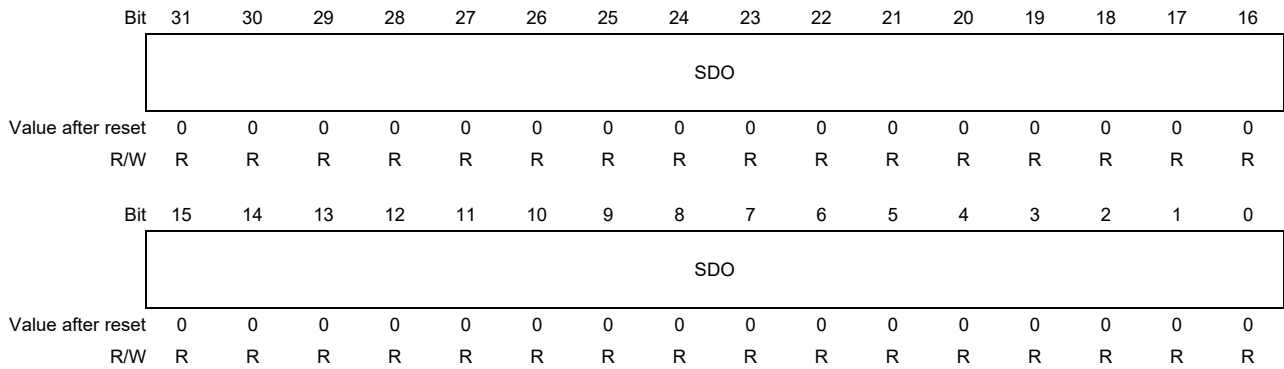


Table 37.53 DFEjSUBDOCHn Register Contents

Bit Position	Bit Name	Function
31 to 0	SDO	Subtraction Output Data Results of subtraction

The data format for the subtraction result data register is shown below.

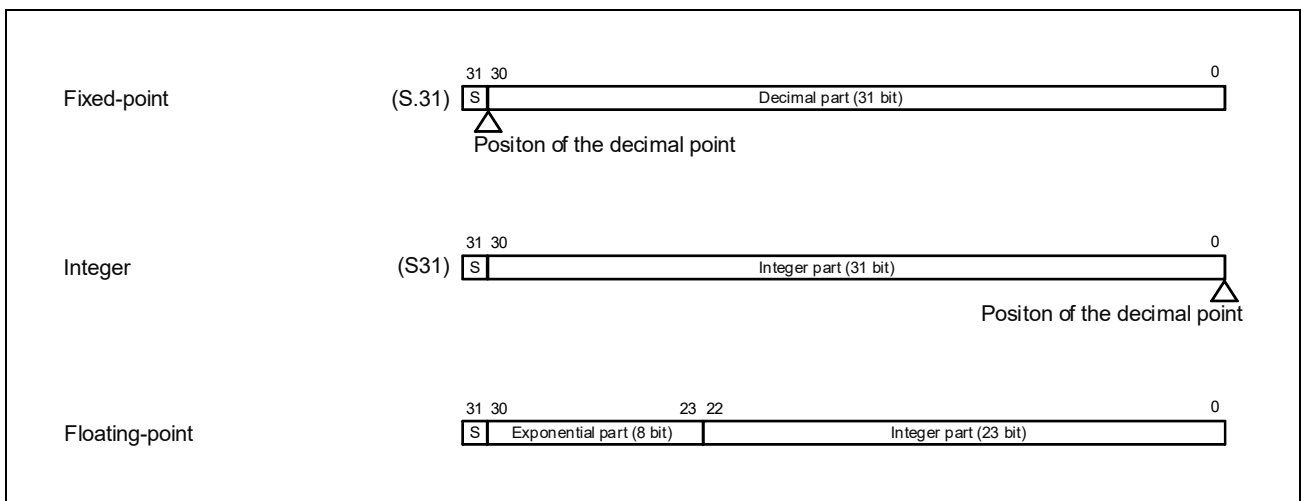


Figure 37.11 The Data Format for the Subtraction Result Data Register

37.3.44 DFEjSUBSTCHn — Subtraction Status Register n (n = 0 to 2)

This register indicates the status of subtraction channels.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SDOEN	MINF	SUBF	SCER	—	—	SGER	SDOOW	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.54 DFEjSUBSTCHn Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	SDOEN	Subtraction Output Data Register Valid 0: Indicates that the value of the subtraction output data register is invalid. While SDOEN = 1, writing 1 to DFEjSUBCLRSTCHn.CLRSDOEN clears SDOEN to 0. While SDOEN = 1, reading the DFEjSUBDOCHn register clears this bit to 0. Value after a reset 1: Indicates that the value of the subtraction output data register is valid. This bit is set to 1 when the DFE stores the subtraction results in the subtraction output data register.
7	MINF	Minuend Channel Data Input Enable 0: Value after a reset Indicates that no filtering result data has been input on the minuend channel. When MINF = 1 and SUBF = 0, the DFE performs subtraction upon completion of the filtering of the subtrahend channel. Storing subtraction result data in the subtraction result register clears MINF to 0. 1: Indicates that filtering result data has been input on the minuend channel. When SUBF = 0, completion of the filtering of the minuend channel sets MINF to 1.
6	SUBF	Subtrahend Channel Data Input Enabling Bit 0: value after a reset Indicates that no filtering result data has been input on the subtrahend channel. When SUBF = 1 and MINF = 0, the DFE performs subtraction upon completion of the filtering of the minuend channel. Storing subtraction result data in the subtraction result register clears SUBF to 0. 1: Indicates that filtering result data has been input on the subtrahend channel. When MINF = 0, completion of the filtering of the subtrahend channel sets SUBF to 1
5	SCER	Subtraction Circuit Cascade Rounding Error 0: No cascade rounding error is present: value after a reset. When SCER = 1, writing 1 to DFEjSUBCLRSTCHn.CLRSCER clears SCER to 0. 1: A cascade rounding error occurred in the subtraction output data. When DFEjSUBCTLCHn.CAEN = 10 _B or 11 _B , this bit is set to 1 if an overflow occurs when the 32-bit subtraction circuit output data is rounded to 16-bit data. When DFEjSUBCTLCHn.CAEN = 00 _B , this bit is set to 1 if the subtraction result is judged as a cascade rounding overflow will be occurred.
4 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 37.54 DFEjSUBSTCHn Register Contents (2/2)

Bit Position	Bit Name	Function
2	SGER	<p>Subtraction Guard Error</p> <p>0: Value after a reset No Subtraction guard error is present. When SGER = 1, writing 1 to DFEjSUBCLRSTCHn.CLRSGER clears SGER to 0.</p> <p>1: A Subtraction guard error is present. This bit is set to 1 if an overflow occurs when the subtraction result is rounded to 32 bits in the filter circuit.</p>
1	SDOOW	<p>Subtraction Result Register Overwrite Error</p> <p>0: Value after a reset No subtraction result register overwrite error is present. When SDOOW = 1, writing 1 to DFEjSUBCLRSTCHn.CLRSDOOW clears SDOOW to 0.</p> <p>1: A subtraction result register overwrite error occurred. When SDOEN = 1, storing subtraction output data in the subtraction result register sets SDOOW to 1.</p>
0	—	<p>Reserved</p> <p>When read, the value after reset is returned. When writing, write the value after reset.</p>

37.3.45 DFEjSUBCLRSTCHn — Subtraction Clear Status Register n (n = 0 to 2)

This register is used to clear the subtraction status register. Writing 1 to clearing bits clears the corresponding DFEjSUBSTCHn bits to 0.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLRS DOEN	—	—	CLRS CER	—	—	CLR SGER	CLRS DOOW	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W*	R	R	R/W*	R	R	R/W*	R/W*	R

Table 37.55 DFEjSUBCLRSTCHn Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	CLRSDOEN	Subtraction Output Data Enable Bit Clear 0: DFEjSUBSTCHn.SDOEN is not cleared: value after a reset. 1: DFEjSUBSTCHn.SDOEN is cleared. When DFEjSUBSTCHn.SDOEN is cleared, this bit is automatically cleared to 0. CAUTION: This bit is always read as 0.
7 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	CLRSCER	Subtraction Circuit Cascade Rounding Error Bit Clear 0: DFEjSUBSTCHn.SCER is not cleared: value after a reset. 1: DFEjSUBSTCHn.SCER is cleared. When DFEjSUBSTCHn.SCER is cleared, this bit is automatically cleared to 0. CAUTION: This bit is always read as 0.
4 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	CLRSGER	Subtraction Guard Error Bit Clear 0: DFEjSUBSTCHn.SGER is not cleared: value after a reset. 1: DFEjSUBSTCHn.SGER is cleared. When DFEjSUBSTCHn.SGER is cleared, this bit is automatically cleared to 0. CAUTION: This bit is always read as 0.
1	CLRSDOOW	Subtraction Output Data Register Overwrite Error Bit Clear 0: DFEjSUBSTCHn.SDOOW is not cleared: value after a reset. 1: DFEjSUBSTCHn.SDOOW is cleared. When DFEjSUBSTCHn.SDOOW is cleared, this bit is automatically cleared to 0. CAUTION: This bit is always read as 0.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

37.3.46 DFEjSUBERMCHn — Subtraction Error Mask Register n (n = 0 to 2)

This register is used to mask errors of the subtraction channel.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	MSK SCER	—	—	MSK SGER	MSK SDOO W	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R

Table 37.56 DFEjSUBERMCHn Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	MSKSCER	Subtraction Circuit Cascade Rounding Error Bit Mask 0: DFEjSUBSTCHn.SCER is not masked: Value after a reset. 1: DFEjSUBSTCHn.SCER is masked. Note: When the subtraction circuit is enabled (DFEjSUBCTLCHn.SEN = 1) and the cascade function is disabled (DFEjSUBCTLCHn.CAEN = 00 _B), set this bit to 1.
4 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	MSKSGER	Subtraction Guard Error Bit Mask 0: DFEjSUBSTCHn.SGER is not masked: value after a reset. 1: DFEjSUBSTCHn.SGER is masked.
1	MSKSDOOW	Subtraction Output Data Register Overwrite Error Bit Mask 0: DFEjSUBSTCHn.SDOOW is not masked: Value after a reset. 1: DFEjSUBSTCHn.SDOOW is masked.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

37.3.47 DFEjSUBTRGCHn — Subtraction Trigger Setting Register n (n = 0 to 2)

This register is used to generate a trigger flag for subtraction channels n, using software and timer triggers. It is prohibited to set both the initialization and ending flags to a software trigger for subtraction processing. If prohibited settings are specified, the integrity of ensuing operations cannot be guaranteed.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SBFE	—	—	—	—	—	—	—	SBE	—	—	—	—	—	—	SBT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R/W	R/W

Table 37.57 DFEjSUBTRGCHn Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
15, 14	SBFE	Subtraction End Flag Trigger Setting 00 _B : Value after a reset Subtraction end flag is not generated. 01 _B : Subtraction end flag is generated by a timer trigger. 10 _B : Subtraction end flag is generated by a software trigger. 11 _B : Setting prohibited CAUTION: It is prohibited to set both the subtraction start flag and the subtraction end flag to a software trigger. (It is prohibited to set DFEjSUBTRGCHn.SBFE = 10 _B and DFEjSUBTRGCHn.SBE = 10 _B). When the subtraction end flag is generated by a timer trigger (DFEjSUBTRGCHn.SBFE = 01 _B), the subtraction end flag is automatically selected from among compare signals B0 to B19 from the timer. Whichever signal corresponds to (has the same number as) the timer trigger (one of compare signals A0 to A19) selected by the subtraction trigger setting register (DFEjSUBTRGCHn.SBT) is selected.
13 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7, 6	SBE	Subtraction Start Flag Trigger Setting 00 _B : Value after a reset Subtraction start flag is not generated. 01 _B : Subtraction start flag is generated by a timer trigger. 10 _B : Subtraction start flag is generated by a software trigger. 11 _B : Setting prohibited
5 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 37.57 DFEjSUBTRGCHn Register Contents (2/2)

Bit Position	Bit Name	Function
1, 0	SBT	<p>Subtraction Start Flag and Subtraction End Flag Timer Trigger Select</p> <p>00_B: Value after a reset Timer trigger 0 is selected for generating the subtraction start flag and subtraction end flag.</p> <p>01_B: Timer trigger 1 is selected for generating the subtraction start flag and subtraction end flag.</p> <p>10_B: Timer trigger 2 is selected for generating the subtraction start flag and subtraction end flag.</p> <p>11_B: Setting prohibited</p> <p>CAUTION: This bit is enabled when the subtraction start flag trigger setting bits (DFEjSUBTRGCHn.SBE) are 01_B. Timer triggers 0 to 2 are selected using DFEjSUBTRG0.SUBTMTRG0 to SUBTMTRG2.</p>

37.3.48 DFEjSUBTRHCHn — Subtraction Trigger History Register n (n = 0 to 2)

This register is used to monitor the execution history of software triggers, timer triggers, and trigger flags related to subtraction processing.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SITS	SETS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.58 DFEjSUBTRHCHn Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SITS	Subtraction Start Flag Trigger History 0: Value after a reset A subtraction start flag trigger has not been generated. When SITS = 1, this bit becomes 0 upon completion of the execution of the subtraction start flag. This bit also becomes 0 if DFEjSUBCTLCHn.SBEN is set to 0 and subtraction processing is not being performed on a subtraction channel n. 1: This bit becomes 1 when the subtraction start flag trigger is generated.
0	SETS	Subtraction End Flag Trigger History 0: Value after a reset A subtraction end flag trigger has not been generated. When SETS = 1, this bit becomes 0 upon completion of the execution of the subtraction end flag. This bit becomes 0 if DFEjSUBCTLCHn.SBEN is set to 0 and subtraction processing is not being performed on a subtraction channel n. 1: This bit becomes 1 when the subtraction end flag trigger is generated.

37.3.49 DFEjSUBTRG0 — Subtraction Start/End Timer Trigger Select Register 0

This register selects which of the 20 timer trigger compare interfaces--compare signals A0/B0 to A19/B19 in the timer--to use for subtraction start/end timer triggers 0 to 2.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	SUBTMTRG2				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SUBTMTRG1				—	—	—	SUBTMTRG0					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 37.59 DFEjSUBTRG0 Register Contents

Bit Position	Bit Name	Function
31 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20 to 16	SUBTMTRG2	Subtraction Start/End Flag Timer Trigger 2 0000 _B : Value after a reset Uses compare A0/B0 of the Timer and the subtraction start/end timer trigger 2. 00001 _B : Uses compare A1/B1 of the Timer and the subtraction start/end timer trigger 2. 00010 _B : Uses compare A2/B2 of the Timer and the subtraction start/end timer trigger 2. ... 10011 _B : Uses compare A19/B19 of the Timer and the subtraction start/end timer trigger 2. 10100 _B to 11111 _B : Setting prohibited.
15 to 13	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	SUBTMTRG1	Subtraction Start/End Flag Timer Trigger 1 0000 _B : Value after a reset Uses compare A0/B0 of the Timer and the subtraction start/end timer trigger 1. 00001 _B : Uses compare A1/B1 of the Timer and the subtraction start/end timer trigger 1. 00010 _B : Uses compare A2/B2 of the Timer and the subtraction start/end timer trigger 1. ... 10011 _B : Uses compare A19/B19 of the Timer and the subtraction start/end timer trigger 1. 10100 _B to 11111 _B : Setting prohibited.
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	SUBTMTRG0	Subtraction Start/End Flag Timer Trigger 0 0000 _B : Value after a reset Uses compare A0/B0 of the Timer and the subtraction start/end timer trigger 0. 00001 _B : Uses compare A1/B1 of the Timer and the subtraction start/end timer trigger 0. 00010 _B : Uses compare A2/B2 of the Timer and the subtraction start/end timer trigger 0. ... 10011 _B : Uses compare A19/B19 of the Timer and the subtraction start/end timer trigger 0. 10100 _B to 11111 _B : Setting prohibited.

37.3.50 DFEjCAPCTLCHn — Capture Control Register n (n = 0 to 2)

This register is used when the capture function is selected for a DFE channel.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CAPCH			—	CIEE	—	CEN	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R	R/W

Note: While the setting of DFEjCAPCTLCHn.CEN is 1, writing to CAPCH and CIEE is prohibited.

Table 37.60 DFEjCAPCTLCHn Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7 to 4	CAPCH	Capture-Target Channel Select 0 _H : Channel 0 is set to the capture channel. Value after a reset 1 _H : Channel 1 is set to the capture channel 2 _H : Channel 2 is set to the capture channel ... B _H : Channel 11 is set to the capture channel C _H to F _H : Setting prohibited. Note: DFE1 is prohibited from 4 _H to F _H .
3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	CIEE	Capture Error Interrupt Request Enable 0: Value after a reset A capture error interrupt request is disabled. 1: A capture error interrupt request is enabled.
1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CEN	Capture Channel Enable 0: Value after a reset Capture channel disabled. No capture processing is executed in this channel. 1: Capture channel enabled. The Capture processing is executed in this channel.

37.3.51 DFEjCAPDOCHn — Capture Result Register n (n = 0 to 2)

This register stores the output data register value on the capture-target channel. If the data is read with floating-point conversion format, set the output data register (DFEjDOCHn) of the capture-target channel to floating-point conversion format.

Value after reset: 0000 0000_H

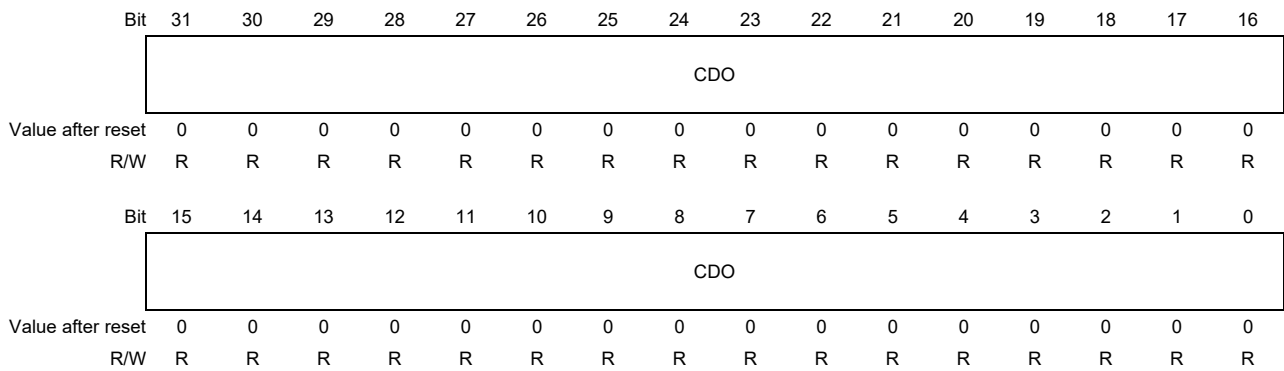


Table 37.61 DFEjCAPDOCHn Register Contents

Bit Position	Bit Name	Function
31 to 0	CDO	Capture Output Data Stores the output data register value on the capture-target channel.

The data format of the capture result data register is shown below.

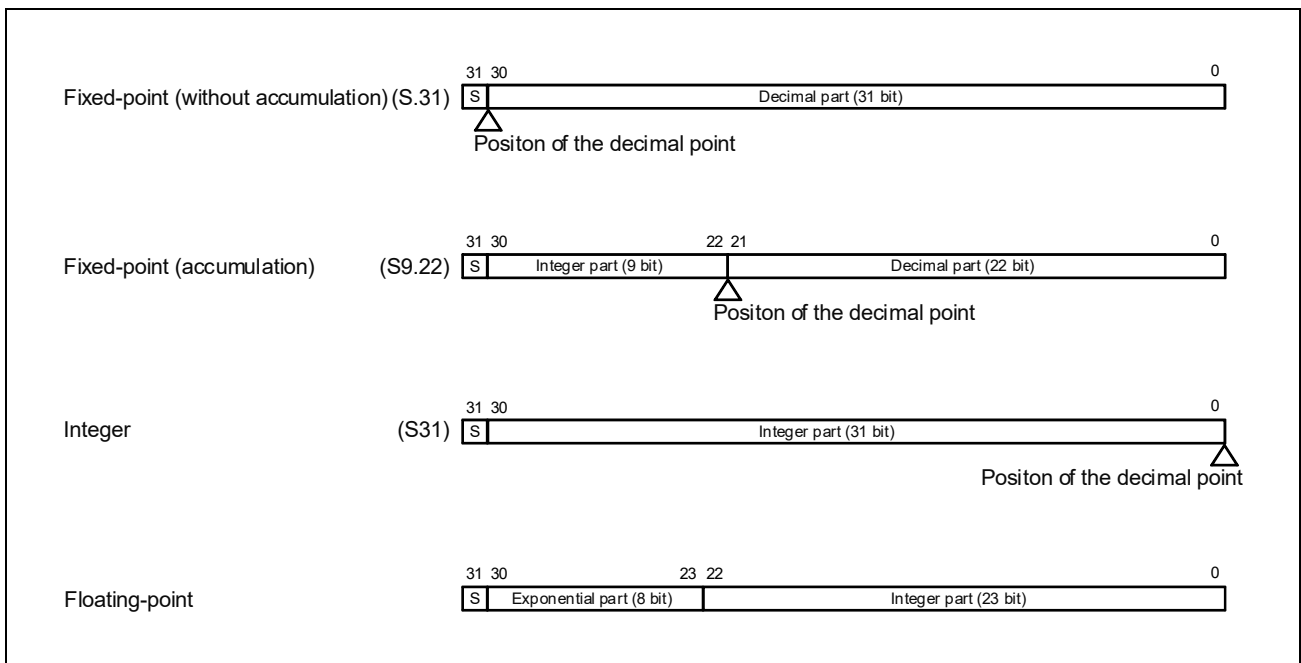


Figure 37.12 The Data Format of the Capture Result Data Register

37.3.52 DFEjCAPSTCHn — Capture Status Register n (n = 0 to 2)

This register indicates the status of the capture channel.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CDOEN	—	—	—	—	—	—	CDOOW	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.62 DFEjCAPSTCHn Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	CDOEN	Capture Output Data Register Valid 0: Value after a reset Indicates that the value of the capture output data register is invalid. While CDOEN = 1, writing 1 to DFEjCAPCLRSTCHn.CLRCDEN clears this bit to 0. While CDOEN = 1, reading the DFEjCAPDOCHn register clears this bit to 0. 1: Indicates that the value of the capture output data register is valid. This bit is set to 1 when the DFE stores the capture results in the capture output data register.
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CDOOW	Capture Result Register Overwrite Error 0: value after a reset No capture result register overwrite error. When CDOOW = 1, writing 1 to DFEjCAPCLRSTCHn.CLRCDOOW clears CDOOW to 0. 1: A capture result register overwrite error occurred. When CDOEN = 1, storing capture data in the capture result register sets CDOOW to 1
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

37.3.53 DFEjCAPCLRSTCHn — Capture Clear Status Register n (n = 0 to 2)

This register is used to clear the capture status register. Writing 1 to these bits clears the corresponding DFEjCAPSTCHn bits to 0.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLR CDOEN	—	—	—	—	—	—	CLR CDOO W	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R

Table 37.63 DFEjCAPCLRSTCHn Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	CLRCDOEN	Capture Result Data Enable Bit Clear 0: DFEjCAPSTCHn.CDOEN is not cleared: Value after a reset. 1: DFEjCAPSTCHn.CDOEN is cleared. When DFEjCAPSTCHn.CDOEN is cleared, this bit is automatically cleared to 0. CAUTION: This bit is always read as 0.
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CLRCDOOW	Capture Result Data Register Overwrite Error Bit Clear 0: DFEjCAPSTCHn.CDOOW is not cleared: Value after a reset. 1: DFEjCAPSTCHn.CDOOW is cleared. When DFEjCAPSTCHn.CDOOW is cleared, this bit is automatically cleared to 0. CAUTION: This bit is always read as 0.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

37.3.54 DFEjCAPERMCHn — Capture Error Mask Register n (n = 0 to 2)

This register is used to mask errors in the capture channel.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSK CDOO W	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 37.64 DFEjCAPERMCHn Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	MSKCDOOW	Capture Result Data Register Overwrite Error Bit Mask 0: DFEjCAPSTCHn.CDOOW is not masked: Value after a reset. 1: DFEjCAPSTCHn.CDOOW is masked.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

37.3.55 DFBFACCTL — Buffer A Common Control Register

This is a FIFO macro register. Common settings on buffer A processing are specified using this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AUNE	AIEE	AIEO	AEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Note: When DFBFACCTL.AEN is 1, writing to AUNE, AIEE and AIEO is prohibited.

Table 37.65 DFBFACCTL Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AUNE	Buffer A Data Update Flag Enable 0: Value after a reset. The buffer A data update flag (DFBFAD0CHn.AUN) is not used. For data format, see the buffer A output register. 1: The buffer A data update flag (DFBFAD0CHn.AUN) is used. For data format, see the buffer A output register.
2	AIEE	Buffer A Error Interrupt Request Enable 0: Value after a reset Buffer A error interrupt request is disabled. 1: Buffer A error interrupt request is enabled.
1	AIEO	Buffer A Output Data Interrupt Request Enable 0: Value after a reset Buffer A output data interrupt is disabled when output data from the selected DFE channel is written to the buffer A output data register. 1: Buffer A output data interrupt is enabled when output data from the selected DFE channel is written to the buffer A output data register.
0	AEN	Buffer A Enable 0: Value after a reset Buffer A is disabled. No Buffer A processing is executed. 1: The Buffer A processing is executed.

37.3.56 DFBFACTLCHn — Buffer A Control Register n (n = 0 to 5)

This is a FIFO macro register. Settings for each buffer A channel are specified using this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	BFACH				—	—	—	—	ADSL	—	—	CHEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R	R	R/W

Note: When DFBFACCTL.AEN is 1, writing to BFACH, ADSL and CHEN is prohibited.

Table 37.66 DFBFACTLCHn Register Contents

Bit Position	Bit Name	Function
31 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 8	BFACH	Buffer A Capture-target Channel Select 0 _H : Channel 0 is set to the buffer A capture-target channel. Value after a reset 1 _H : Channel 1 is set to the buffer A capture-target channel 2 _H : Channel 2 is set to the buffer A capture-target channel ... B _H : Channel 11 is set to the buffer A capture-target channel C _H to F _H : Setting prohibit. Note: DFE1 is prohibited from 4 _H to F _H .
7 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	ADSL	Buffer A DFE Select 0: Buffer A channel n is selected from DFE0. 1: Buffer A channel n is selected from DFE1.
2, 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CHEN	Buffer A Channel n Enable 0: Value after a reset Buffer A channel n disabled When CHEN = 0, buffer A channel n data is not stored in FIFO. 1: Buffer A channel n enabled When CHEN = 0, buffer A channel n data is stored in FIFO.

37.3.57 DFBFADOCHn — Buffer A Output Register n (n = 0 to 5)

This register is a buffer A output data register that holds the output data from the capture-target channel.

A FIFO data format accords with the data format of capture-target channel.

If the data is read with floating-point conversion format, set the output data register (DFEjDOCHn) of the capture-target channel to floating-point conversion format.

- When DFBFACCTL.AUNE = 0:

Value after reset: 0000 0000_H

When DFBFACCTL.AUNE is set to 0, after clear is valid by DFBFACLR.CLRA, the clear value is 0000 0000_H.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.67 DFBFADOCHn Register Contents

Bit Position	Bit Name	Function
31 to 0	ADO	FIFO Storage Data The value of the output data register which was stored in an 8-stage FIFO is stored in these bits.

The data that is stored in ADO [31:0] is output data (DFEjDOCHn.DO[31:0]).

When DFBFACCTL.AUNE = 0, the data format of buffer A output register is shown below.

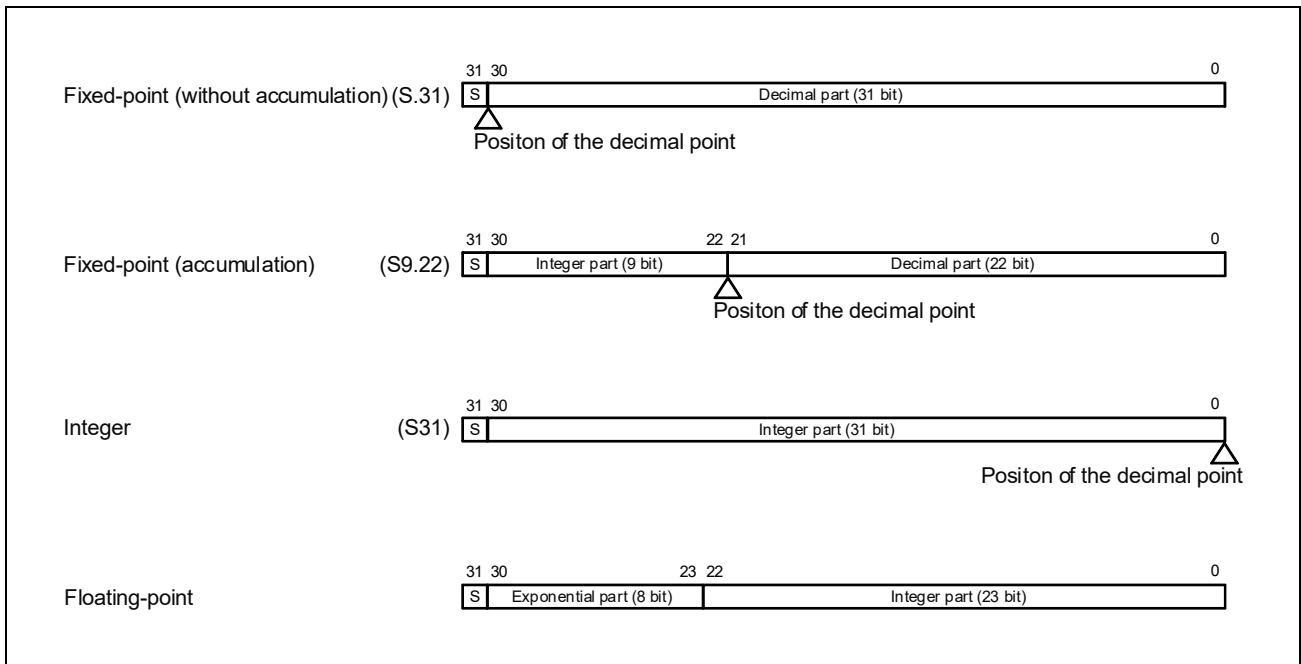


Figure 37.13 The Data Format of Buffer A Output Register (AUNE = 0)

- When DFBFACCTL.AUNE = 1:

Refer to the subsection with DFBFACCTL.AUNE = 0 for information about the reset value.

When DFBFACCTL.AUNE is set to 1, after clear is valid by DFBFACLR.CLRA, the clear value is 0000 0001_H.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADO															AUN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.68 DFBFADOCHn Register Contents

Bit Position	Bit Name	Function
31 to 1	ADO	FIFO Storage Data The value of the output data register which was stored in an 8-stage FIFO is stored in these bits. Note: The data that is stored in ADO [30:0] is output data (DFEjDOCHn.DO[31:1]).
0	AUN	Buffer A Data Update Flag 0: Indicates that the output data from the buffer A capture-target channel stored in the FIFO has been updated. 1: Value after a reset Indicates that the output data from the buffer A capture-target channel stored in the FIFO has not been updated.

When DFBFACCTL.AUNE = 1, the data format of buffer an output register is shown below.

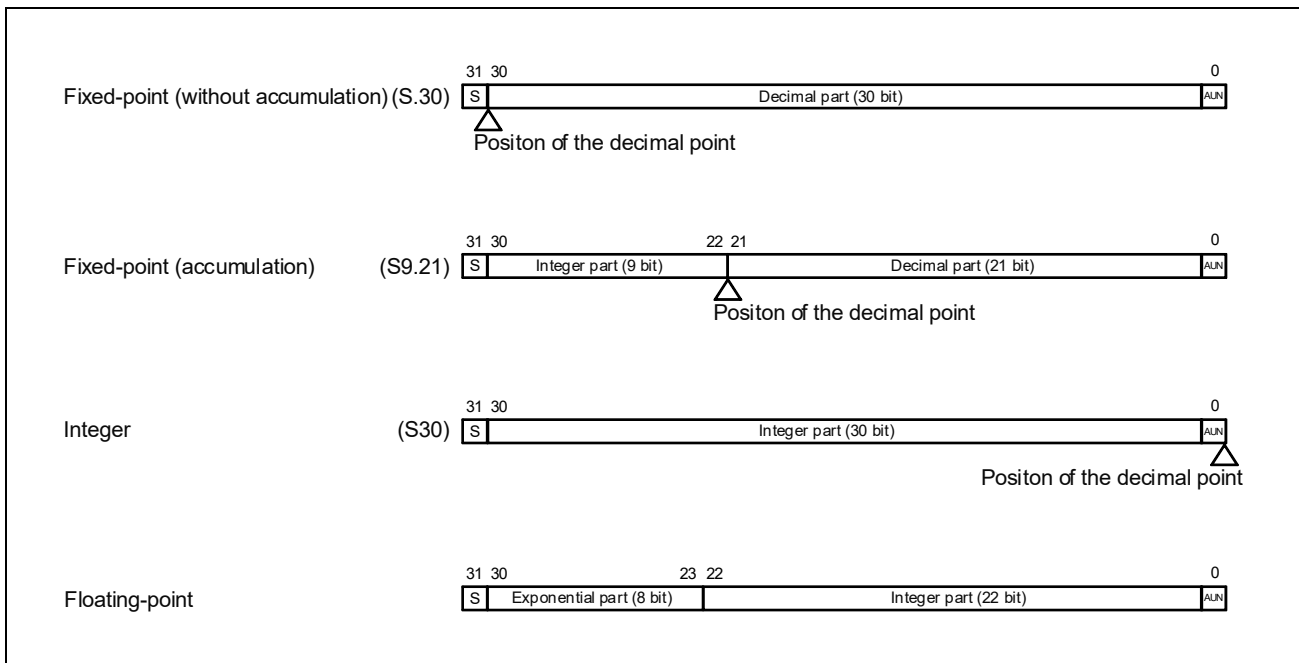


Figure 37.14 The Data Format of Buffer A Output Register (AUNE = 1)

37.3.58 DFBFACLR — Buffer A Clear Register

This register clears the buffer A.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 37.69 BFACLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CLRA	Buffer A Clear 0: Value after a reset The buffer A FIFO is not cleared. 1: The buffer A FIFO is cleared. All bits in the buffer A common status register are initialized. All of the FIFO channel's Buffer A Data Update Flags (DFBFADOCHn.AUN) are set to 1. All of the FIFO channel's FIFO Storage Data (DFBFADOCHn.ADO) are set to ALL 0.
Note: Always read as 0.		

37.3.59 DFBFACST — Buffer A Common Status Register

This register indicates the status of each buffer A channel.

Value after reset: 003F 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	EMPA5	EMPA4	EMPA3	EMPA2	EMPA1	EMPA0
Value after reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	OVFA5	OVFA4	OVFA3	OVFA2	OVFA1	OVFA0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.70 DFBFACST Register Contents

Bit Position	Bit Name	Function
31 to 22	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
21 to 16	EMPA _n	FIFO Channel n Empty Status 0: Indicates that some data remains on FIFO channel n. When EMPA _n = 1, storing data in the FIFO clears this bit to 0. 1: Value after a reset Indicates that FIFO channel n is empty. When EMPA _n = 0 and when 1 _B is written in DFBFACLR.CLRA, EMPA _n for all FIFO channels are set to 1 _B .
15 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	OVFA _n	FIFO Channel n FIFO Overflow Error 0: Value after a reset Indicates no FIFO overflow occurred on FIFO channel n. When OVFA _n = 1 and when 1 _B is written in DFBFACLR.CLRA, OVFA _n for all FIFO channels are set to 0. 1: Indicates a FIFO overflow occurred on FIFO channel n. When the FIFO channel n is full and when new data is stored on FIFO channel n, OVFA _n is set to 1 _B .

37.3.60 DFBFBCTL — Buffer B Control Register

This is a FIFO macro register. Settings for buffer B processing are specified using this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	BFBCH				—	—	—	—	BDSL	BIEE	BIEO	BEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note: When DFBFBCTL.BEN is 1, writing to BFBCH, BDSL, BIEE, and BIEO is prohibited.

Table 37.71 DFBFBCTL Register Contents

Bit Position	Bit Name	Function
31 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 8	BFBCH	Buffer B Capture-target channel Select 0 _H : Channel 0 is set to the buffer B capture-target channel. Value after a reset 1 _H : Channel 1 is set to the buffer B capture-target channel 2 _H : Channel 2 is set to the buffer B capture-target channel ... B _H : Channel 11 is set to the buffer B capture-target channel C _H to F _H : Setting prohibited Note: DFE1 is prohibited from 4 _H to F _H .
7 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	BDSL	Buffer B DFE Select 0: Buffer B channel n is selected from DFE0. 1: Buffer B channel n is selected from DFE1.
2	BIEE	Buffer B Error Interrupt Request Enable 0: Value after a reset Buffer B error interrupt request is disabled. 1: Buffer B error interrupt request is enabled.
1	BIEO	Buffer B Output Data Interrupt Request Enable 0: Value after a reset Buffer B output data interrupt is disabled when output data from the selected DFE channel is written to the buffer B output data register. 1: Buffer B output data interrupt is enabled when output data from the selected DFE channel is written to the buffer B output data register.
0	BEN	Buffer B Enable 0: Value after a reset Buffer B is disabled. No Buffer B processing is executed. 1: Buffer B is enabled. Buffer B processing is executed.

37.3.61 DFBFBDO — Buffer B Output Register

This register is a buffer B output data register that holds the output data from the capture-target channel.

The FIFO data format accords with the data format of the capture-target channel.

If the data is read with floating-point conversion format, set the output data register (DFEjDOCHn) of the capture-target channel to floating-point conversion format.

Value after reset: 0000 0000_H

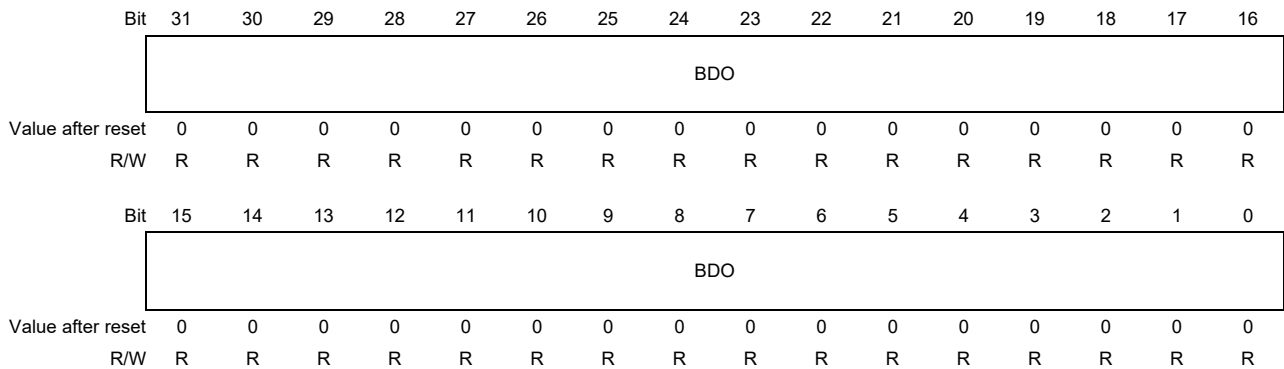


Table 37.72 DFBFBDO Register Contents

Bit Position	Bit Name	Function
31 to 0	BDO	FIFO Storage Data The value of the output data register which was stored in an 8-stage FIFO is stored in these bits.

The data that is stored in BDO [31:0] is the output data (DFEjDOCHn.DO[31:0]).

The data format of the buffer B output register is shown below.

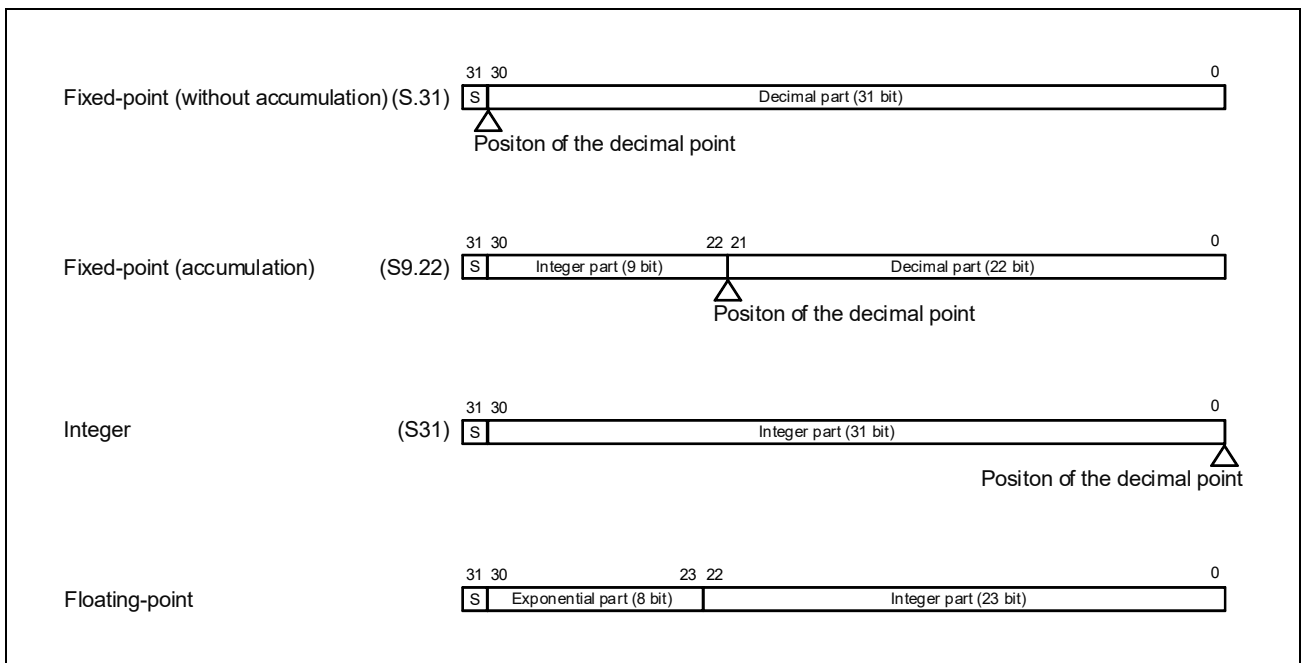


Figure 37.15 The Data Format of Buffer B Output Register

37.3.62 DFBFBCLR — Buffer B Clear Register

This register clears the buffer B.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*

Table 37.73 DFBFBCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CLRB	Buffer B Clear 0: Value after a reset The buffer B FIFO is not cleared. 1: The buffer B FIFO is cleared. All bits in the buffer B common status register are initialized. Buffer B's FIFO Storage Data (DFFBDO.BDO) is set to ALL 0. Note: Always read as 0.

37.3.63 DFBFBST — Buffer B Status Register

This register indicates the status of each buffer B channel.

Value after reset: 0000 0002_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EMPB	OVFB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.74 DFBFBST Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	EMPB	FIFO Channel n Empty Status 0: Indicates that some data remains on FIFO channel n. When EMPB = 1, storing data in the FIFO clears this bit to 0. 1: Value after a reset Indicates that FIFO channel n is empty. When EMPB = 0 and when 1 _B is written in DFBFBCLR.CLRB, EMPB is set to 1 _B .
0	OVFB	FIFO B Overflow Error 0: Value after a reset Indicates no FIFO overflow occurred. When OVFB = 1 and when 1 _B is written in DFBFBCLR.CLRB, OVFB is set to 0. 1: Indicates A FIFO overflow occurred. When FIFO B is full and when new data is stored in FIFO B, OVFB is set to 1 _B .

37.4 Operation

37.4.1 Overview of Operation

37.4.1.1 DFE Macro

The DFE imports output data from the Cyclic-ADC, Delta-Sigma ADC or the SAR-ADC (referred to as AD hereafter) and performs filter (FIR or IIR) processing, accumulation/decimation processing, subtraction processing, peak-hold/comparison calculation processing, or floating-point conversion processing for the output data.

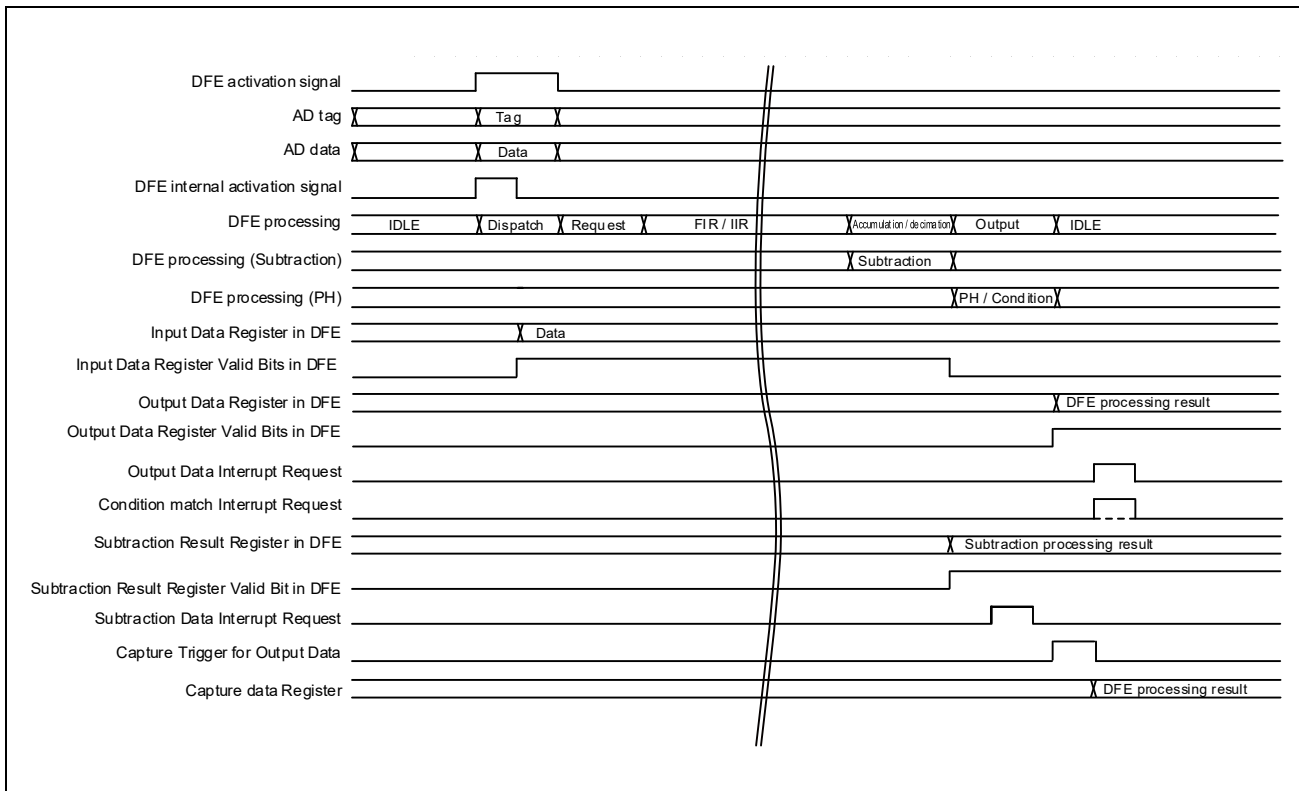


Figure 37.16 Overview of Operation

- The AD inputs the DFE activation signal, AD tag, and AD conversion data (AD data) to the DFE. The DFE imports the AD data in the internal input data register to execute dispatch processing, request processing, FIR/IIR processing, accumulation/decimation processing, subtraction processing, output processing, or peak-hold/comparison processing, and then stores the DFE processing result in the output data register.
- **Dispatch processing:**
The DFE stores data in the input data register of the relevant channel by using the AD tag from the AD and DFE's control register values. Before data is stored, the input data register is enabled.
- **Request processing:**
The DFE arbitrates FIR/IIR processing target data from the input data register of 16 channels and inputs the filter processing target data to the FIR/IIR processing.
- **FIR/IIR processing (filter processing):**
The DFE performs FIR or IIR filter calculation. Coefficient data and intermediate values in the filter calculation are stored in the memory.
- **Subtraction processing:**
The DFE executes subtraction on the results of a FIR/IIR between two different channels. After the subtraction, the

DFE stores the results in the subtraction result register. When storing the results, the subtraction result register is enabled, and the macro executes a subtraction data interrupt request.

- Accumulation/decimation processing:

The DFE executes accumulation processing or decimation processing for the FIR or IIR processing result. After the accumulation/decimation processing has been completed, the input data register is disabled.

- Output processing:

The DFE stores the accumulation/decimation processing result in the output data register of the relevant channel. Before the result is stored, the output data register is enabled and an output data interrupt request is issued.

- Peak-Hold/comparison processing:

The DFE executes Peak-Hold and/or comparison processing on the results of accumulation/decimation processing. If the Peak-Hold processing is selected, the macro issues a condition match interrupt request when the Peak-Hold end trigger is enabled. If comparison processing is selected, the DFE executes a condition match interrupt request when the result of the comparison operation is TRUE. If Peak-Hold processing and comparison processing are executed simultaneously, in the Peak-Hold processing the macro issues a Peak-Hold end interrupt request; in the comparison processing, the macro issues a condition match interrupt request when the result of the comparison operation is TRUE.

- Capture processing:

If an output data capture trigger is input to the output data register, the macro stores the value of the output data register in the capture result register.

37.4.1.2 FIFO Macro

The FIFO macro selects a DFE macro output data register and stores the data in the FIFO. The FIFO macro selects the required channel from the DFE0 and DFE1, and stores the data in the FIFO.

The macro operates differently between buffer A processing and buffer B processing.

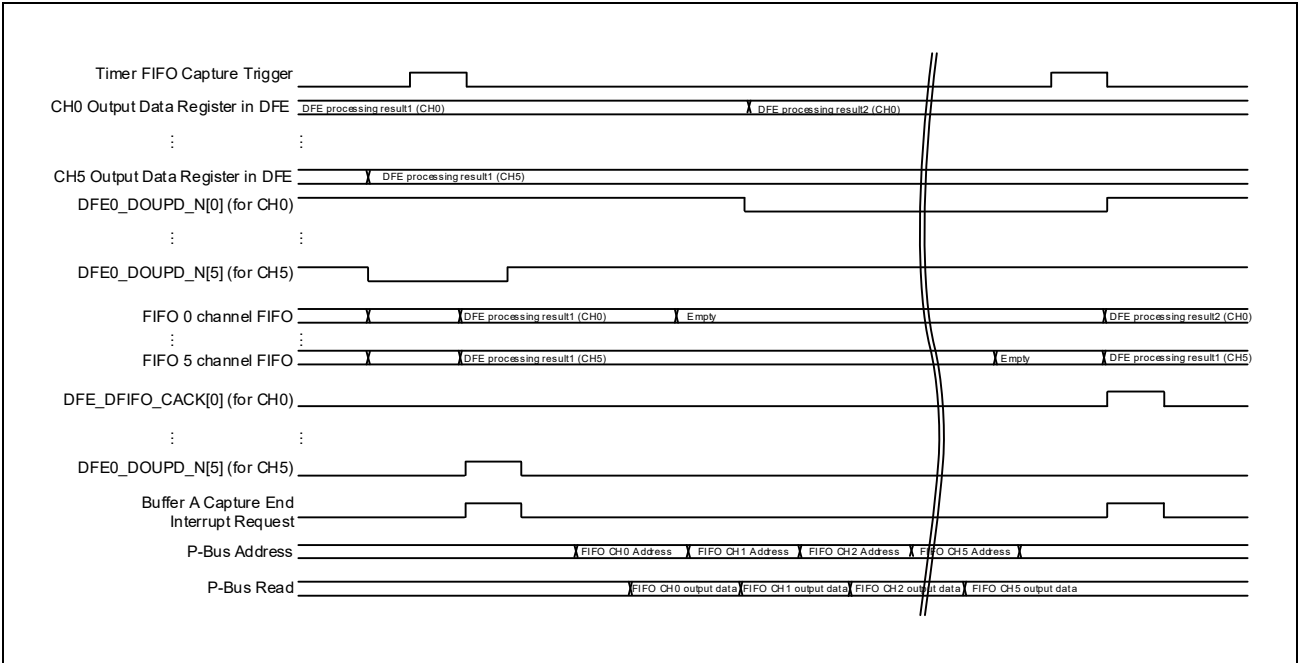


Figure 37.17 Overview of FIFO Macro Operation (buffer A processing)

- Buffer A processing:

The macro stores, in batch, data in the FIFO channel which was enabled by the FIFO capture trigger that was output from the timer. When storing data in the FIFO, the macro issues a buffer A capture end interrupt request. After reading all FIFO channels in which data is stored, the macro issues a buffer A capture end interrupt request if data is stored in the FIFO again. When not reading all FIFO channels in which data is stored, the macro will not issue a buffer A capture end interrupt request even if data is stored in the FIFO again.

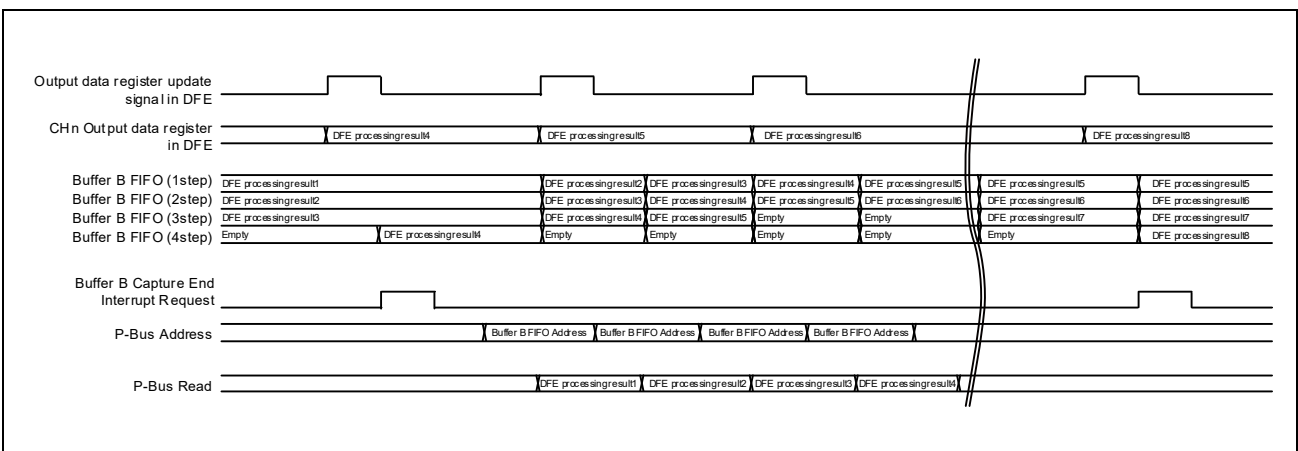


Figure 37.18 Overview of FIFO Macro Operation (buffer B processing)

- Buffer B processing:

Each time the value of the DFE output data register is updated, the macro stores the value of the output data register of a selected channel in one FIFO. Repeating this operation until the values from four output data registers have been stored in the FIFO, the macro issues a buffer B capture end interrupt request.

After reading all four data items the macro issues a buffer B capture end interrupt request if four data items are stored in the FIFO again.

When not reading all four stored data items, the macro will not issue a buffer B capture end interrupt request even if four data items are stored in the FIFO again.

37.4.2 Operating Procedures

37.4.2.1 Channel Start Procedure

The procedure to start the DFE channel is shown below.

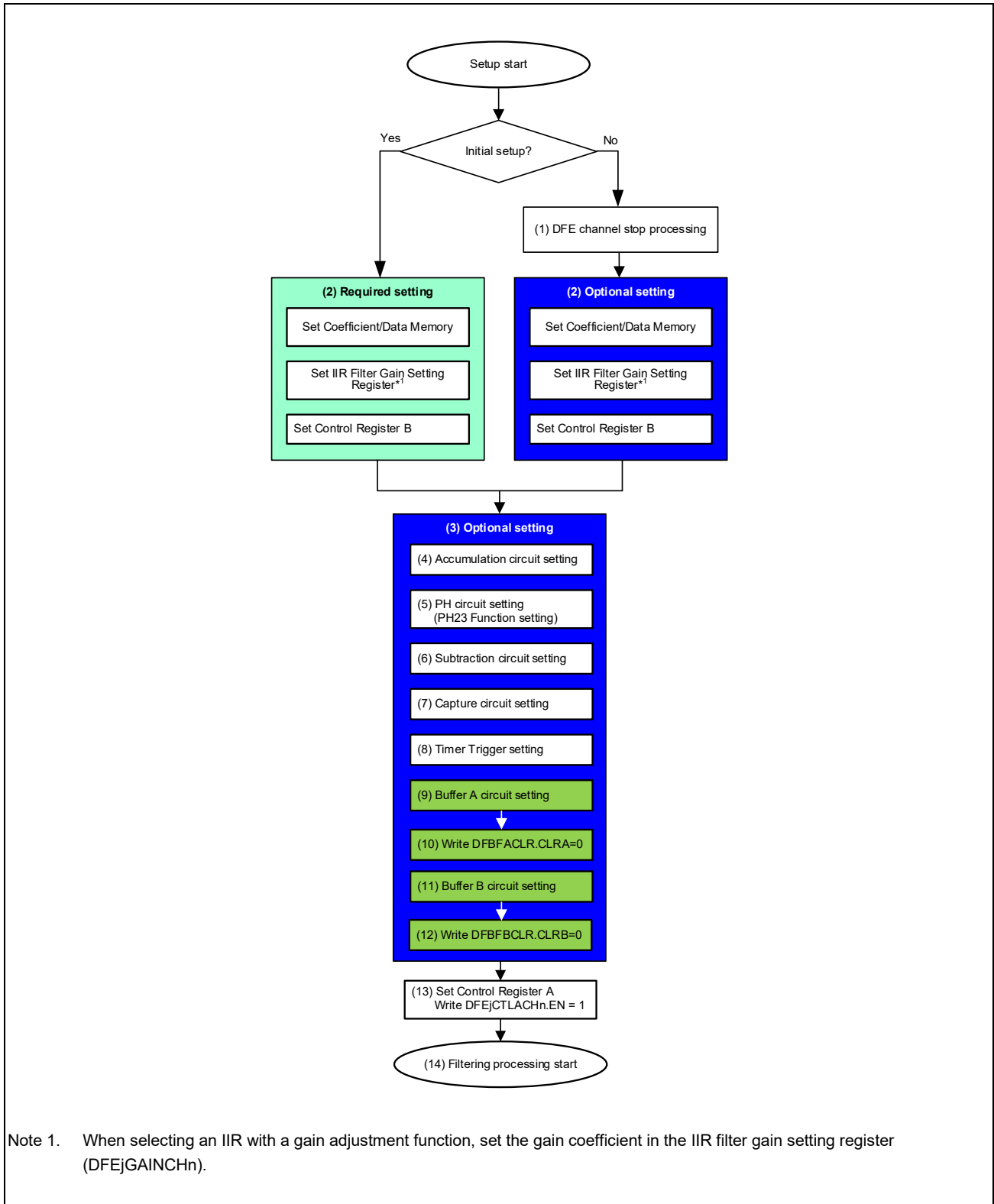


Figure 37.19 Channel Starting Procedure

The channel start setup process of the DFE is different for the first time versus the subsequent times.

Please refer to step (2) in **Figure 37.19** for the first time setup process. Please refer to step (1) in **Figure 37.19** for the restart process.

- (1) For the restart setup process, set the DFE to stop according to **Section 37.4.2.2, Channel Stop Procedure**.
- (2) For the first time setup process, set the registers as shown in **Figure 37.19**. For the restart process, set the registers as shown in **Figure 37.19** as necessary.
 - For the coefficient memory and data memory, set values in the channel area to be used via the peripheral bus. For the coefficient memory address assignment, see **Section 37.5.3, Section 37.5.4, and Section 37.5.9**. For the data memory address assignment, see **Section 37.5.3, Section 37.5.5, and Section 37.5.9**.
- (*) When selecting an IIR with the gain adjustment function, set the gain coefficient in the IIR filter gain setting register (DFEjGAINCHn).
 - Set control register B (DFEjCTLBCHn) for accumulation/decimation and PH/comparison operation.
 - If DFEjCTLBCHn.DISA = 0, after writing DFEjCTLACHn.EN = 1 in step (13), the macro starts the accumulation/decimation processing on the first input data.
 - If DFEjCTLBCHn.DISB = 0, after writing DFEjCTLACHn.EN = 1 in step (13), the macro starts the peak-hold processing on the first input data.
- (3) Set the registers according to the specific functions to be used, in any order.
- (4) When the accumulation circuit is used, set the accumulation /decimation count setting registers (DFEjACA to DFEjACD).
- (5) When the Peak-Hold circuit is used, set the following registers as necessary:

Peak-Hold Initial Value Setting Registers	DFEjPHIA to DFEjPHID
Comparison Value Setting Registers	DFEjCPA to DFEjCPD
Peak-Hold Update Notification Setting Register	DFEjPHUPDCn

When using the PH23 function, set the following registers:

Peak-Hold 23 Common Control Register 0	DFEjPH23CCTL
Peak-Hold 2 Control Register 0	DFEjPH2CTL
Peak-Hold 3 Control Register 0	DFEjPH3CTL
Comparison Offset Value Setting Register	DFEjCPOFSTn

- (6) When using the subtraction circuit, set the subtraction control registers (DFEjSUBCTLCHn).
- (7) When using the capture circuit, set the capture control registers (DFEjCAPCTLCHn).

- (8) When using a trigger flag, the trigger setting register and the timer trigger setting register need to be set. Set the timer trigger setting registers for the pertinent trigger flag functions according to the specific functions to be used.

Trigger Setting Register	DFEJTRGCHn
Timer Trigger Setting Register	DFEJTMTRGCHn
Software Trigger Register	DFEJTRG
Peak-Hold Initialization/End Timer Trigger Select Register	DFEJPITRG
Peak-Hold Initialization/End Timer Trigger Select Register1	DFEJPITRG1
Accumulation/Decimation Initialization/Prohibition Timer Trigger Select Register	DFEJMITRG
Accumulation/Decimation Initialization/Prohibition Timer Trigger Select Register1	DFEJMITRG1
Filter Initialization Timer Trigger Select Register	DFEJFITRG
Filter Initialization Timer Trigger Select Register 1	DFEJFITRG1
Peak-Hold Mask-Start/End Timer Trigger Select Register 0	DFEJPMITRG0
Peak-Hold Mask-Start/End Timer Trigger Select Register 1	DFEJPMITRG1
Subtraction Trigger Setting Register	DFEJSUBTRGCHn
Subtraction Start/End Timer Trigger Select Register 0	DFEJSUBTRG0

- (9) When the buffer A circuit is used, set the buffer A common control register (DFBFACCTL) and the buffer A control registers (DFBFACTLCHn).
- (10) When the buffer A circuit is used, after the Buffer A circuit is set in step (9), initialize the buffer A circuit by the buffer A clear register (DFBFACLR).
- (11) When the buffer B circuit is used, set the buffer B control registers (DFBFBCTL).
- (12) When the buffer B circuit is used, after the Buffer B circuit is set in step (11), initialize the buffer B circuit by the buffer B clear register (DFBFBCLR).
- (13) When all settings have been completed, use control register A (DFEjCTLACHn) to set filter processing, enable writing to the interrupt request output, enable the channel, and set other necessary parameters. Setting the enable bit (DFEjCTLACHn.EN) for the channel to be used to 1 enables the filter processing on that channel.
- (14) When an activation trigger is input from the AD, the macro starts the filter processing on the input data from the AD.

37.4.2.2 Channel Stop Procedure

The procedure to stop the DFE channel is shown below.

Once the registers have been set and filter processing has started, execute the channel stop procedure as follows. Set the registers according to **Section 37.4.2.1, Channel Start Procedure**.

When the registers of the Buffer A circuit have been set and the filter processing has started, execute the channel stop procedure for all channels selected by the FIFO channels as follows. Set the registers according to **Section 37.4.2.1, Channel Start Procedure**.

When the registers of the Buffer B circuit have been set and the filter processing has started, execute the channel stop procedure for the target channel as follows. Set the registers according to **Section 37.4.2.1, Channel Start Procedure**.

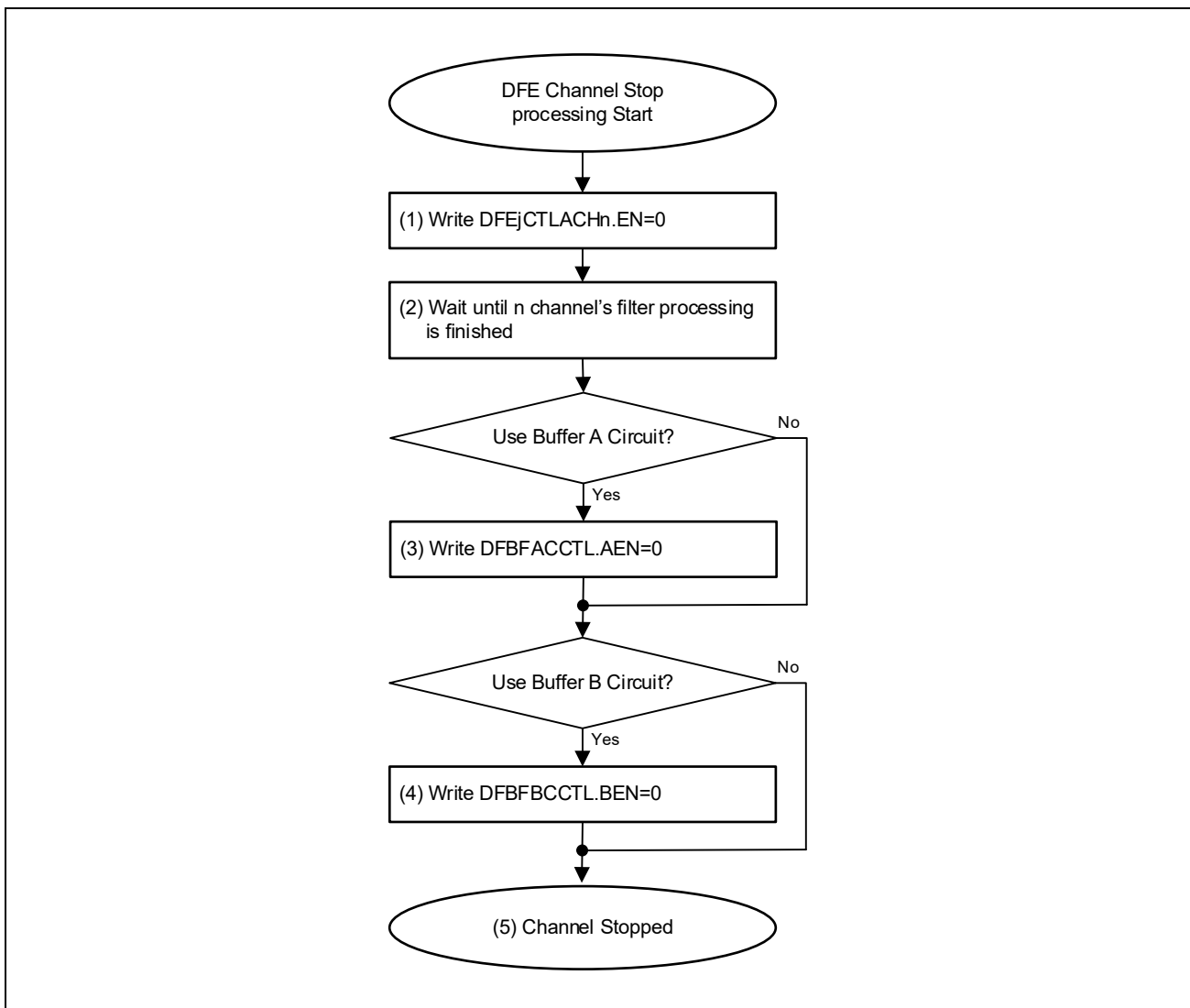


Figure 37.20 Channel Stop Procedure

- (1) Clear the enabling bit (DFEjCTLACHn.EN) for the channel to be stopped.
- (2) Wait until the processing on the stopped channel terminates.
When the buffer A circuit is used, after all channels which are target to FIFO channels have stopped, set the Buffer A Enable (DFBFACCTL.AEN) to 0. By setting DFBFACCTL.AEN = 0, the Buffer A circuit is stopped.

- (3) When the buffer B circuit is used, after the target channel of Buffer B has stopped, set the Buffer B Enable (DFBFBCTL.BEN) to 0. By setting DFBFBCTL.BEN = 0, Buffer B circuit is stopped.
- (4) The text below describes the operations that occur during the stopping of channel processing:
 - Writing 0 to the enable bit (DFEjCTLACHn.EN) suppresses any subsequent processing of input data and input triggers.
 - Any data that was input before 0 was written to the enable bit (DFEjCTLACHn.EN) continues to be processed after 0 is written.
 - After writing 0 to the enable bit (DFEjCTLACHn.EN), use DFEjSTCHn.VALID to confirm the termination of processing after 0 is written.
 - Any trigger that was input before 0 is written to the enable bit (DFEjCTLACHn.EN) is disabled if there is no data to be processed.

After the channel is stopped, clear the status bits for that channel, for the subtraction channel, and for the capture channel as necessary.

37.4.2.3 Error Processing Procedure

(1) DFE Macro

The DFE operating procedure for an error is shown below.

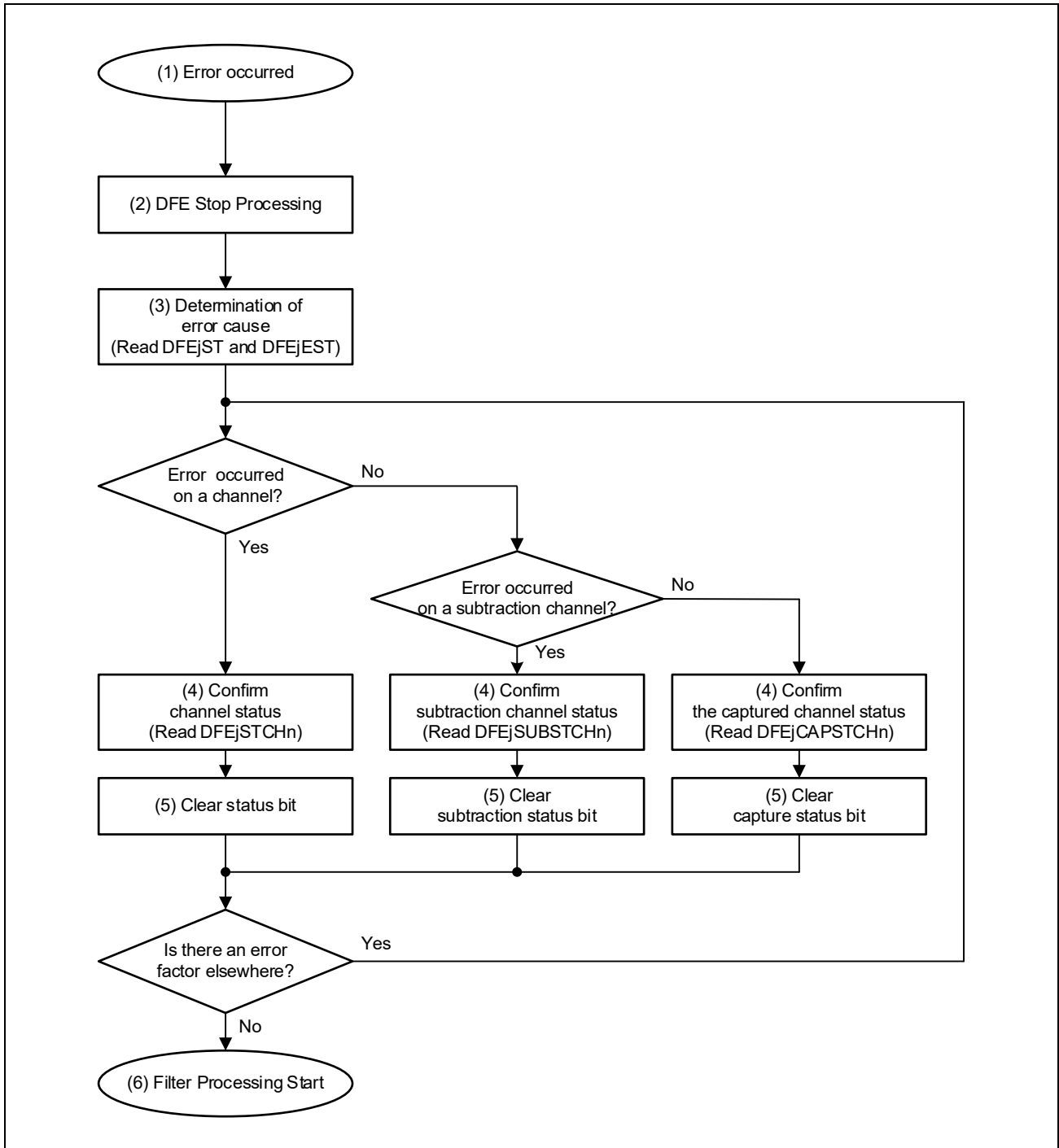


Figure 37.21 DFE Macro Operating Procedure for an Error

- (1) If an error occurs during the processing, the DFE macro issues an interrupt request. Possible sources of errors include (1) channels, (2) subtraction channels, and (3) capture channels. For each of these sources, error interrupts and error sources can be masked by the following registers.

Classification	Error Interrupt Mask	Error Source Mask
Channel	DFEjCTLACHn.IEE	DFEjERMCHn
Subtraction channel	DFEjSUBCTLCHn.SIEE	DFEjSUBERMCHn
Capture channel	DFEjCAPCTLCHn.CIEE	DFEjCAPERMCHn

- (2) The macro performs DFE stop processing according to the channel stop processing flow of events.
- (3) The macro identifies the channel, the subtraction channel or the capture channel on which error occurred. By reading the DFEjEST register, the channel on which error occurred can be identified. By reading the DFEjEST register, the subtraction channel or the capture channel on which error occurred can be identified.
- (4) The error sources can be identified by checking the status register for the channel, subtraction channel, or capture channel on which the error occurred.
(If an error occurred on channel 0, for example, the macro reads DFEjSTCH0.)
- (5) When an error source is identified, the macro writes to the clear status bits corresponding to the error status, and clears the error bits.
(Note) If an error status is not cleared, and if the error status recurs, no error interrupt will be issued.
- (6) The macro starts filter processing according to the channel start processing flow of events.

(2) FIFO Macro

The FIFO macro operating procedure for an error is shown below.

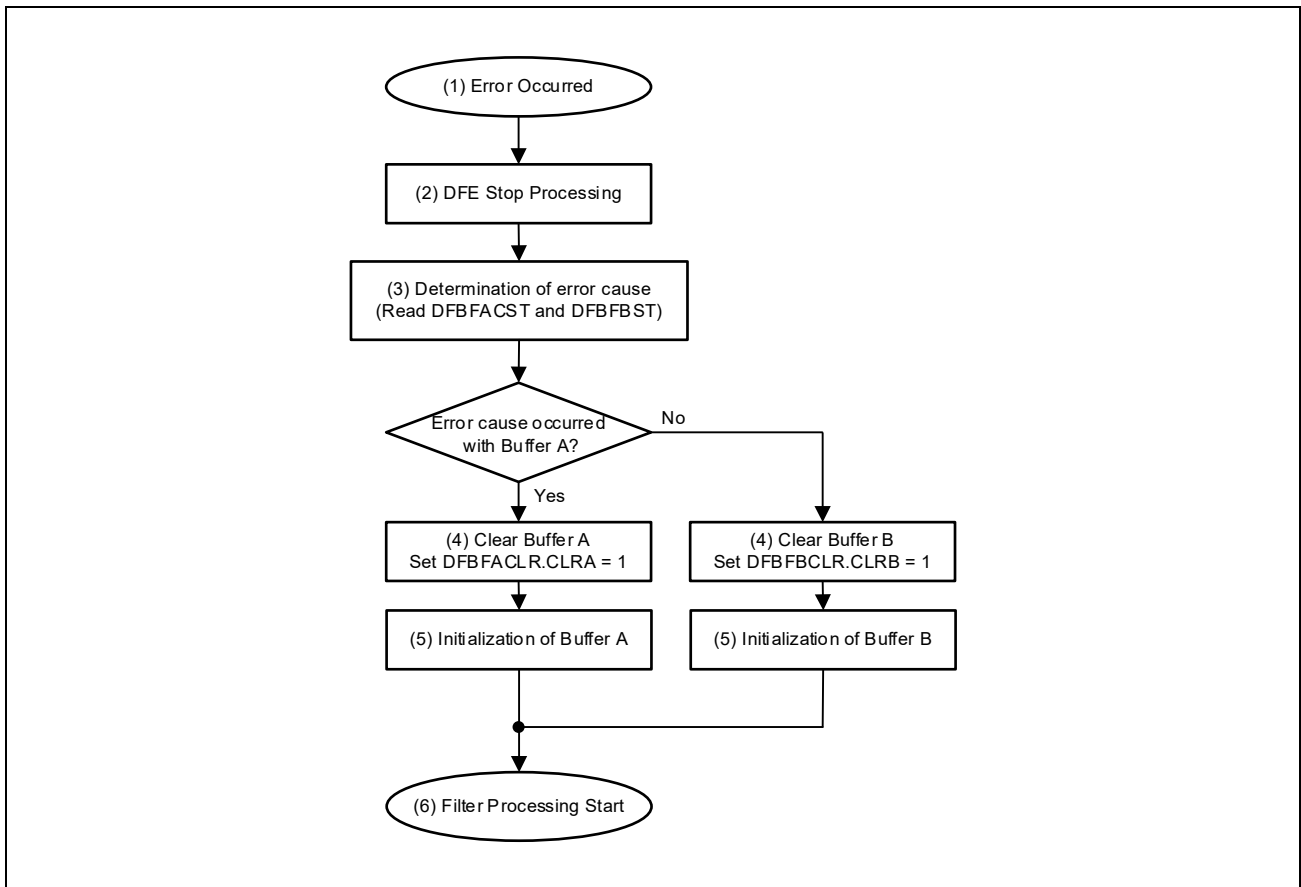


Figure 37.22 FIFO Macro Operating Procedure for an Error

- (1) If an error occurs during processing, the FIFO macro issues an interrupt request.

Possible sources of errors include buffer A and buffer B.

For each of these sources, error interrupt request outputs can be controlled by the following registers.

Classification	Mask Bit of Error Interrupt Request
Buffer A processing	DFBFACCTL.AIEE
Buffer B processing	DFBFBCTL.BIEE

- (2) The macro performs DFE stop processing according to the channel stop processing flow of events.
- (3) The macro identifies the source of the error as buffer A or B.
By reading the DFBFACST register, the macro can identify the FIFO channel on which the buffer A error occurred. By reading the DFBFBST register, the macro can determine that the error occurred in buffer B.
- (5) When an error source is identified, the macro writes to the clearing register, and initializes buffer A or B.
- (6) The macro starts filter processing according to the channel start processing flow of events.

(3) About register rewrite

When an R/W register is written, the bit value is reflected in the DFE processor immediately. When each function is executed, and when an R/W register is changed, the value of an operation result and the status are not guaranteed. Therefore, when an R/W register is changed, it is necessary to stop each function.

Rewrite the following registers when the target channel is stopped (DFEjCTLACHn.EN=0 & DFEjSTCHn.VALID=0).

Register Name	Remarks
DFEjCTLACHn	All bits except EN
DFEjCTLBCHn	All bits
DFEjTRGCHn	All bits
DFEjTMTRGCHn	All bits
DFEjERMCHn	All bits
DFEjPHUPDCn	All bits
DFEjGAINCHn	All bits

Rewrite the following registers when the target channel of the Peak-Hold 23 function is stopped (DFEjCTLACHn.EN=0 & DFEjSTCHn.VALID=0).

Register Name	Remarks
DFEjPH23CCTL0	All bits except PEN
DFEjPH2CTL0	All bits
DFEjPH3CTL0	All bits

Rewrite the following registers when the target channel of the subtraction function is stopped (DFEjCTLACHn.EN=0 & DFEjSTCHn.VALID=0).

Register Name	Remarks
DFEjSUBCTLCHn	All bits except SEN
DFEjSUBTRGCHn	All bits
DFEjSUBERMCHn	All bits

Rewrite the following registers when the capture function is stopped (DFEjCAPCTLCHn.CEN=0).

Register Name	Remarks
DFEjCAPCTLCHn	All bits except CEN
DFEjCAPERMCHn	All bits

Rewrite the following registers when the Buffer A circuit is stopped (DFBFACCTL.AEN=0).

Register Name	Remarks
DFBFACCTL	All bits except AEN
DFBFACTLCHn	All bits

Rewrite the following register when the Buffer B circuit is stopped (DFBFBCTL.BEN=0).

Register Name	Remarks
DFBFBCTL	All bits except BEN

When the following registers are used by multiple channels, rewrite these registers when these channels are stopped (DFEjCTLACHn.EN=0 & DFEjSTCHn.VALID=0).

Register Name	Remarks
DFEjCPA - DFEjCPD	All bits
DFEjPHIA - DFEjPHID	All bits
DFEjACA - DFEjACD	All bits

Rewrite the following registers when the channels which use timer trigger n are stopped (DFEjCTLACHn.EN=0 & DFEjSTCHn.VALID=0).

Register Name	Remarks
DFEjPITRG	All bits
DFEjPITRG1	All bits
DFEjPMITRG0	All bits
DFEjMITRG	All bits
DFEjMITRG1	All bits
DFEjFITRG	All bits
DFEjFITRG1	All bits

Rewrite the following register when the target channel of the subtraction function which use timer trigger n are stopped (DFEjCTLACHn.EN=0 & DFEjSTCHn.VALID=0).

Register Name	Remarks
DFEjSUBTRG0	All bits

37.5 Details

37.5.1 Data Flow

37.5.1.1 DFE Macro

Figure 37.23 shows the data flow and the block diagram of the DFE.

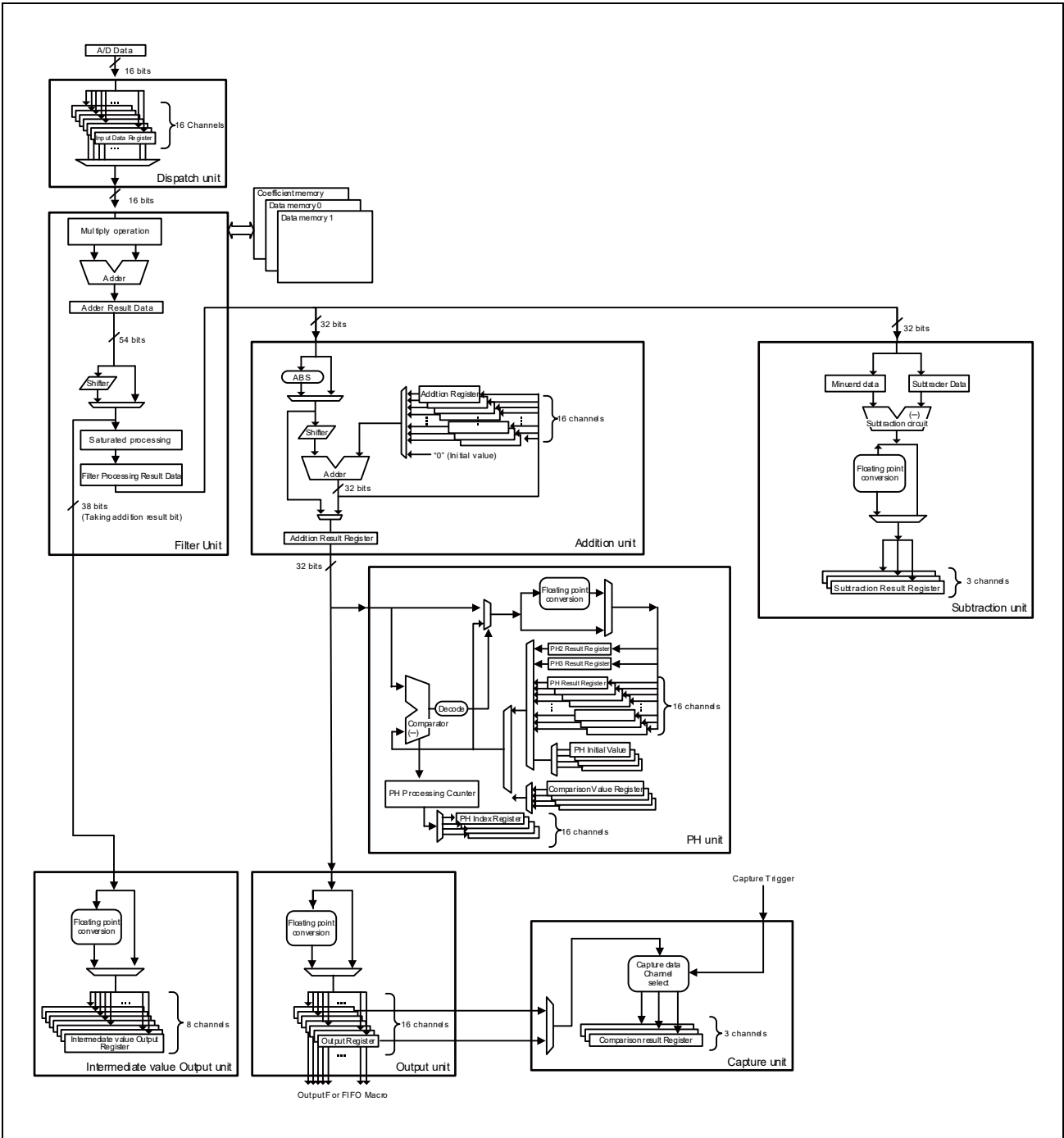


Figure 37.23 Data Flow of the DFE Macro

37.5.1.2 FIFO Macro

The data flow and the block diagram of the FIFO macro is shown below.

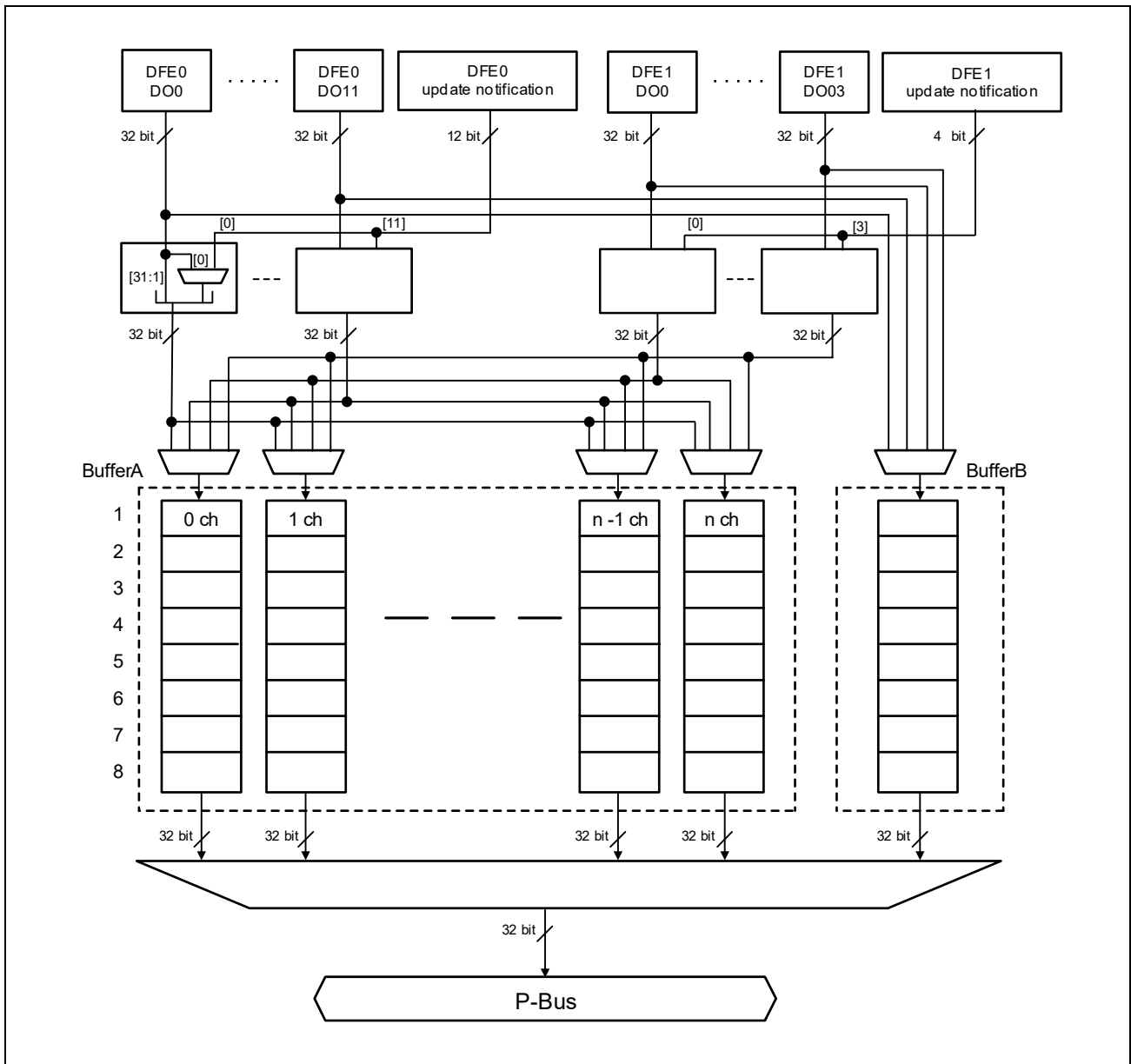


Figure 37.24 Data Flow of the FIFO Macro

37.5.2 Address Map

37.5.2.1 DFE Macro

Figure 37.25 shows an address map of the DFE.

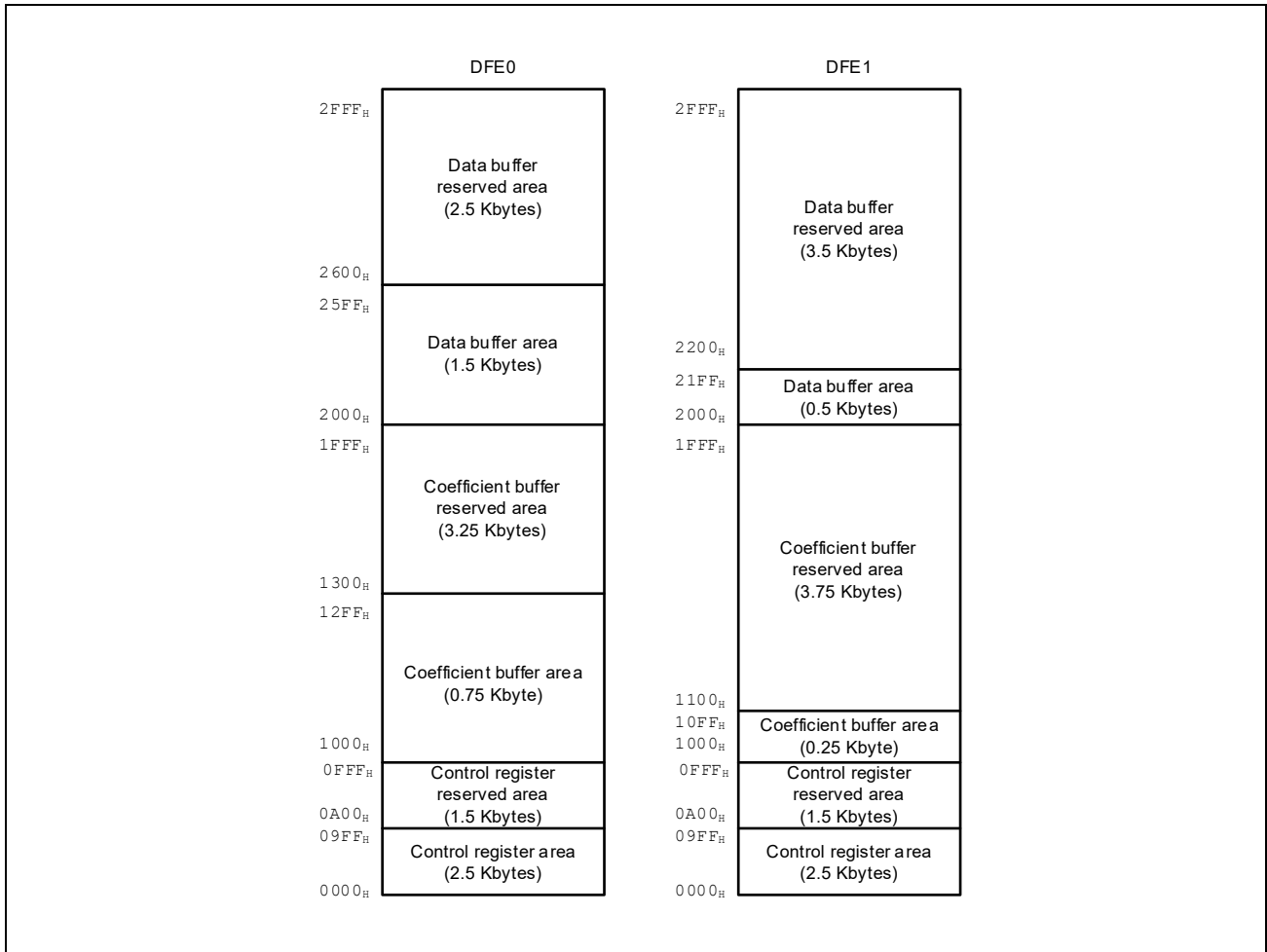


Figure 37.25 Memory Map of the DFE Macro

37.5.2.2 FIFO Macro

An address map of FIFO macro is shown below.

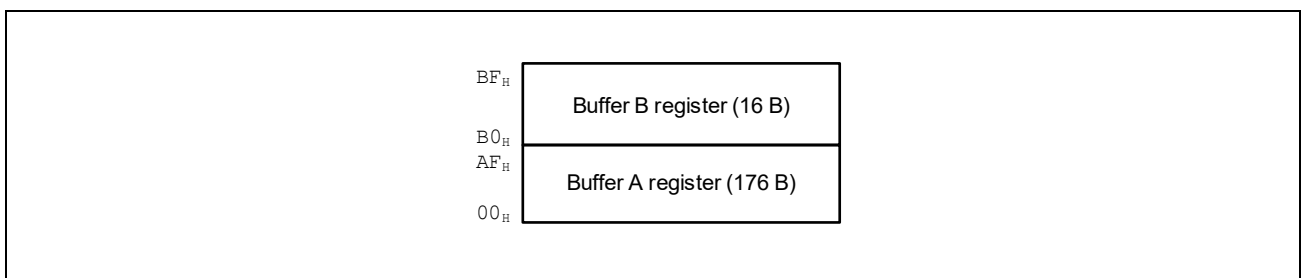


Figure 37.26 Register Map of the FIFO Macro

37.5.3 Memory

Coefficients required for filter processing are stored in coefficient memory (CMEM). Data memory holds the input values for the FIR, and the initial and intermediate values for the IIR. Each channel has its own area. When the FIR 64-TAP is selected on an even-numbered channel (n), the coefficient memory is allocated as shown in the left part of the figure below. Although data memory areas are available to the next odd-numbered channel (n + 1), coefficient memory areas are not available; therefore, use of the next odd-numbered channel (n + 1) is prohibited.

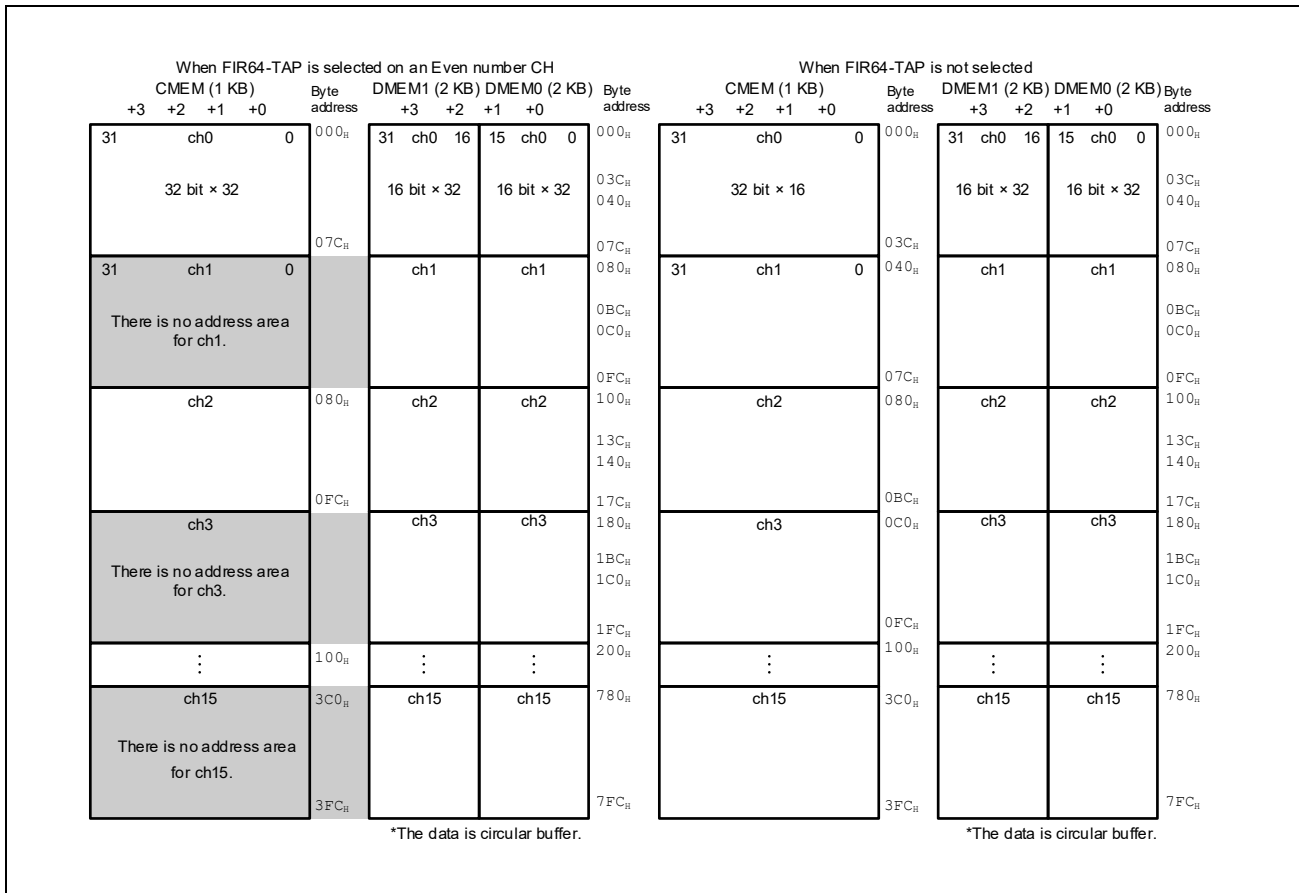


Figure 37.27 Usage of Memory (with and without the Selection of a 64-Tap FIR Filter)

37.5.4 Format of Coefficient Memory

Figure 37.28 shows the data format of the coefficient memory.

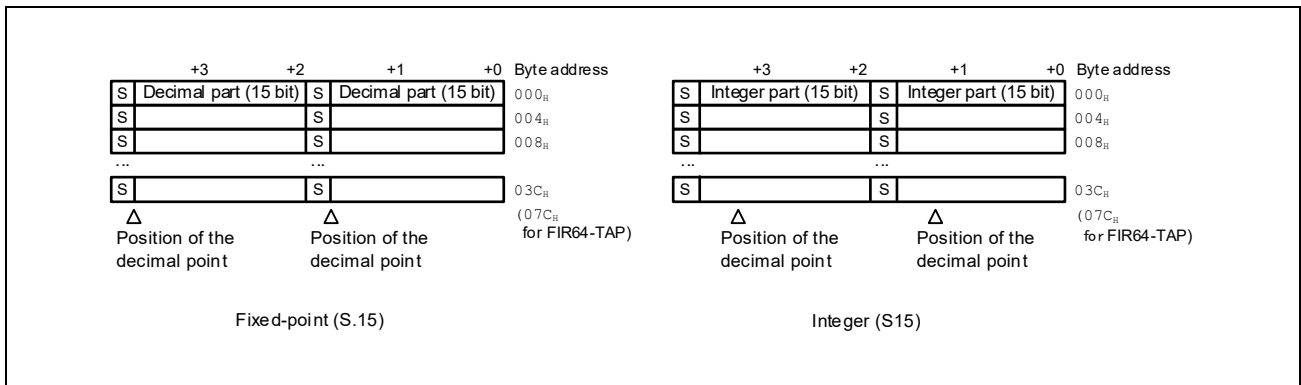


Figure 37.28 Decimal Point Positions in the Coefficient Memory (when the FIR is selected)

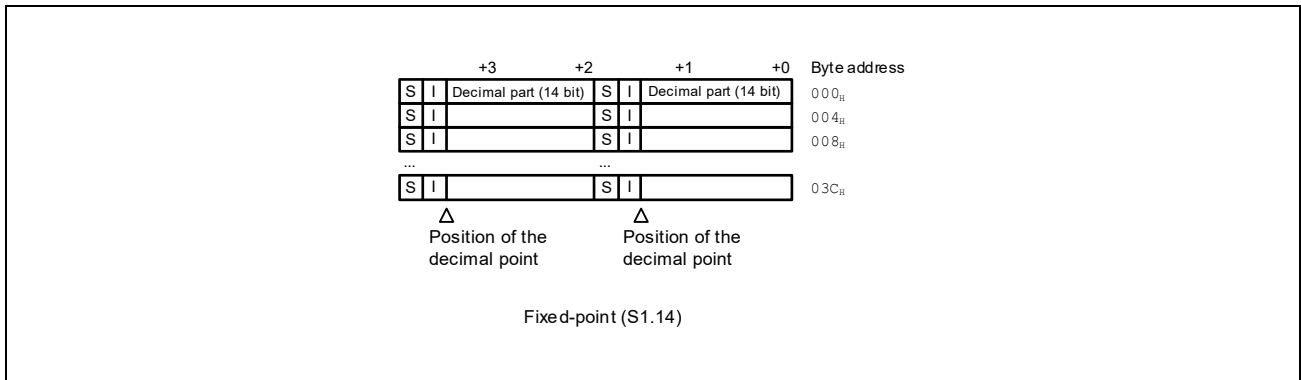


Figure 37.29 Decimal Point Positions in the Coefficient Memory (when the IIR is selected)

37.5.5 Format of Data Memory

Figure 37.30 shows decimal point positions of data in the data memory.

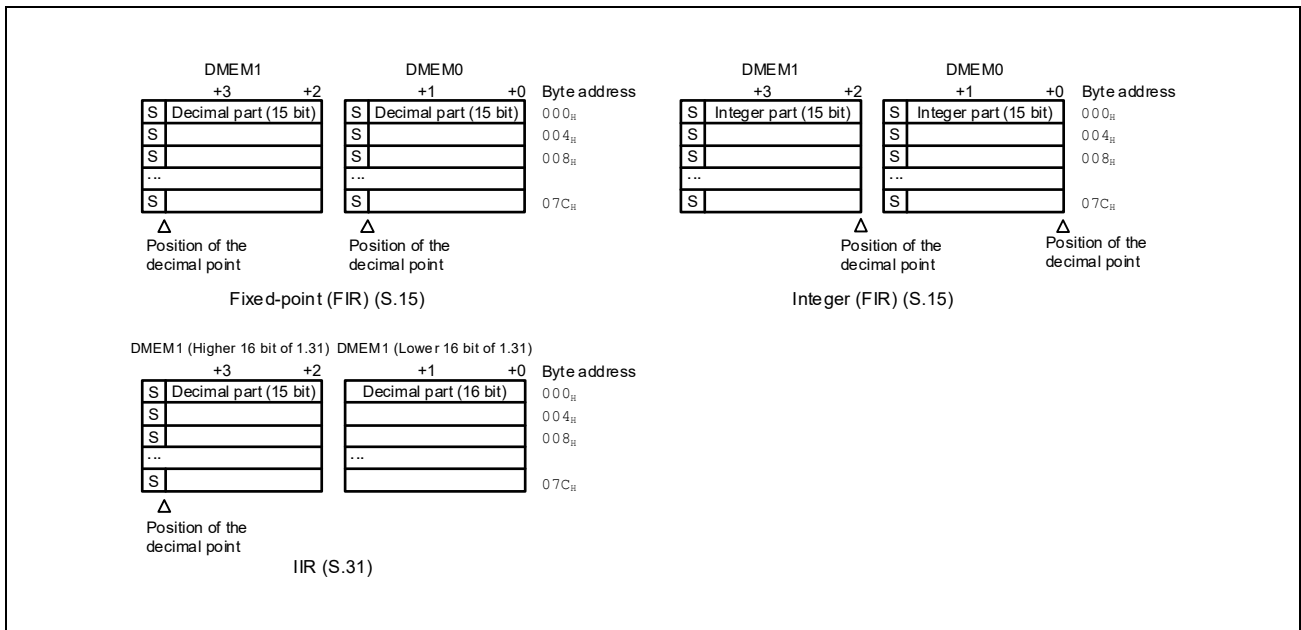


Figure 37.30 Decimal Point Positions in the Data Memory

37.5.6 Setting Control Registers

37.5.6.1 FIR (Fixed-Point)

The following table shows combinations of calculations and cascades that are selectable in the control register A (DFEjCTLACHn) and the control register B (DFEjCTLBCHn) when the FIR and fixed-point format are specified in the control register A. Don't use the cascade function in channels for which accumulation is selected.

Table 37.75 FIR (Fixed-Point)

Input Format	Filter	Accumulation or Decimation	Peak-Hold or Comparison	Floating-Point Conversion	Execution (Result)	Cascade (Output)*1
Fixed-point (S1.15)	FIR	Accumulation	Peak-hold	Conversion specified	OK (floating-point)	NG
				Conversion not specified	OK (S9.22)	NG
			Comparison	Conversion specified	OK (floating-point)	NG
				Conversion not specified	OK (S9.22)	NG
			Not specified	Conversion specified	OK (floating-point)	NG
				Conversion not specified	OK (S9.22)	NG
		Decimation	Peak-hold	Conversion specified	OK (floating-point)	OK (S.15)
				Conversion not specified	OK (S.31)	OK (S.15)
			Comparison	Conversion specified	OK (floating-point)	OK (S.15)
				Conversion not specified	OK (S.31)	OK (S.15)
			Not specified	Conversion specified	OK (floating-point)	OK (S.15)
				Conversion not specified	OK (S.31)	OK (S.15)
		Not specified	Peak-hold	Conversion specified	OK (floating-point)	OK (S.15)
				Conversion not specified	OK (S.31)	OK (S.15)
			Comparison	Conversion specified	OK (floating-point)	OK (S.15)
				Conversion not specified	OK (S.31)	OK (S.15)
			Not specified	Conversion specified	OK (floating-point)	OK (S.15)
				Conversion not specified	OK (S.31)	OK (S.15)

Note 1. This setting is prohibited for NG combinations. For OK combinations, set fixed-point as the input data format of the cascade destination channels.

37.5.6.2 FIR (Integer)

The following table shows combinations of calculations and cascades that are selectable in the control register A (DFEjCTLACHn) and the control register B (DFEjCTLBCHn) when the FIR and integer format are specified in the control register A. When integer is specified, only the FIR filter processing can be executed. Do not specify floating-point conversion.

Table 37.76 FIR (Integer)

Input Format	Filter	Accumulation or Decimation	Peak-Hold or Comparison	Floating-Point Conversion	Execution (Result)* ¹	Cascade (Output)* ²	
Integer (S15)	FIR	Accumulation	Peak-hold	Conversion specified	NG	NG	
				Conversion not specified	NG	NG	
				Comparison	Conversion specified	NG	NG
					Conversion not specified	NG	NG
				Not specified	Conversion specified	NG	NG
					Conversion not specified	NG	NG
			Decimation	Peak-hold	Conversion specified	NG	OK (S15)
					Conversion not specified	OK (S31)	OK (S15)
				Comparison	Conversion specified	NG	OK (S15)
					Conversion not specified	OK (S31)	OK (S15)
				Not specified	Conversion specified	NG	OK (S15)
					Conversion not specified	OK (S31)	OK (S15)
		Not specified	Peak-hold	Conversion specified	NG	OK (S15)	
				Conversion not specified	OK (S31)	OK (S15)	
				Comparison	Conversion specified	NG	OK (S15)
					Conversion not specified	OK (S31)	OK (S15)
			Not specified	Conversion specified	NG	OK (S15)	
				Conversion not specified	OK (S31)	OK (S15)	

Note 1. The setting is prohibited for NG combinations.

Note 2. The setting is prohibited for NG combinations. For OK combinations, set integer as the input data format of the cascade destination channels.

37.5.6.3 IIR (Fixed-Point)

The following table shows combinations of calculations and cascades that are selectable in the control register A (DFEjCTLACHn) and the control register B (DFEjCTLBCHn) when the IIR is specified in the control register A. When the IIR is specified, only fixed-point format filter processing can be executed. When accumulation is specified, specification of the cascade function is prohibited.

Table 37.77 IIR

Input Format	Filter	Accumulation or Decimation	Peak-Hold or Comparison	Floating-Point Conversion	Execution (Result)	Cascade (Output)*1
Fixed-point (S.15)	IIR	Accumulation	Peak-hold	Conversion specified	OK (floating-point)	NG
				Conversion not specified	OK (S9.22)	NG
			Comparison	Conversion specified	OK (floating-point)	NG
				Conversion not specified	OK (S9.22)	NG
			Not specified	Conversion specified	OK (floating-point)	NG
				Conversion not specified	OK (S9.22)	NG
		Decimation	Peak-hold	Conversion specified	OK (floating-point)	OK (S.15)
				Conversion not specified	OK (S.31)	OK (S.15)
			Comparison	Conversion specified	OK (floating-point)	OK (S.15)
				Conversion not specified	OK (S.31)	OK (S.15)
			Not specified	Conversion specified	OK (floating-point)	OK (S.15)
				Conversion not specified	OK (S.31)	OK (S.15)
		Not specified	Peak-hold	Conversion specified	OK (floating-point)	OK (S.15)
				Conversion not specified	OK (S.31)	OK (S.15)
			Comparison	Conversion specified	OK (floating-point)	OK (S.15)
				Conversion not specified	OK (S.31)	OK (S.15)
			Not specified	Conversion specified	OK (floating-point)	OK (S.15)
				Conversion not specified	OK (S.31)	OK (S.15)

Note 1. The setting is prohibited for NG combinations. For OK combinations, set fixed-point as the input data format of the cascade destination channels.

37.5.7 Trigger Control

37.5.7.1 Timer Trigger Input

The digital filter uses compare An ($n = 0$ to 19) and compare Bn ($n = 0$ to 19), for a total of 40 signals as timer triggers through the synchronizer.

To compare An, five timer trigger flags (accumulation/decimation initialization flag, PH initialization flag, PH mask start flag, filter initialization flag, and subtraction starting flag) can be assigned.

To compare Bn, four timer trigger flags (accumulation/decimation disable flag, PH end flag, PH mask end flag, and subtraction end flag) are paired with the timer trigger functions selected for compare An are automatically assigned.

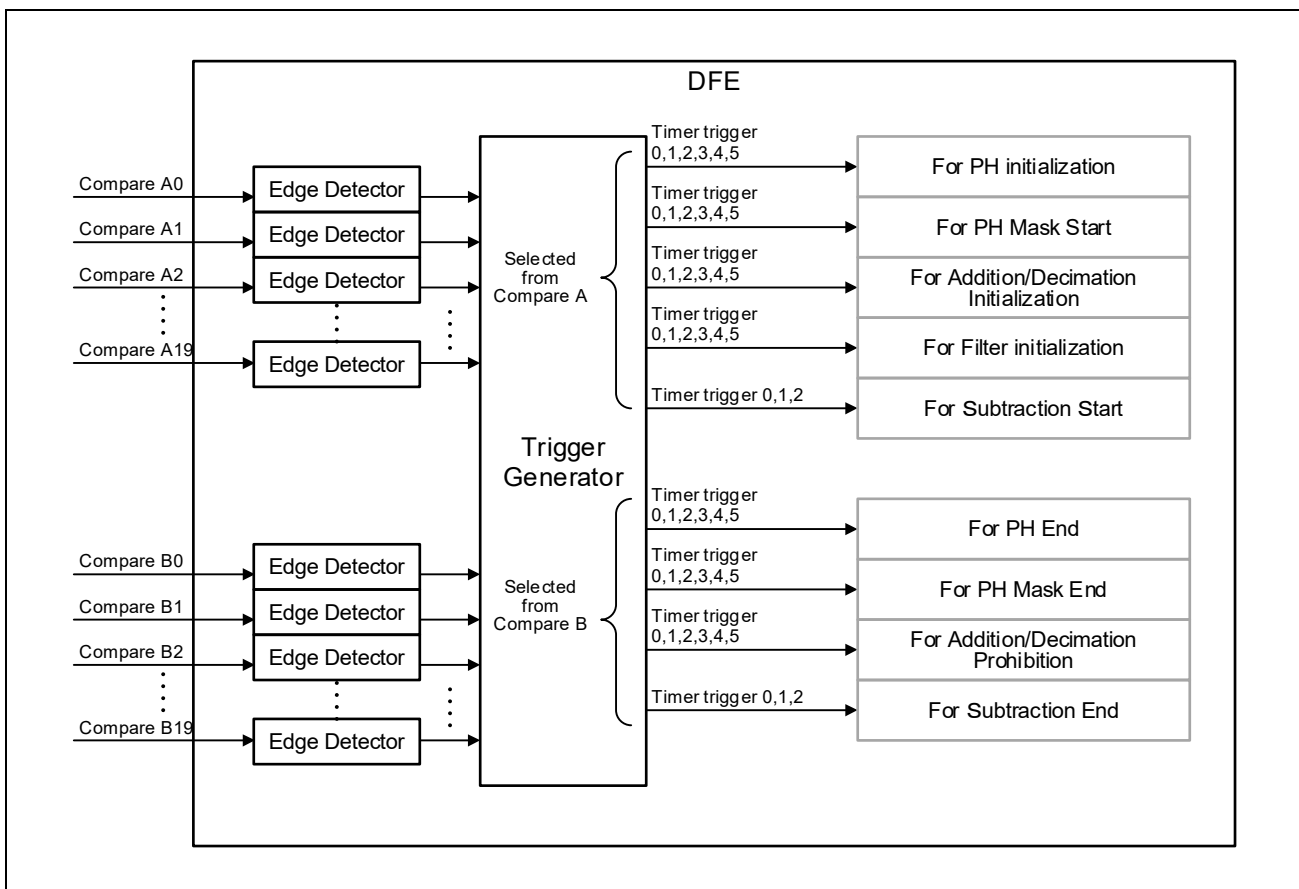


Figure 37.31 Timer Trigger Input

To compare Bn, the signal with the same number as the compare An selected for each flag is automatically assigned as the timer trigger for the corresponding trigger flag.

The table below shows the relationships between the trigger flags assigned to compare An and compare Bn, and the number of timer triggers available for each trigger flag.

Flag Corresponding to Compare An	Flag Corresponding to Compare Bn	Number of Selectable Timer Triggers
Accumulation/decimation initialization flag	Accumulation/decimation disable flag	6 (Timer Trigger 0-5)
Peak-Hold initialization flag	Peak-Hold end flag	6 (Timer Trigger 0-5)
Peak-Hold mask start flag	Peak-Hold mask end flag	6 (Timer Trigger 0-5)
Filter initialization flag	—	6 (Timer Trigger 0-5)
Subtraction start flag	Subtraction end flag	3 (Timer Trigger 0-2)

Compare Bn signals with the same number as the compare An signals that are selected for the accumulation/decimation initialization flag are automatically allocated to the accumulation/decimation disable flag timer triggers.

(1) Setting of Accumulation/Decimation Initialization/Disable Flag

For the accumulation/decimation initialization flag and accumulation/decimation disable flag, use of timer triggers 0 to 5 or timer triggers 0 to 3 is enabled with the settings shown below. It is recommended to use timer triggers in the mode where timer triggers 0 to 5 can be used.

(A) Settings for the Mode Where Timer Triggers 0 to 5 are Used

Bit Name	Setting	Description
DFEJTRGCHn.AFE	01 _B	Assigns the accumulation/decimation disabling flag to a timer flag.
DFEJTRGCHn.AE	11 _B	Selects the mode where timer triggers 0 to 5 are used.
DFEJTMTRGCHn.AT	000 _B to 101 _B	Selects a timer trigger to be used from timer triggers 0 to 5.
DFEjMITRG.MITMTRGn (n = 0,1,2,3)	00000 _B to 10011 _B	Selects a compare An to be assigned to timer trigger n. (Set timer triggers 0 to 3.)
DFEjMITRG1.MITMTRGn (n = 4,5)	00000 _B to 10011 _B	Selects a compare An to be assigned to timer trigger n. (Sets timer triggers 4 and 5.)

(B) Settings for the Mode Where Timer Triggers 0 to 3 are Used

Bit Name	Setting	Description
DFEJTRGCHn.AFE	01 _B	Assigns the accumulation/decimation disabling flag to a timer flag.
DFEJTRGCHn.AE	01 _B	Selects the mode where timer triggers 0 to 3 are used.
DFEJTRGCHn.AT	00 _B to 11 _B	Selects a timer trigger to be used from timer triggers 0 to 3.
DFEjMITRG.MITMTRGn (n = 0,1,2,3)	00000 _B to 10011 _B	Selects a compare An to be assigned to timer trigger n.

(2) Setting of Peak-Hold Initialization/End Flag

For the Peak-Hold initialization flag and Peak-Hold end flag, use of timer triggers 0 to 5 or timer triggers 0 to 3 is enabled with the settings shown below. It is recommended to use timer triggers in the mode where timer triggers 0 to 5 can be used.

(A) Settings for the Mode Where Timer Triggers 0 to 5 are Used

Bit Name	Setting	Description
DfEjTRGCHn.PFE	01 _B	Assigns the Peak-Hold end flag to a timer flag.
DfEjTRGCHn.PE	11 _B	Selects the mode where timer triggers 0 to 5 are used.
DfEjTMTRGCHn.PT	000 _B to 101 _B	Selects a timer trigger to be used from timer triggers 0 to 5.
DfEjPITRG.PITMTRGn (n = 0,1,2,3)	00000 _B to 10011 _B	Selects a compare An to be assigned to timer trigger n.
DfEjPITRG1.PITMTRGn (n = 4,5)	00000 _B to 10011 _B	(Sets timer triggers 0 to 3.)

(B) Settings for the Mode Where Timer Triggers 0 to 3 are Used

Bit Name	Setting	Description
DfEjTRGCHn.PFE	01 _B	Assigns the Peak-Hold end flag to a timer flag.
DfEjTRGCHn.PE	01 _B	Selects the mode where timer triggers 0 to 3 are used.
DfEjTRGCHn.PT	00 _B to 11 _B	Selects a timer trigger to be used from timer triggers 0 to 3.
DfEjPITRG.PITMTRGn (n = 0,1,2,3)	00000 _B to 10011 _B	Selects a compare An to be assigned to timer trigger n. (Sets timer triggers 0 to 3.)

(3) Setting of Filter Initialization Flag

For the filter initialization flag, use of timer triggers 0 to 5 or timer triggers 0 to 3 is enabled with the settings shown below. It is recommended to use timer triggers in the mode where timer triggers 0 to 5 can be used.

(A) Settings for the Mode Where Timer Triggers 0 to 5 are Used

Bit Name	Setting	Description
DfEjTRGCHn.FE	11 _B	Selects the mode where timer triggers 0 to 5 are used.
DfEjTMTRGCHn.FT	000 _B to 101 _B	Selects a timer trigger to be used from timer triggers 0 to 5.
DfEjFITRG.FITMTRGn (n = 0,1,2,3)	00000 _B to 10011 _B	Selects a compare An to be assigned to timer trigger n. (Sets timer triggers 0 to 3.)
DfEjFITRG1.FITMTRGn (n = 4,5)	00000 _B to 10011 _B	Selects a compare An to be assigned to timer trigger n. (Sets timer triggers 4 and 5.)

(B) Settings for the Mode Where Timer Triggers 0 to 3 are Used

Bit Name	Setting	Description
DfEjTRGCHn.FE	01 _B	Selects the mode where timer triggers 0 to 3 are used.
DfEjTRGCHn.FT	00 _B to 11 _B	Selects a timer trigger to be used from timer triggers 0 to 3.
DfEjFITRG.FITMTRGn (n = 0,1,2,3)	00000 _B to 10011 _B	Selects a compare An to be assigned to timer trigger n. (Sets timer triggers 0 to 3.)

(4) Setting of Peak-Hold Mask Start/End Flag

For the Peak-Hold mask start flag and Peak-Hold mask end flag, use of timer triggers 0 to 5 is enabled with the settings shown below.

Bit Name	Setting	Description
DFEJTRGCHn.PMFE	01 _B	Assigns the PH mask ending flag to a timer flag.
DFEJTRGCHn.PME	01 _B	Selects the mode where timer triggers are used.
DFEJTMTRGCHn.PMT	000 _B to 101 _B	Selects a timer trigger to be used from timer triggers 0 to 5.
DFEJPMITRG0.PMITMTRGn (n = 0,1,2,3)	00000 _B to 10011 _B	Selects a compare An to be assigned to timer trigger n. (Sets timer triggers 0 to 3.)
DFEJPMITRG1.PMITMTRGn (n = 4,5)	00000 _B to 1001 _B	Selects a compare An to be assigned to timer trigger n. (Sets timer triggers 4 and 5.)

(5) Setting of Subtraction Start/End Flag

For the subtraction start flag and subtraction end flag, use of timer triggers 0 to 2 is enabled with the settings shown below.

Bit Name	Setting	Description
DFEJSUBTRGCHn.SBFE	01 _B	Assigns the subtraction end flag to a timer flag.
DFEJSUBTRGCHn.SBE	01 _B	Selects the mode where timer triggers are used.
DFEJSUBTRGCHn.SBT	00 _B to 10 _B	Selects a timer trigger to be used from timer triggers 0 to 2.
DFEJSUBTRG0.SUBTMTRGn (n = 0,1,2)	00000 _B to 1001 _B	Selects a compare An to be assigned to timer trigger n. (Sets timer triggers 0 to 2.)

37.5.7.2 Trigger Flag Functions

When the channel enable bit (DFEjCTRLACHn.EN) is 1, the DFE can generate flags of the following ten functions for each channel by using software triggers, timer triggers that are allocated to the trigger functions, and the trigger setting register (DFEjTRGCHn). (These flags are referred to as trigger flags hereafter.)

The accumulation/decimation processing is set with DFEjCTLBCHn.PRCSA, and the peak-hold processing is set with DFEjCTLBCHn.PRCB.

Subtraction is set with DFEjSUBCTLCHn.SEN.

Whether accumulation/decimation processing is currently disabled or enabled can be read from DFEjCTLBn.DISA. Similarly, whether peak-hold processing is currently disabled or enabled can be read from DFEjCTLBn.DISB.

Both DFEjCTLBn.DISA and DFEjCTLBn.DISB can be set before the DFE starts processing (channel is disabled (DFEjCTRLACHn.EN = 0) and there is no data to be processed (DFEjSTCHn.VALID = 0)).

For details, see **Section 37.3.4, DFEjCTLBCHn — Control Register B n (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1)**.

Table 37.78 Trigger Flags (1/2)

Flag Name	Description
Accumulation/decimation initialization flag	When the accumulation/decimation initialization flag is enabled by timer triggers or software triggers specified for each channel while accumulation/decimation processing is in progress or in the enabled or disabled state, the DFE initializes the counter of the accumulation circuit and executes accumulation/decimation processing for the subsequent input data.
Accumulation/decimation disable flag	When the accumulation/decimation disable flag is enabled by timer triggers or software triggers specified for each channel while accumulation/decimation processing is in progress or in the enabled state, the DFE disables accumulation/decimation processing for the subsequent input data. To enable accumulation/decimation processing again, enable the accumulation/decimation initialization flag.
Peak-hold initialization flag	When the peak-hold initialization flag is enabled by timer triggers or software triggers specified for each channel while peak-hold processing is in progress or in the enabled or disabled state, the DFE compares the next input data with the peak-hold initialization register value, not with the peak-hold processing result as before, and after that the DFE compares the subsequent data with the peak-hold processing result.
Peak-hold end flag	When the peak-hold end flag is enabled by timer triggers or software triggers specified for each channel while peak-hold processing is in progress or in the enabled state, the DFE disables peak-hold processing for the subsequent input data. The peak-hold end interrupt can be output when the peak-hold end flag is enabled. To enable peak-hold processing again, enable the peak-hold initialization flag.
Filter initialization flag	When the filter initialization flag is enabled by timer triggers or software triggers specified for each channel, the DFE initializes the filter processing address pointer and executes FIR/IIR processing from the initial value.

CAUTION

When timer triggers are used for the accumulation/decimation disable flag or peak-hold end flag, timer triggers 0 to 5 are automatically allocated as described in **Section 37.5.7.1, Timer Trigger Input**.

Table 37.78 Trigger Flags (2/2)

Flag Name	Description
Peak-Hold mask start flag	In any of the states where Peak-Hold processing is in progress, processing is enabled, or processing is disabled, if the Peak-Hold mask starting flag is enabled by the timer trigger or software trigger input set for each channel, Peak-Hold processing is halted on the subsequent input data. To restart Peak-Hold processing, enable the Peak-Hold mask end flag.
Peak-Hold mask end flag	In any of the states where Peak-Hold processing is in progress, processing is enabled, or processing is disabled, if the Peak-Hold mask ending flag is enabled by the timer trigger or software trigger input set for each channel, Peak-Hold processing is restarted on the subsequent input data.
Subtraction start flag	If the subtraction start flag is enabled by the timer trigger or software trigger input set for each channel, subtraction is initialized and started on the subsequent input data.
Subtraction end flag	If the subtraction end flag is enabled by the timer trigger or software trigger input set for each channel, subtraction is disabled on the subsequent input data. To enable subtraction again, enable the subtraction starting flag.
Automatic initialization flag	<p>If the automatic initialization flag is enabled by writing the channel enable bit (0 → 1), the processing below is executed on the subsequent input data.</p> <ul style="list-style-type: none"> • The target register is initialized when filtering process is started. • After the address pointer for filtering process is initialized, FIR/IIR processing is executed from the initial values (same as the function of the filter initialization flag). • The counter of the accumulation circuit is initialized. • Peak-Hold processing can be started from the state where the input data is compared to the value of the Peak-Hold initial value setting register.

CAUTION

If timer triggers are used for the Peak-Hold mask end flag and subtraction end flag, timer triggers are automatically assigned, as described in **Section 37.5.7.1, Timer Trigger Input**.

Timer triggers and software triggers can be allocated to trigger flags for each channel.

However, the timer triggers for the accumulation/decimation disable flag, Peak-Hold end flag, and Peak-Hold mask end flag are automatically assigned. Detailed operations of each trigger flag are described in the descriptions of the accumulation circuit, Peak-Hold circuit, and filtering circuit.

Table 37.79 Example of Trigger Flag Settings

	Accumulation/ Decimation Initialization Flag	Accumulation/ Decimation/ Disable Flag	Peak-Hold Initialization Flag	Peak-Hold Disable Flag	Filter Initialization Flag	Peak-Hold Mask Start Flag	Peak-Hold Mask End Flag
Channel 0	Timer trigger 0 (compare A0)	Compare B0 is automatically allocated according to the timer trigger selection.	None	None	Timer trigger 3	None	None
Channel 1	None	None	Timer trigger 2 (compare A9)	Compare B9 is automatically allocated according to the timer trigger selection.	Software trigger	Timer trigger 4 (compare A15)	Compare B15 is automatically allocated according to the timer trigger selection
...
Channel 9	Timer trigger 3 (compare A5)	Compare B5 is automatically allocated according to the timer trigger selection.	None	None	None
Channel 10	Timer trigger 1	Software trigger	None	None	Timer trigger 0	None	None
...
Channel 15	None	None	Timer trigger 4 (compare A4)	Compare B4 is automatically allocated according to the timer trigger selection	None	None	None

Timer triggers and software triggers can be assigned to trigger flags for each subtraction channel. However, the timer triggers for the subtraction end flag are automatically assigned. Detailed operations of the trigger flags are described in the descriptions of the subtraction circuit.

Table 37.80 Example of Subtraction Trigger Flag Settings

	Subtraction Starting Flag	Subtraction Ending Flag
0SCH	Timer trigger 0 (compare A16)	Compare B16 is automatically assigned when a timer trigger is selected.
1SCH	Timer trigger 1 (compare A17)	Compare B17 is automatically assigned when a timer trigger is selected.
2SCH	Timer trigger 2 (compare A18)	Software trigger

Trigger flags are valid for data that is input after a software trigger or timer trigger is generated.

If timer trigger 0 is set for the accumulation initialization flag in channel 0, for example, accumulation processing is initialized for the first input data in channel 0 after a timer trigger is generated.

To process the same data on multiple channels, the trigger flag settings should be the same to keep consistency in the processed data.

For example, when data input from AD0 is specified for channels 0 and 1 and the accumulation initialization flag is used, the same accumulation initialization flag timer trigger should be set for both channels 0 and 1.

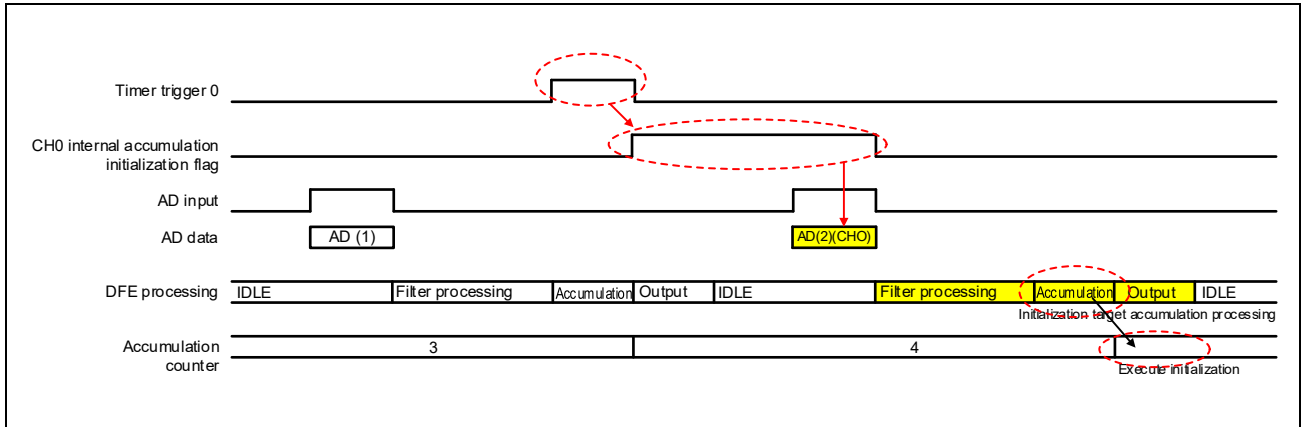


Figure 37.32 Example of Trigger Flag Operation

37.5.7.3 Trigger Flag Functions and Processing

The accumulation circuit is set with DFEjCTLBCHn.PRCSA.

The Peak-Hold circuit is set with DFEjCTLBCHn.PRCSB and DFEjPH23CCTL0.PEN.

Table 37.81 shows the register settings for the accumulation circuit and Peak-Hold circuit, and the relationships between the output data and interrupts according to the combinations of the trigger flag functions.

Table 37.81 Trigger Flag Functions and Processing

Accumulation Circuit		PH Circuit		Input Data Processing after Trigger Flag is Enabled					
Register Settings	Trigger Flag Functions	Register Settings	Trigger Flag Functions	Output Data Write	Output Data Interrupt	Condition Match Interrupt	Peak-Hold End Interrupt	Peak-Hold Update Notification	
PRCSA = 00 _B (Neither accumulation nor decimation executed)	Both accumulation/decimation initialization flag and disable flag disabled	PH/comparison disabled	Both PH initialization flag and disable flag disabled	Provided	Not provided*2	Not provided	Not provided	Not provided	
		Comparison executed exclusively		Provided	Not provided*2	Provided	Not provided	Not provided	
		PH executed exclusively	PH enabled	Provided	Not provided*2	Not provided	Not provided	Not provided	Provided
			PH ended	Provided	Not provided*2	Not provided*1	Not provided	Not provided	Not provided
		PH and comparison executed simultaneously	PH enabled	Provided	Not provided*2	Provided	Not provided	Not provided	Provided
			PH ended	Provided	Not provided*2	Provided	Provided	Not provided	Not provided
		PH23 executed	PH enabled	Provided	Not provided*2	Provided	Not provided	Not provided	Provided
PH ended	Provided		Not provided*2	Not provided	Provided	Not provided	Not provided		
PRCSA = 01 _B or 10 _B (Either accumulation or decimation executed)	Accumulation/decimation enabled	PH/comparison disabled	Both PH initialization flag and disable flag disabled	Provided	Provided	Not provided	Not provided	Not provided	
		Comparison executed exclusively		Provided	Provided	Provided	Not provided	Not provided	
		PH executed exclusively	PH enabled	Provided	Provided	Not provided	Not provided	Not provided	Provided
			PH ended	Provided	Provided	Not provided*1	Not provided	Not provided	Not provided
		PH and comparison executed simultaneously	PH enabled	Provided	Provided	Provided	Not provided	Not provided	Provided
			PH ended	Provided	Provided	Provided	Provided	Not provided	Not provided
		PH23 executed	PH enabled	Provided	Provided	Provided	Not provided	Not provided	Provided
PH ended	Provided		Provided	Not provided	Provided	Not provided	Not provided		
PRCSA = 01 _B or 10 _B (Either accumulation or decimation executed)	Accumulation/decimation disabled	PH/comparison disabled	Both PH initialization flag and disable flag disabled	Not provided	Not provided	Not provided	Not provided	Not provided	
		Comparison executed exclusively		Not provided	Not provided	Not provided	Not provided	Not provided	
		PH executed exclusively		Not provided	Not provided	Not provided	Not provided	Not provided	
		PH and comparison executed simultaneously		Not provided	Not provided	Not provided	Not provided	Not provided	
		PH23 executed		Not provided	Not provided	Not provided	Not provided	Not provided	

Note 1. The interrupt is output when the peak-hold end flag is enabled.

Note 2. Refer to the Output Circuit Register.

When accumulation/decimation processing and peak-hold/comparison processing are used in the same channel with the accumulation/decimation count specified, peak-hold processing is executed on each specified count.

If accumulation/decimation processing is disabled by the trigger flag function, peak-hold/comparison processing is not executed, either. Do not input software triggers or timer triggers for the trigger flag function for peak-hold initialization, peak-hold end trigger flag, peak-hold mask start or peak-hold mask end trigger flag functions to a channel with accumulation/decimation processing disabled.

37.5.7.4 Automatic Initialization of the Channel

When the automatic initialization enable bit (DFEjCTLACHn.AIME) is 1, automatic initialization of the channels is available. When the channel enable bit (DFEjCTLACHn.EN) is modified from 0 to 1, the automatic initialization flag of the channel is enabled.

The automatic initialization flag of the channel is also enabled when 1 is simultaneously written to the automatic initialization enable bit and the channel enable bit.

When the automatic initialization flag of the channel is enabled, the address pointer of the filtering process of the next input data is initialized, and then FIR/IIR processing with initialization is executed.

Also, the registers are initialized simultaneously with the filtering process, and after the filtering process, the accumulation circuit and PH circuit are returned to the initial state. But the accumulation circuit and the peak-hold circuit is not set valid by the automatic initialization flag.

The registers below are initialized.

Register Name		Function
Output data register	DFEjDOCHn	Cleared when the channel processing is started.
Intermediate values output registers L	DFEjHOLCHn	Same as above
Intermediate values output registers H	DFEjHOHCHn	Same as above
Intermediate values output mirror registers L	DFEjHOLMCHn	Same as above
Peak-Hold result register	DFEjPHCHn	Same as above
Peak-Hold index register	DFEjPHINDCHn	Same as above
Status register Status Register PH23 processing status	DFEjSTCHn.PH23ST	Same as above
Peak-Hold 2 Result Register 0*1	DFEjPH20	Same as above
Peak-Hold 3 Result Register 0*1	DFEjPH30	Same as above
Peak-Hold 2 index register*1	DFEjPH2IND0	Same as above
Peak-Hold 3 index register*1	DFEjPH3IND0	Same as above

Note 1. Cleared when the PH23 function is enabled (DFEjPH23CCTL0.PEN = 1) and the PH23 function is enabled for the pertinent channel (the pertinent channel number is set with DFEjPH23CCTL0.CHS).

When the channel enable bit (DFEjCTLACHn.EN) is modified from 0 to 1, the automatic initialization flag of the channel is enabled. When the automatic initialization flag of the channel is enabled and processing of the next channel is started, FIR or IIR processing with initialization is executed.

Also, the pertinent registers are initialized simultaneously with the start of the filtering process.

After the filtering process, the accumulator and counter for accumulation in the accumulation circuit are initialized even if accumulation/decimation processing is disabled.

After the accumulation process, the Peak-Hold counter in the PH circuit is initialized even if the PH processing has been finished. Also, the PH circuit retains the state in which comparison is possible between the data that was input to the PH circuit during the PH processing and the value of the PH initial value register.

After that, in the PH processing of the subsequent filtering process after PH processing is enabled, the data that was input to the PH circuit is compared to the value of the PH initial value register. Subsequently, in PH processing, the data that was input to the PH circuit is compared to the value of the PH result register.

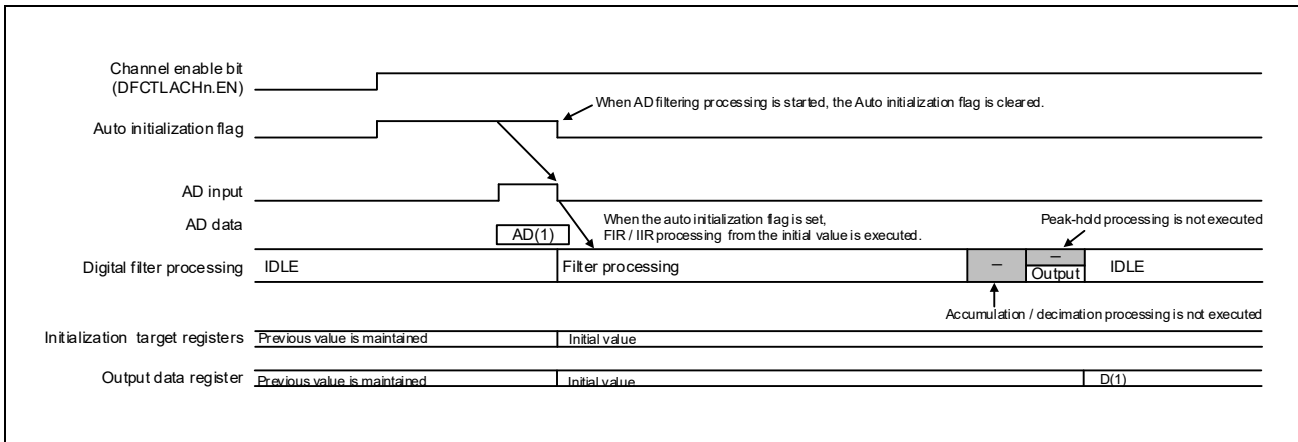


Figure 37.33 Operations of the Channel Automatic Initialization Flag

After the automatic initialization flag of the channel is enabled, the automatic initialization flag is maintained until the filtering process of the channel is requested. However, the automatic initialization flag is cleared if the channel enable bit becomes 0. If 1 is written again to the channel enable bit with the automatic initialization enable bit (DFEjCTLACHn.AIME) being 1, the automatic initialization flag is enabled.

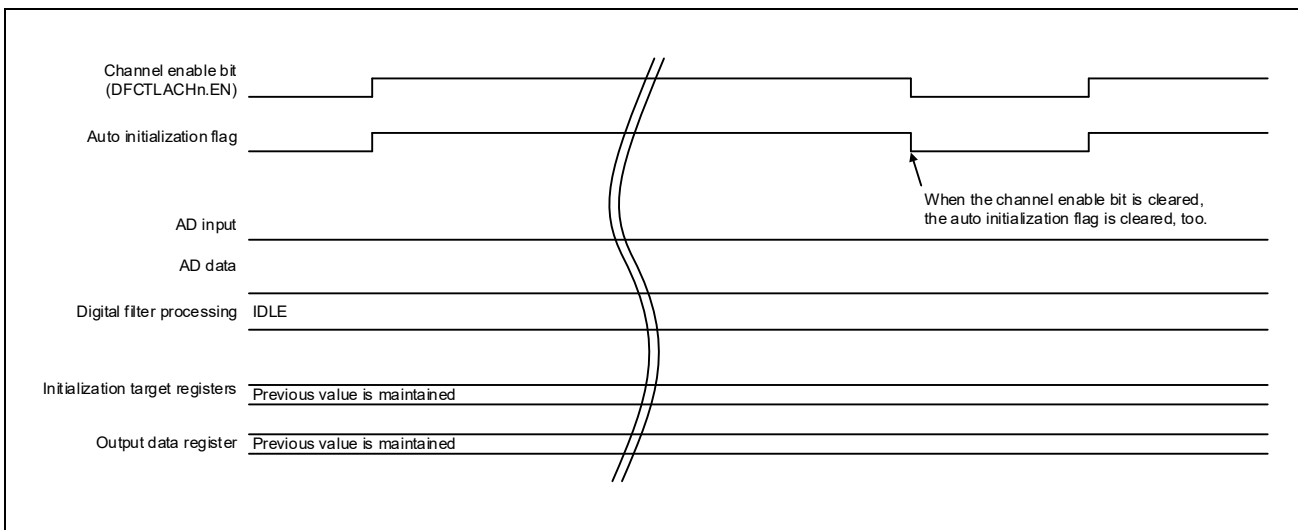


Figure 37.34 Clear Channel Automatic Initialization Flag

37.5.8 Input Data Control

This macro accepts the input of the data listed in the table below.

Table 37.82 Classification of Input Data

Classification	Input Data
AD input	SAR-AD
	Delta-Sigma AD
	Cyclic-AD
Cascade Input	Filter Result Data
	Subtraction Result Data
Software Input	Software Input Data (DFEjDI.DI)

37.5.8.1 AD Input

The digital filter compares the TAG value (called the channel tag hereafter) set in the TAG bits in DFEjCTLACHn for each channel and the TAG input from the AD, and executes processing in the channel that has the matched tag value. This function allows single piece of AD data to be input to processing on multiple channels.

37.5.8.2 Cascade Input

The digital filter has a cascade input function to input the filter processing result or Subtraction result again.

When the filter processing result is input, this cascade input function rounds the 32-bit accumulation circuit output data to 16-bit data, and then executes filter processing again.

When the Subtraction result is input, this cascade input function rounds the 32-bit subtraction circuit output data to 16-bit data, and then executes filter processing again.

In integer mode, when the accumulation circuit output data or subtraction circuit output data is a positive value and the 16 high-order bits contain a valid value, the output data is rounded to a 16-bit positive maximum value (7FFF_H). When the accumulation circuit output data or subtraction circuit output data is a negative value and the 16 high-order bits contain a valid value, the output data is rounded to a 16-bit negative minimum value (8000_H). If rounding of both positive and negative values occurs, the cascade rounding error flag is set to 1.

In fixed-point mode, the accumulation circuit output data or subtraction circuit output data is added to 0000 8000_H and rounding is executed. As a result of this addition, if an overflow occurs in 32-bit positive value, the added data is rounded to a positive maximum value (7FFF_H) and the cascade rounding error flag is set.

The CAEN bits in the DFEjCTLACHn or DFEjSUBCTLCHn control registers can enable or disable the cascade input function and enable or disable the output data register mask function.

Table 37.83 CAEN Flag Function

CAEN Bits	Description
00 _B	Cascade input is disabled. Calculation results are written to the output data register: Value after a reset.
01 _B	Setting prohibited
10 _B	Cascade input is enabled. Calculation results are written to the output data register.
11 _B	Cascade input is enabled. Calculation results are not written to the output data register.

(1) Cascade Input of Filter Processing Results

When a cascade input is enabled, the cascade input destination channel is specified by using the cascade tag value for (cascade tag hereafter). The cascade tag is set by the CATAG bits in DFEjCTLACHn. The DFE compares the cascade tag with the channel tag, and executes cascade input filter processing on the channel where the tags matched. The following figures show operations when the cascade input is enabled.

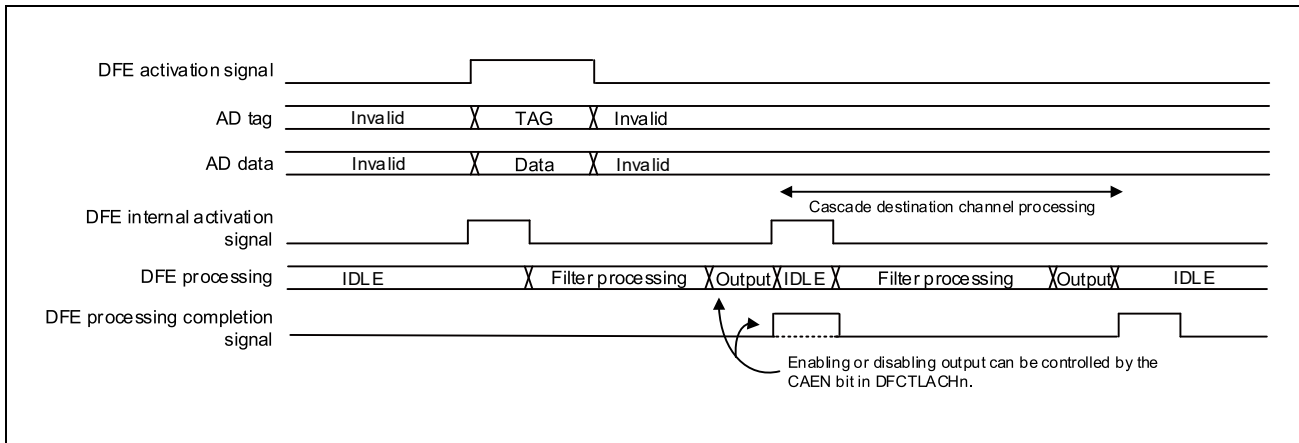


Figure 37.35 Cascade Input Processing (Normal Processing)

When a cascade input is enabled and decimation is specified, a decimation result becomes the cascade input data. When the decimation count is set to 4, for example, the data after filter processing of the fourth AD input becomes the cascade input data.

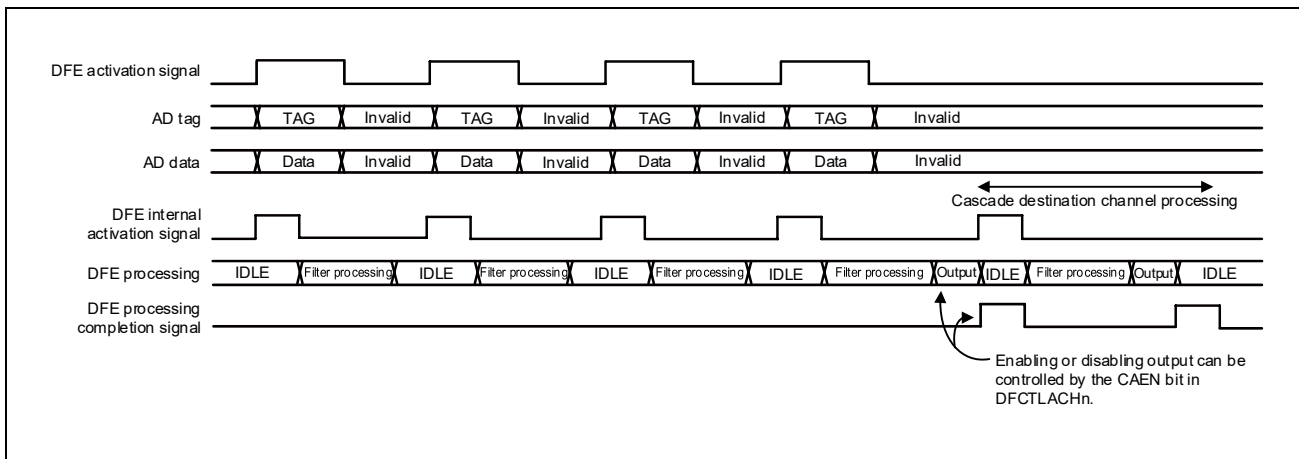


Figure 37.36 Cascade Input Processing (Decimation: 4 Times)

(2) Cascade Input of Subtraction Results

When the cascade input is enabled, a cascade tag is used for the subtraction channel to specify a cascade input destination channel. The cascade tag for subtraction channels is specified with the DFEjSUBCTLCHn.CATAG. The DFE, comparing the cascade tag for the subtraction channel with the channel tag, performs filtering on the cascade input on a matching channel. The text below describes the operation that occurs when the cascade input is enabled.

The cascade input on the results of the subtraction operation is used for each filtering process on the subtrahend channel and the minuend channel, regardless of the specified number of accumulation/decimation operations.

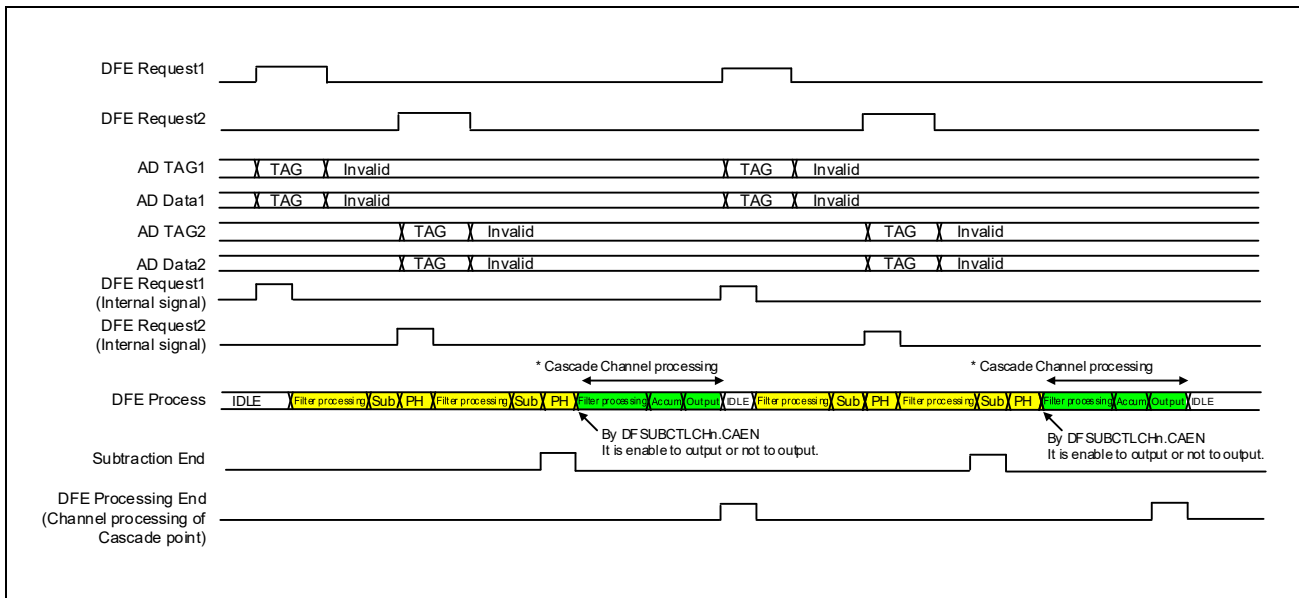


Figure 37.37 Cascading Input Processing on a Subtraction Channel

37.5.8.3 Software Input

The DFE has a function to input DFE processing target data through the peripheral bus by using the CPU or DMA. (This function is referred to as software input function hereafter.) DFE processing can be started by writing filter processing target data and a tag value to the software input data register (DI) from the peripheral bus.

37.5.8.4 Dispatch

A channel to be processed is selected by an activation signals of an AD input (Delta-Sigma AD , SAR-AD and Cyclic-AD), a cascade input, or a software input and the channel tag, and then the data to be processed is stored in the input data register of each channel. This processing is called dispatch.

After the data to be processed is stored in the input data register, the input data register is enabled until the accumulation circuit processing is completed. If new data is stored in the same channel while the input data register is enabled, the data is overwritten and causes an input data overwrite error (DIOW).

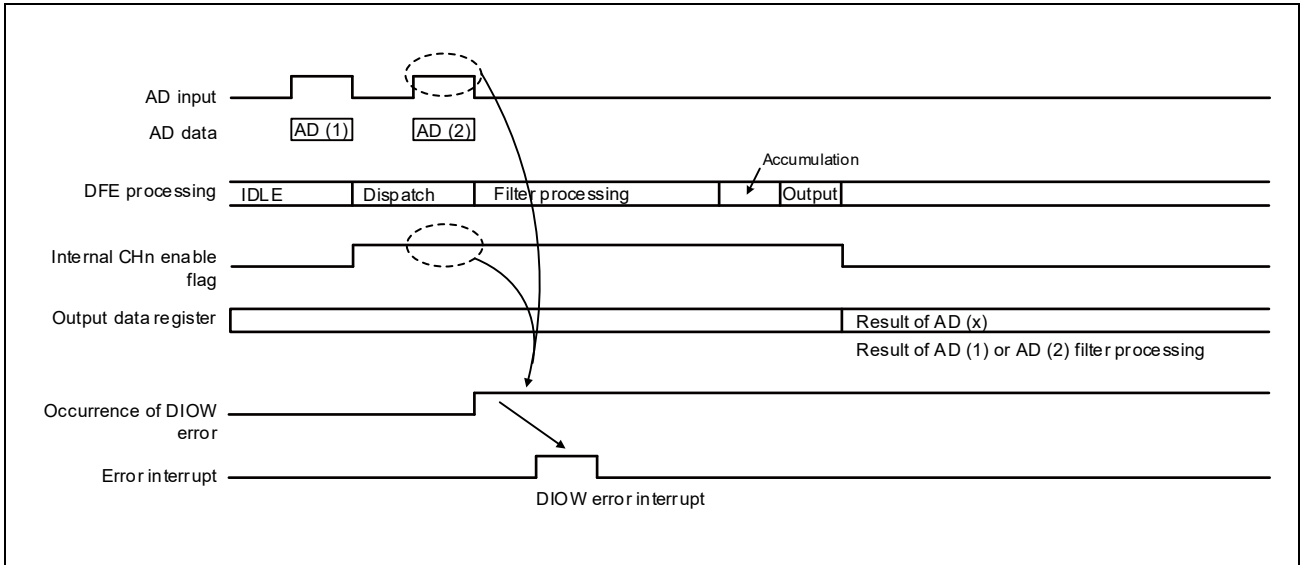


Figure 37.38 Input Data Overwrite (DIOW) Error

When the input data register of each channel is enabled, filter processing target data is arbitrated. The highest priority is given to the processing for channel 0, the second highest priority is given to the processing for channel 1, and the lowest priority is given to the processing for channel 11 in the DFE0, for channel 3 in the DFE1. The input data register value of arbitrated channels is input to the filtering circuit where FIR/IIR processing is applied to the input data register value according to the control register value.

If the same data is set for multiple channels, processing is also executed for all of the specified channels in the channel priority order. If the trigger flag function is enabled, it is not applied until the next input data processing.

For example, if the same input data is set for multiple channels, accumulation processing is set, and the same accumulation/decimation disable trigger is set for all the channels, accumulation is not disabled until the next input data processing if there are any channels for which accumulation processing has not been performed when the disable trigger is input.

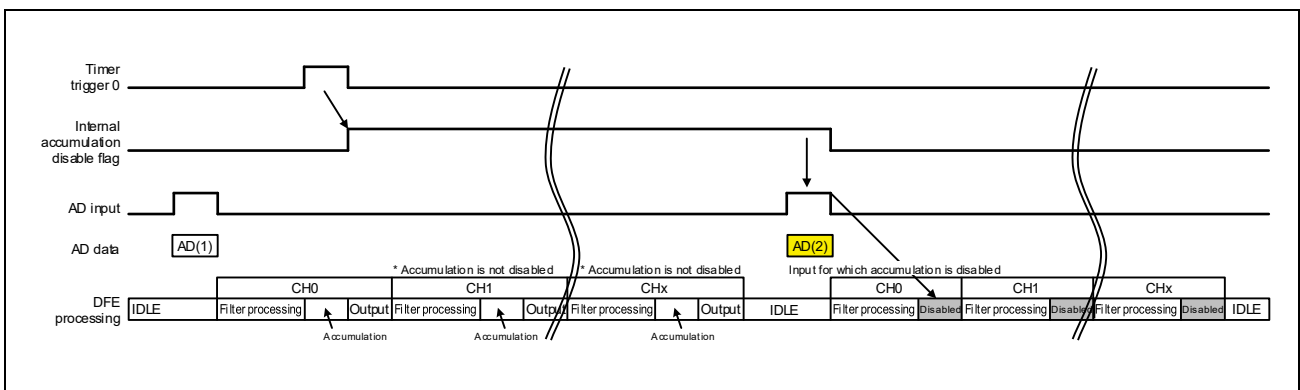


Figure 37.39 Processing when the Same Input Data is Set for Multiple Channels

37.5.9 Filtering Circuit

The filter circuit accepts requests for a single channel selected by the dispatch circuit from among channels 0 to 11 in the DFE0, channel 0 to 3 in the DFE1. When the FIR 64TAP is used on an even-numbered channel, although requests from the channel are accepted, no requests from the next odd-numbered channel are accepted. Once a request is accepted, the selected type of filtering (FIR or IIR) starts. Another request cannot be accepted during execution of filter processing. Upon completion of processing in the accumulation circuit, the DFE accepts a new request. Coefficients required for filtering are stored in the coefficient memory and input data to be used for FIR processing and intermediate data to be used for IIR processing are stored in the data memory. These memory areas are allocated for each channel.

37.5.9.1 FIR (8, 16, 24, 32 Taps)

Figure 37.40 shows the FIR filter configuration.

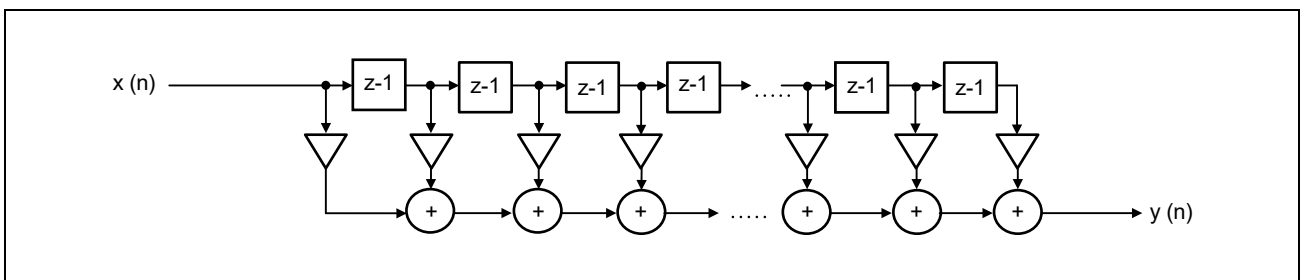


Figure 37.40 FIR Filter Configuration

Before starting FIR filter processing, set values in the coefficient memory and data memory buffer area.

For example, in the case of a 32-tap FIR filter, set value from $C[0]$ to $C[31]$ in coefficient memory and set 0000_{H} to circular buffer area (= ring buffer area) in data memory.

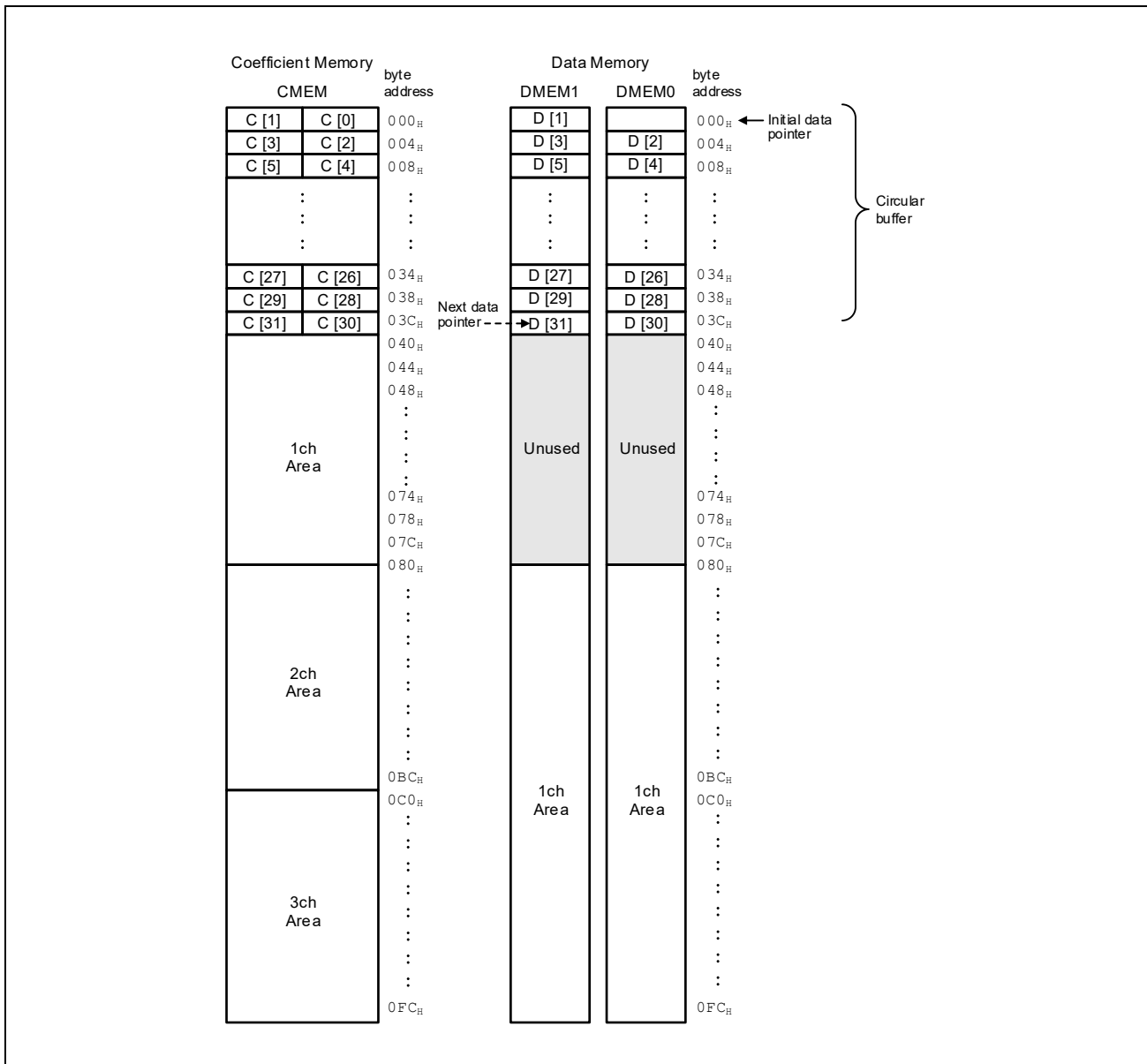


Figure 37.41 FIR Memory (FIR 32 TAP)

Figure 37.41 shows the access area of coefficient and data memory in the case of a 32-tap FIR.

When 8-tap/16-tap/24-tap FIR filters is selected, the access area of coefficient memory and data memory is same as 32-tap FIR.

When the filter processing initialization flag indicates “enabled”, DFE writes 0000_H in circular buffer area for the number of taps before filter processing starts. For example, when 32-tap FIR filter is selected, after DFE writes 0000_H in circular buffer area from D[0] to D[31], FIR processing is executed.

37.5.9.2 FIR (64 Taps)

When 64-tap FIR filter is selected, set values in the coefficient memory, and data memory buffer area as in 8-tap/16-tap/24-tap/32-tap FIR filter, before filter processing is executed.

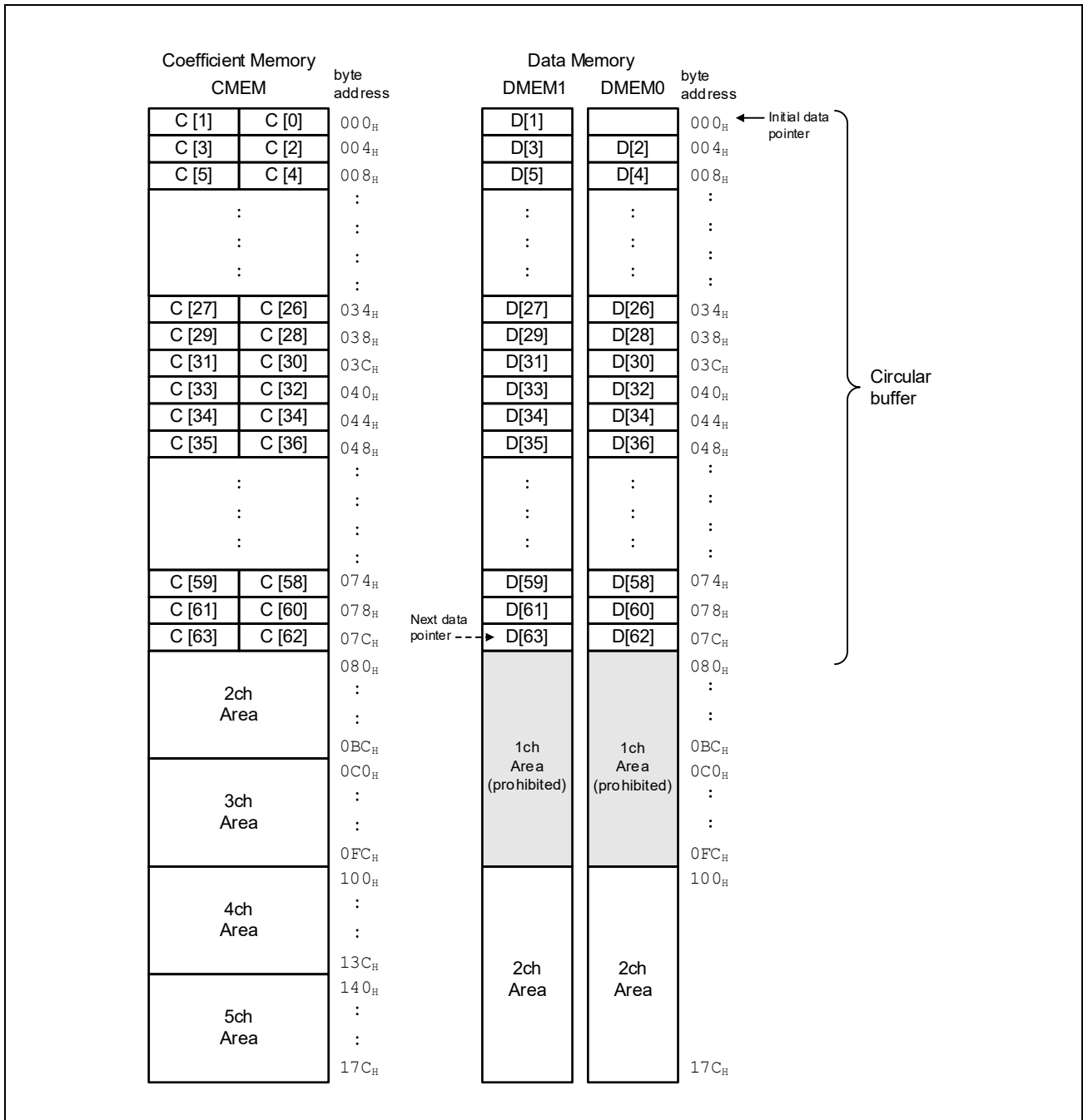


Figure 37.42 FIR Memory (when a 64-Tap FIR Filter is selected)

Figure 37.42 shows the access area in coefficient memory and data memory in the case of a 64-tap FIR. 64-tap FIR can be selected in only even number channels. When the filter processing initialization flag indicates “enabled”, DFE writes 0000_H in circular buffer area from D[0] to D[63] before filter processing is executed.

37.5.9.3 One-Stage IIR

Figure 37.43 shows the configuration of a one-stage IIR filter.

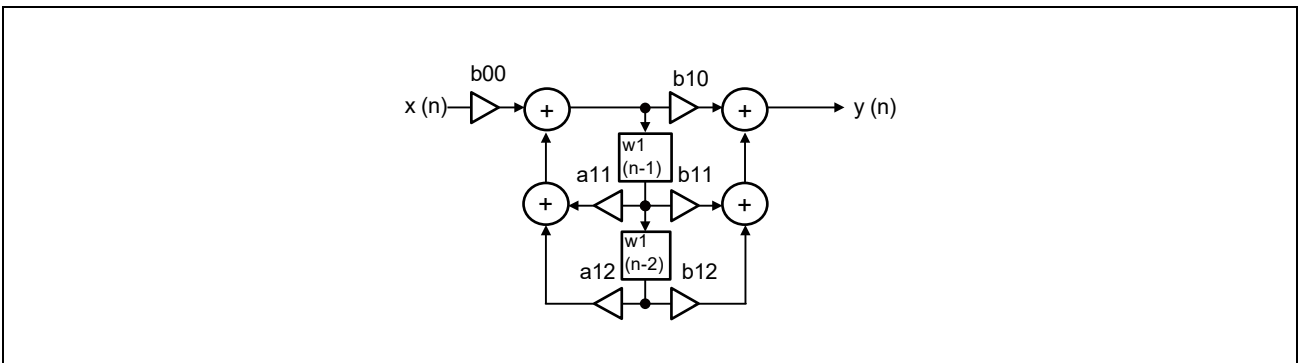


Figure 37.43 Configuration of a One-Stage IIR Filter

Before starting IIR filter processing, set values in the coefficient memory and data memory buffer area. Set the intermediate value in circular buffer area of the data memory. And set the initial value in initial value buffer area of the data memory.

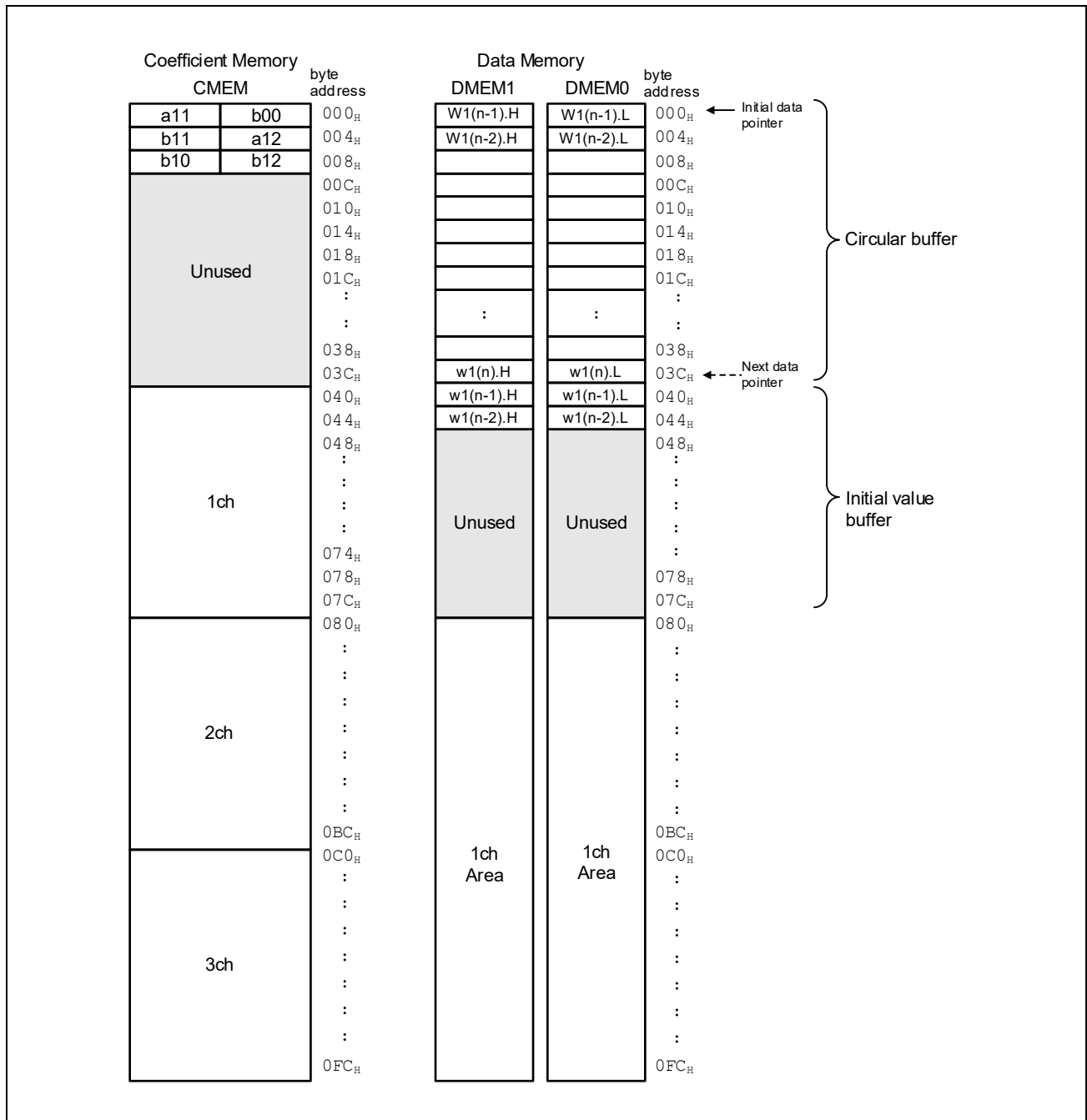


Figure 37.44 Memory Usage for a One-Stage IIR Filter

Figure 37.44 shows the access area in coefficient memory and data memory in the case of a one-stage IIR.

When the filter processing initialization flag indicates “enabled”, DFE initializes data pointer of the circular buffer area to “0”. And DFE writes the data of initial value buffer to the target circular buffer before filter processing is executed.

37.5.9.4 Two-Stage IIR

Figure 37.45 shows the configuration of a two-stage IIR filter.

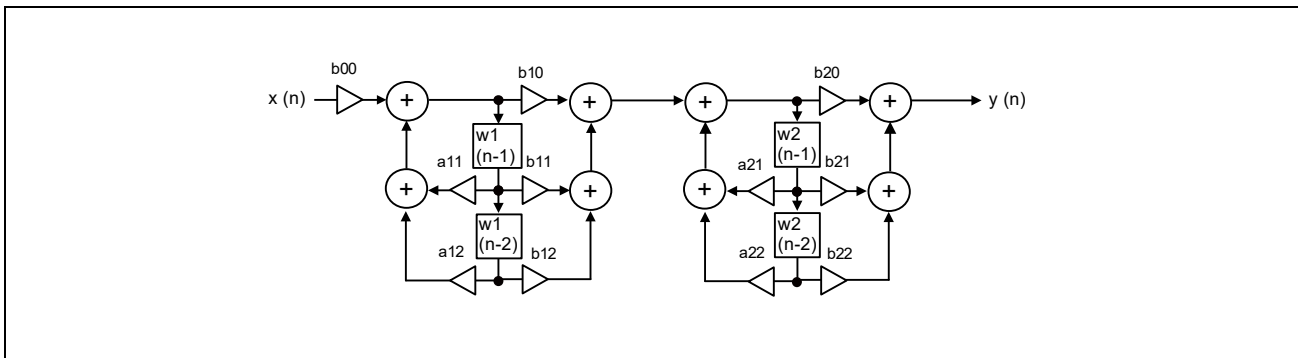


Figure 37.45 Configuration of a Two-Stage IIR Filter

The figure below shows an IIR filter configuration of a second-order 2-stage biquad IIR with a gain adjustment function.

The second-order 2-stage biquad IIR with a gain adjustment function can be used when $DFEjCTLACHn.CMD = 1101_B$. The value of GAIN2 can be set in the IIR filter gain setting register (DFEjGAINCHn).

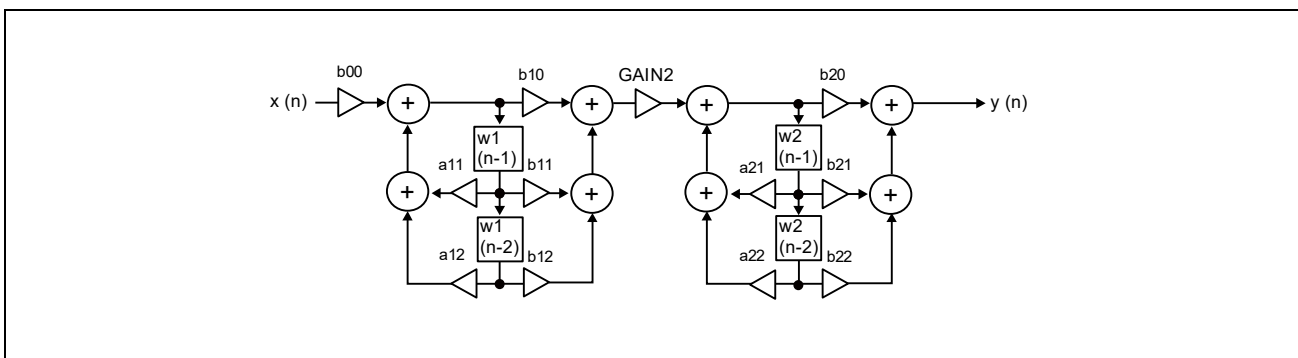


Figure 37.46 Configuration of a Two-Stage IIR Filter with Gain Adjustment Function

Before starting IIR filter processing, set values in IIR Filter Gain Setting Register (DFEjGAINCHn), the coefficient memory and data memory buffer area. Set the intermediate value in circular buffer area of the data memory. And set the initial value in initial value buffer area of the data memory.

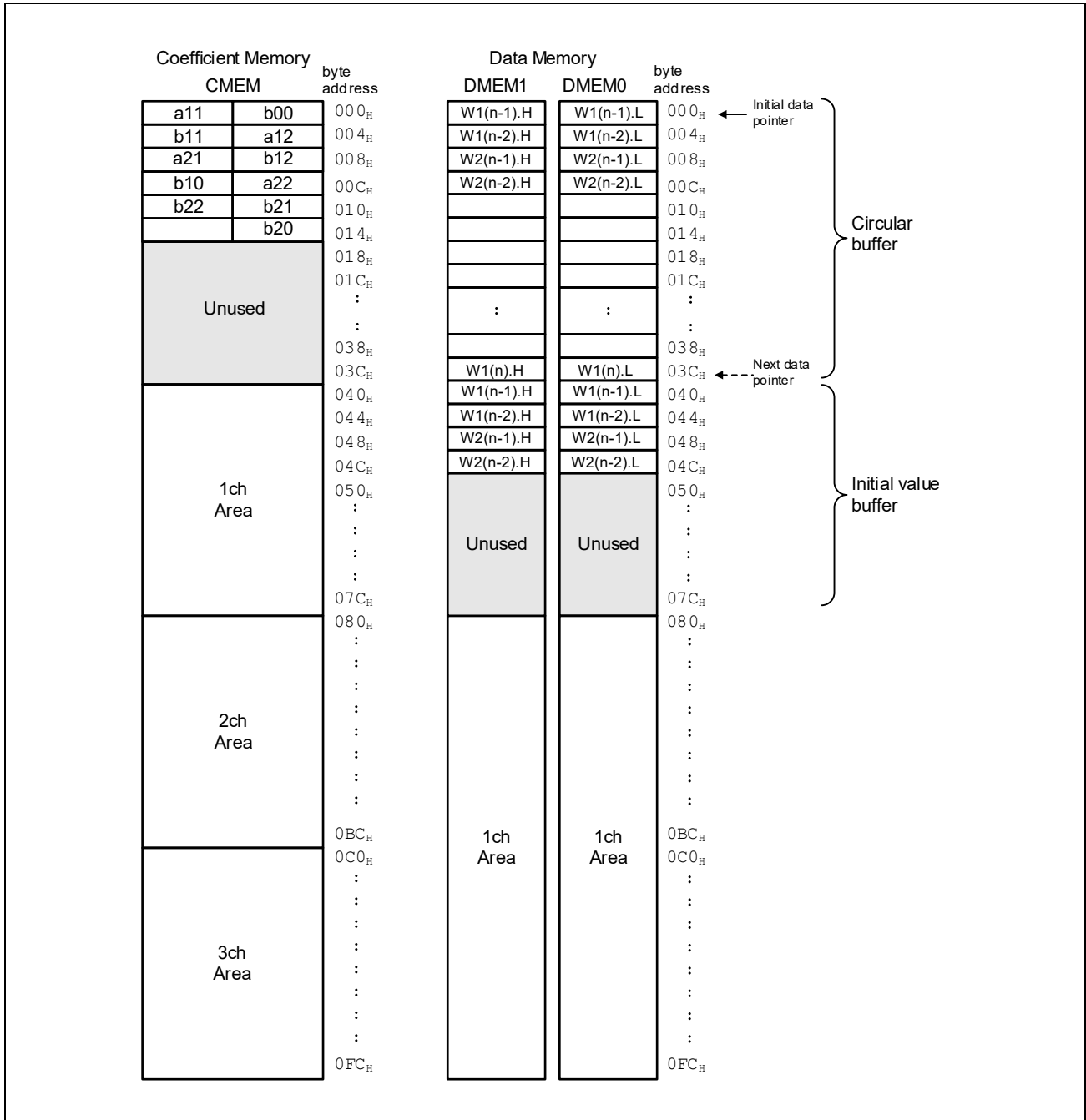


Figure 37.47 Memory Usage for a Two-Stage IIR Filter

Figure 37.47 shows the access area in the coefficient memory and data memory in the case of a two-stage IIR.

When the filter processing initialization flag indicates “enabled”, DFE initializes data pointer of the circular buffer area to “0”. And DFE writes data of initial value buffer to the target circular buffer before filter processing is executed.

37.5.9.5 Three-Stage IIR

Figure 37.48 shows the configuration of a three-stage IIR filter.

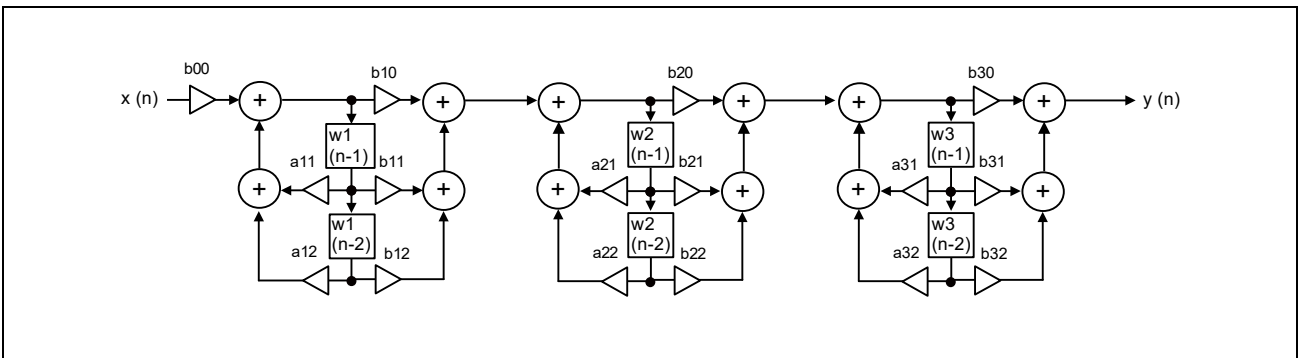


Figure 37.48 Configuration of a Three-Stage IIR Filter

The figure below shows an IIR filter configuration of a second-order 3-stage biquad IIR with a gain adjustment function. The second-order 3-stage biquad IIR with a gain adjustment function can be used when $DFEjCTLACHn.CMD = 1110_B$. The values of GAIN2 and GAIN3 can be set in the IIR filter gain setting register (DFEjGAINCHn).

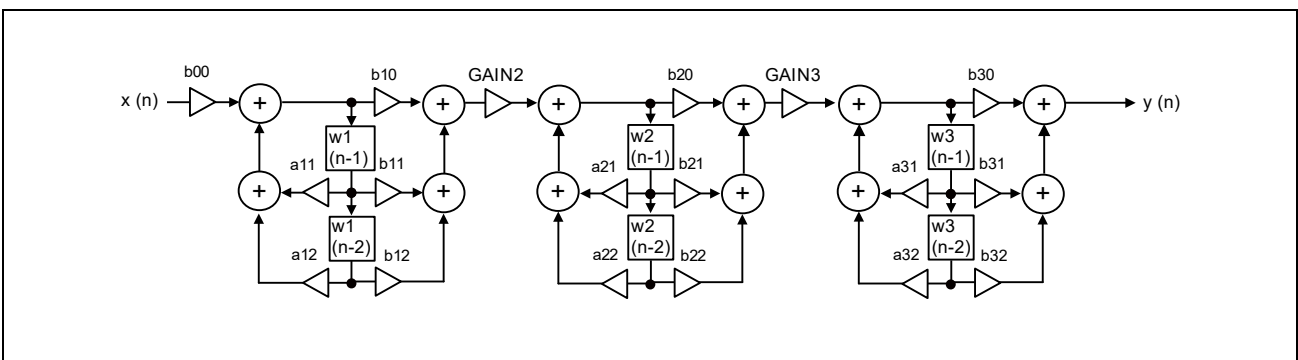


Figure 37.49 Configuration of a Three-Stage IIR with Gain Adjustment Function

Before starting IIR filter processing, set values in IIR Filter Gain Setting Register (DFEjGAINCHn), the coefficient memory and data memory buffer area. Set the intermediate value in circular buffer area of the data memory. And set the initial value in initial value buffer area of the data memory.

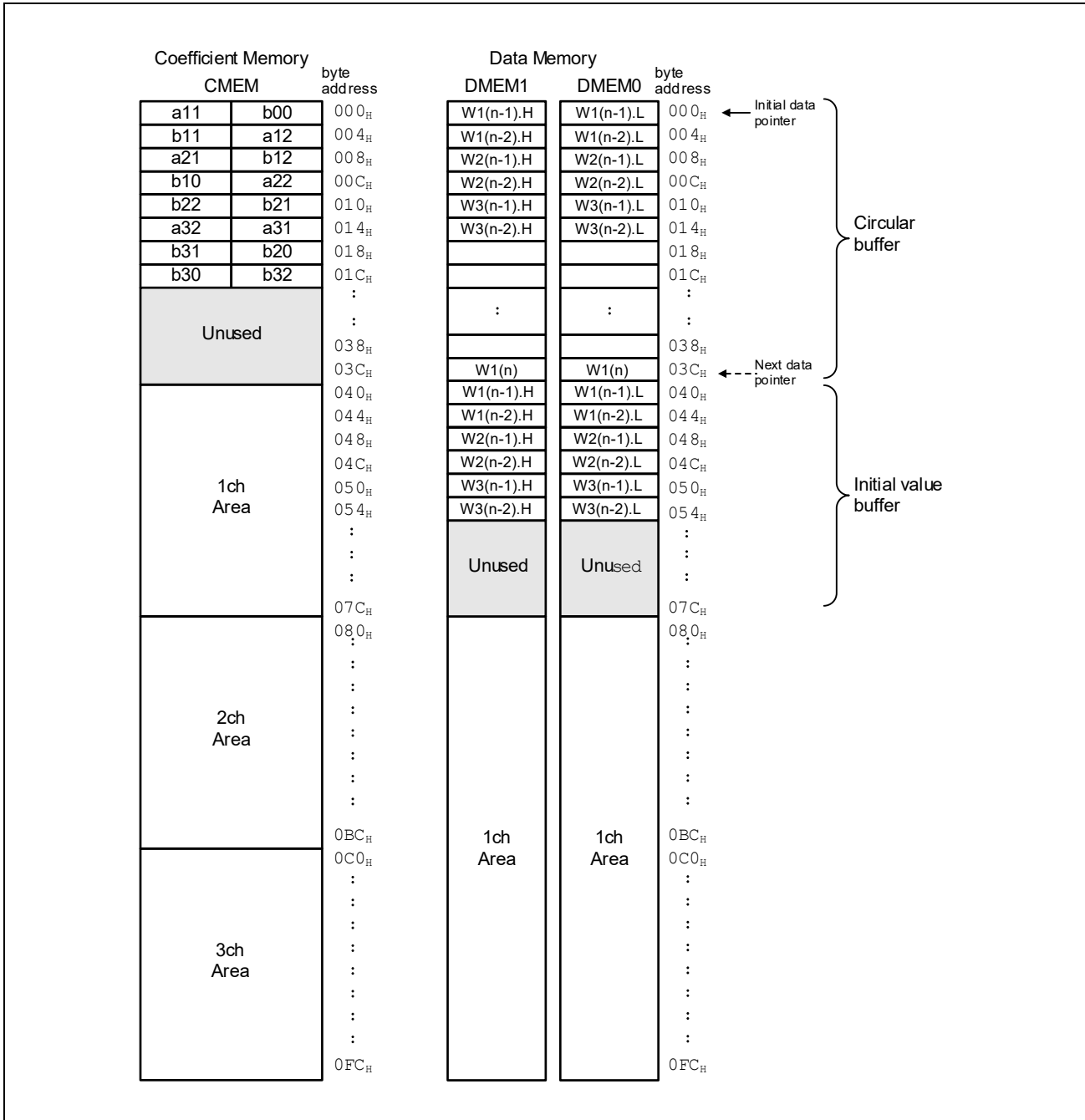


Figure 37.50 Memory Usage for a Three-Stage IIR Filter

Figure 37.50 shows the access area in the coefficient memory and data memory in the case of a three-stage IIR. When the filter processing initialization flag indicates “enabled”, DFE initializes data pointer of the circular buffer area to “0”. And DFE writes data of initial value buffer to the target circular buffer before filter processing is executed.

37.5.9.6 Errors in the Filtering Circuit

If multiplication of $8000_{\text{H}} \times 8000_{\text{H}}$ is performed by the multiplier, a multiplication error occurs and the multiplication error flag is set to 1.

To prevent an overflow in the multiplication result processing, 3 input adder (54bit) is composed to multiplication data area (48bit) and guard bit area (6bit). To round 54bit of multiplication result to 32bit, DFE accumulates $00\ 0000\ 0000\ 4000_{\text{H}}$ (when IIR 2.14 is selected, $00\ 0000\ 0000\ 2000_{\text{H}}$) to multiplication result (54bit). When guard bits contain a valid value after the rounding operation, the error flag is set to 1. This error is called guard bit error.

When the rounded multiplication result is larger than the 32-bit positive maximum value ($7\text{FFF}\ \text{FFFF}_{\text{H}}$), the result is rounded to $7\text{FFF}\ \text{FFFF}_{\text{H}}$ (saturation processing). When the rounded multiplication result is smaller than the 32-bit negative minimum value ($8000\ 0000_{\text{H}}$), the result is rounded to $8000\ 0000_{\text{H}}$ (saturation processing).

37.5.10 Accumulation Circuit

The accumulation circuit has a function to execute absolute value conversion, accumulation processing, or decimation processing for the filter processing result data that is output from the filtering circuit. The accumulation circuit then outputs the processing result data to the output circuit and/or the peak-hold circuit. When the cascade function is used, the accumulation circuit sends the accumulation circuit output data to the dispatch circuit.

37.5.10.1 Absolute Value Calculation

The filter processing result data that is output from the filtering circuit is converted to an absolute value. When the ABS bit in DFEjCTLBCHn is 1, absolute value conversion is executed. When the ABS bit is 0, absolute value conversion is not executed. If the filter processing result data that is output from the filtering circuit is $8000\ 0000_{\text{H}}$, an absolute value error is generated. The absolute value conversion result when this occurs is $7\text{FFF}\ \text{FFFF}_{\text{H}}$.

Absolute value conversion can be performed even if the PRCSA bit is set to 00_{B} in DFEjCTLBCHn . Accumulation/decimation processing is performed on the absolute value of conversion result.

37.5.10.2 Summary of Accumulation Processing

In accumulation processing, the filter processing result data that is input to the accumulation circuit is arithmetically shifted to the right (9 bits), and then this filter processing result data is added to the filter processing result data that has been input for each channel.

The data format after the accumulation processing is (10.22). Accumulation processing can be executed up to 511 times. After processing for the specified accumulation count has been completed, accumulation processing is executed again.

When the PRCSA bits in DFEjCTLBCHn are 01_{B} , accumulation processing is executed. Specify the accumulation processing count with the SELA bits in DFEjCTLBCHn by selecting one of the four accumulation/decimation count setting registers (ACA to ACD).

37.5.10.3 Operation of Accumulation Processing

The operation of accumulation processing is explained using an example when $ACA = 000_H$, $ACB = 001_H$, $ACC = 002_H$, and $DFEjCTLBCHn.PRCSA = 01_B$.

When the SELA bits in $DFEjCTLBCHn$ are 00_B , ACA is selected and accumulation processing is executed with an accumulation count value of 0.

The 9-bit arithmetical right-shift result data (DADD (1)) from the filter processing result data (D (1)) is written to the output data register and an output data interrupt request is issued.

In accumulation processing with an accumulation count value of 0, one-time filter processing result is right shifted 9-bits arithmetically without addition processing and an interrupt request is issued. This processing is repeated continuously.

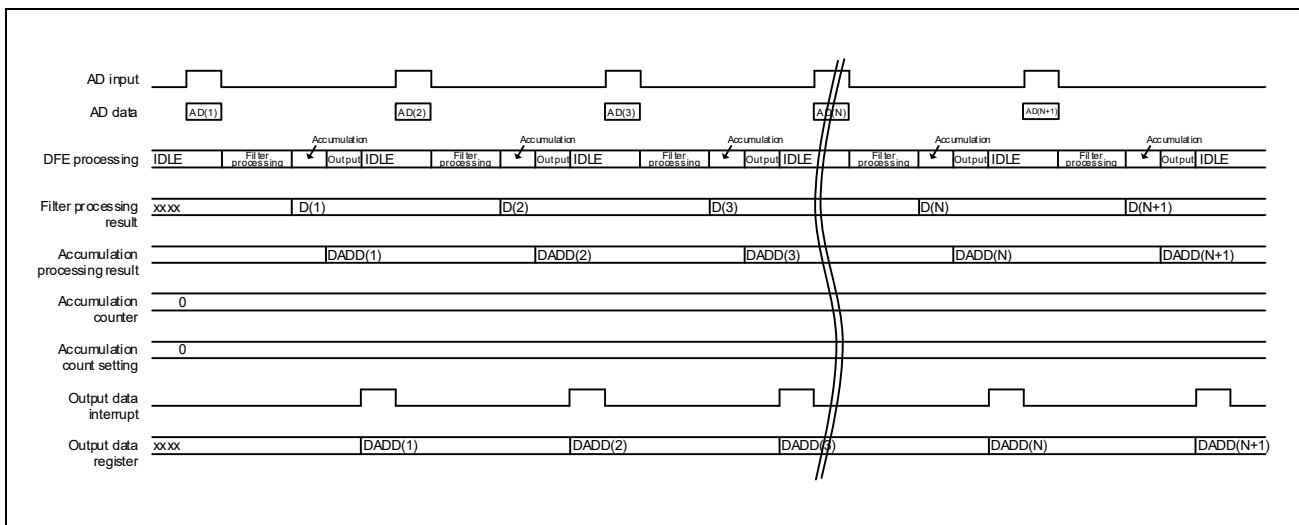


Figure 37.51 Accumulation Processing (Count Value = 0)

When the SELA bits in $DFEjCTLBCHn$ are 01_B , the ACB register is selected and accumulation processing is executed with a set accumulation count value of 1.

When the filter processing result data (D (1)) is input to the accumulation circuit, the 9-bit arithmetical right-shift result data (DADD (1)) from the filter processing result data (D (1)) is written to the accumulator. After that, when the filter processing result data (D (2)) is input to the accumulation circuit, the filter processing result data (D (2)) is right shifted (9 bits) arithmetically (DADD (2)) and this data is added to the accumulator value (DADD (1)). The results of the addition (DADD (1) + DADD (2)) are written to the output data register. After this writing has been completed, an output data interrupt request is issued.

When the accumulation count is set to 1, a 9-bit arithmetical right shift is made for two filter processing results and these results are added, and then an interrupt request is issued once. This processing is repeated continuously.

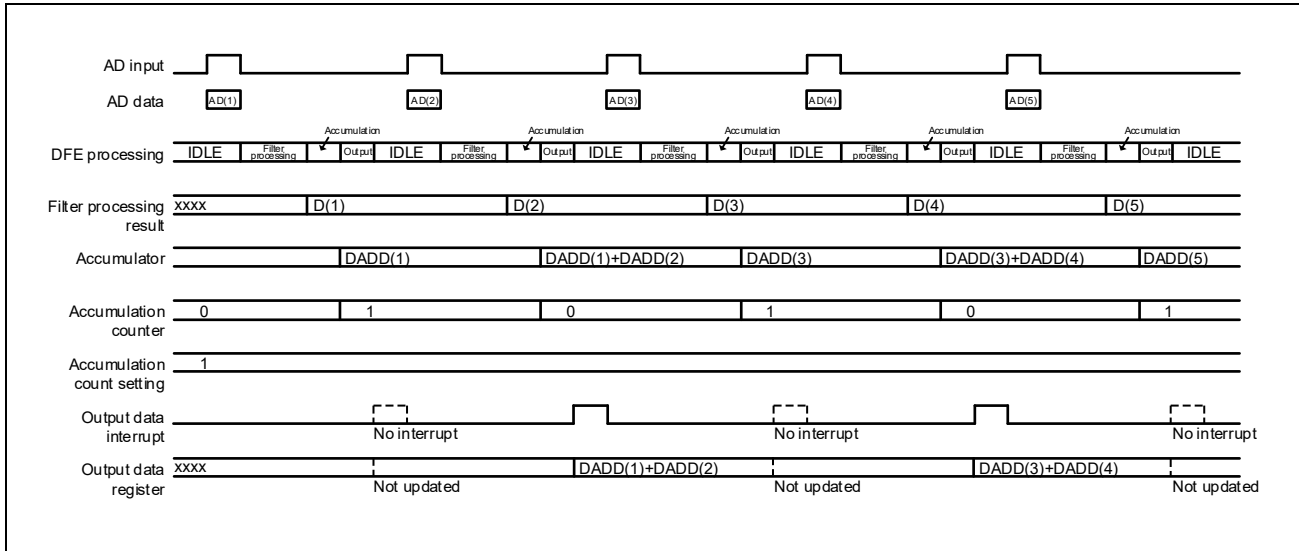


Figure 37.52 Accumulation Processing (Count Value = 1)

When the SELA bits in DFEjCTLBCHn are 10B, the ACC register is selected and accumulation processing is executed with a set accumulation count value of 2.

When the filter processing result data (D (1)) is input to the accumulation circuit, the 9-bit arithmetical right-shift result data (DADD (1)) from the filter processing result data (D (1)) is written to the accumulator. After that, when the filter processing result data (D (2)) is input to the accumulation circuit, the filter processing result data (D (2)) is right shifted (9 bits) arithmetically (DADD (2)) and this data is added to the accumulator value (DADD (1)). The results of the addition (DADD (1) + DADD (2)) are written to the accumulator again. Next, when the filter processing result data (D (3)) is input to the accumulation circuit, the filter processing result data (D (3)) is right shifted (9 bits) arithmetically (DADD (3)) and this data is added to the accumulator value (DADD (1) + DADD (2)). The results of the addition (DADD (1) + DADD (2) + DADD (3)) are written to the output data register. After this writing has been completed, an output data interrupt request is issued.

When the accumulation count is set to 2, a 9-bit arithmetical right shift is made for three filter process results and these results are added, and then an interrupt request is issued once. This processing is repeated continuously.

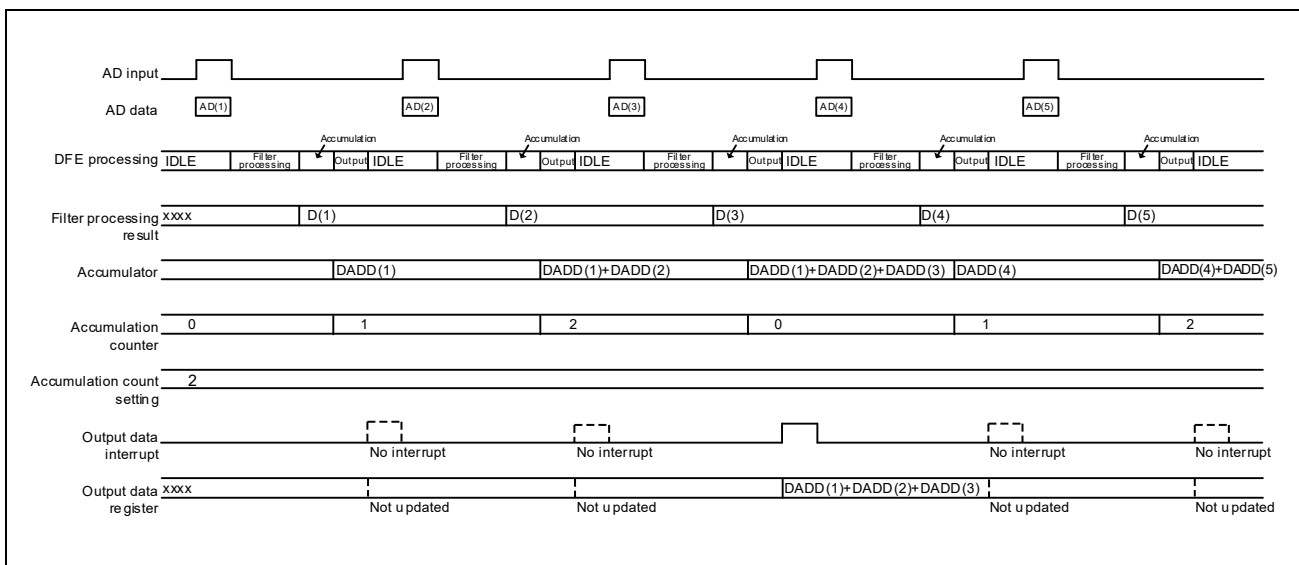


Figure 37.53 Accumulation Processing (Set Count Value = 2)

37.5.10.4 Operation of Accumulation Processing (with Trigger Flag)

The following describes operation of accumulation processing when the accumulation initialization flag and the accumulation disable flag are enabled. Alternate between input to the accumulation initialization and accumulation disable flags. Operations for accumulation processing are not guaranteed in case of consecutive input to either flag.

When the accumulation initialization flag is enabled by a timer trigger or a software trigger while accumulation is specified (PRCSA = 01_B in DFEjCTLBCHn), the accumulation counter is initialized and accumulation processing is executed on the first data from the target channel that immediately follows.

When the accumulation initialization flag is enabled and the accumulation count is set to 2, for example, and data (D (3)) is input to the accumulation circuit for the third time, the 9-bit arithmetical right-shift (DADD (3)) from the third data (D (3)) is written to the accumulator. The accumulation result value (DADD (1) + DADD (2)) is overwritten.

The data (D (4)) that is input to the accumulation circuit is added to the data (D (5)) that is input to the accumulation circuit. Three accumulation results ((DADD (3) + DADD (4) + DADD (5)) are written to the output data register after the accumulation initialization flag is enabled, and an output data interrupt request is issued.

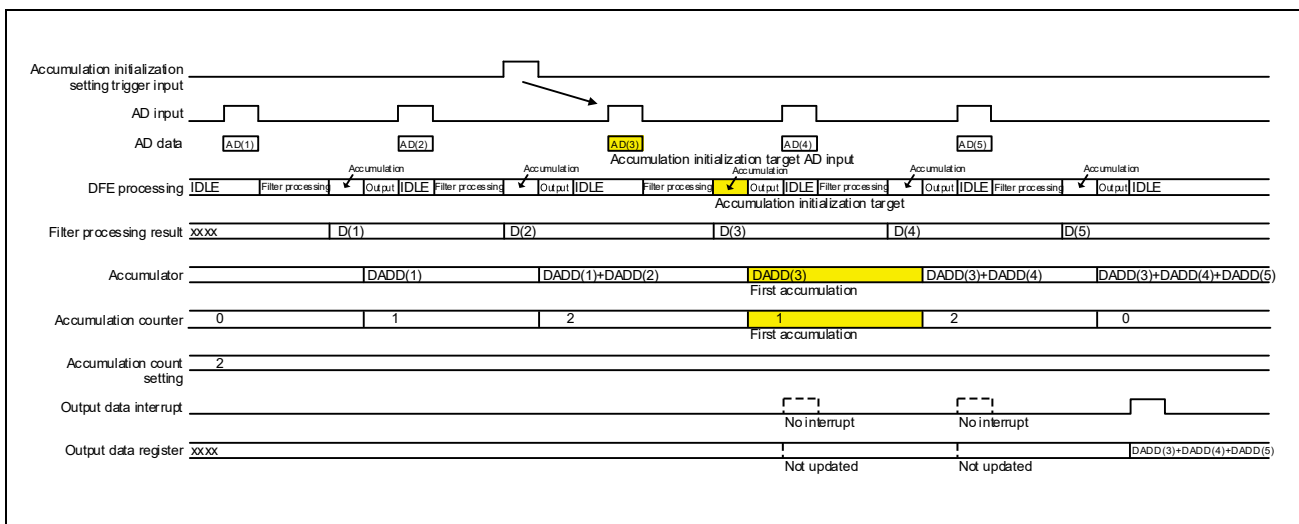


Figure 37.54 Accumulation Processing (with the Initialization Flag Enabled)

When the accumulation disable flag is enabled by a timer trigger or a software trigger while accumulation is specified (PRCSA = 01_B in DFEjCTLBCHn), the DFE's accumulation processing of the target channel that immediately follows is not executed and the internal accumulation disable flag is asserted. No accumulation is performed in the subsequent processing.

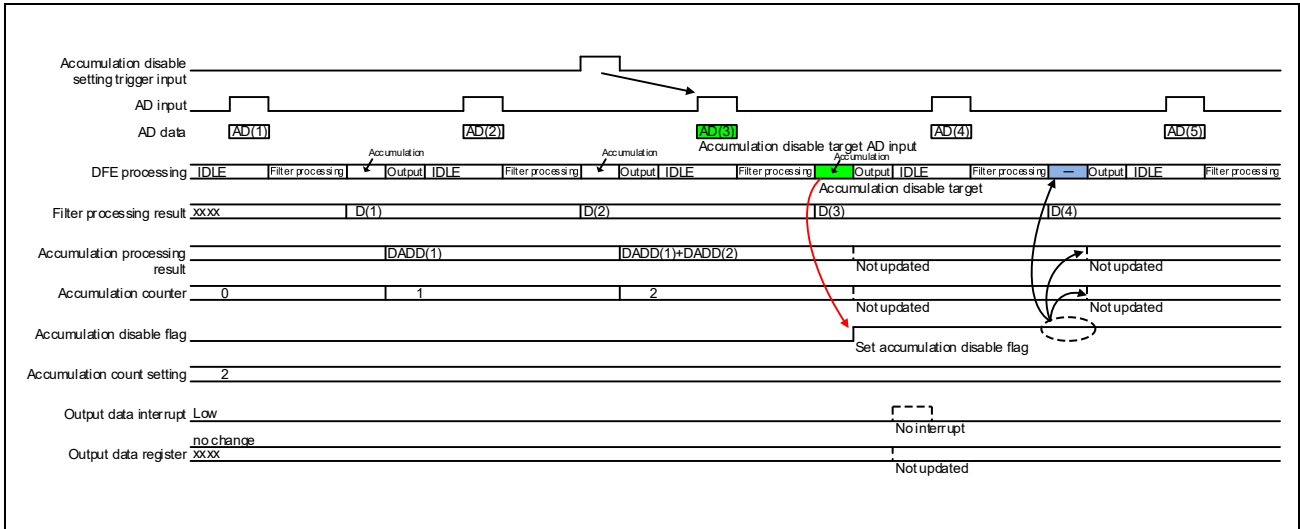


Figure 37.55 Accumulation Processing (with the Accumulation Disable Flag Enabled)

To execute accumulation processing again while the accumulation disable flag is asserted, enable the accumulation initialization flag.

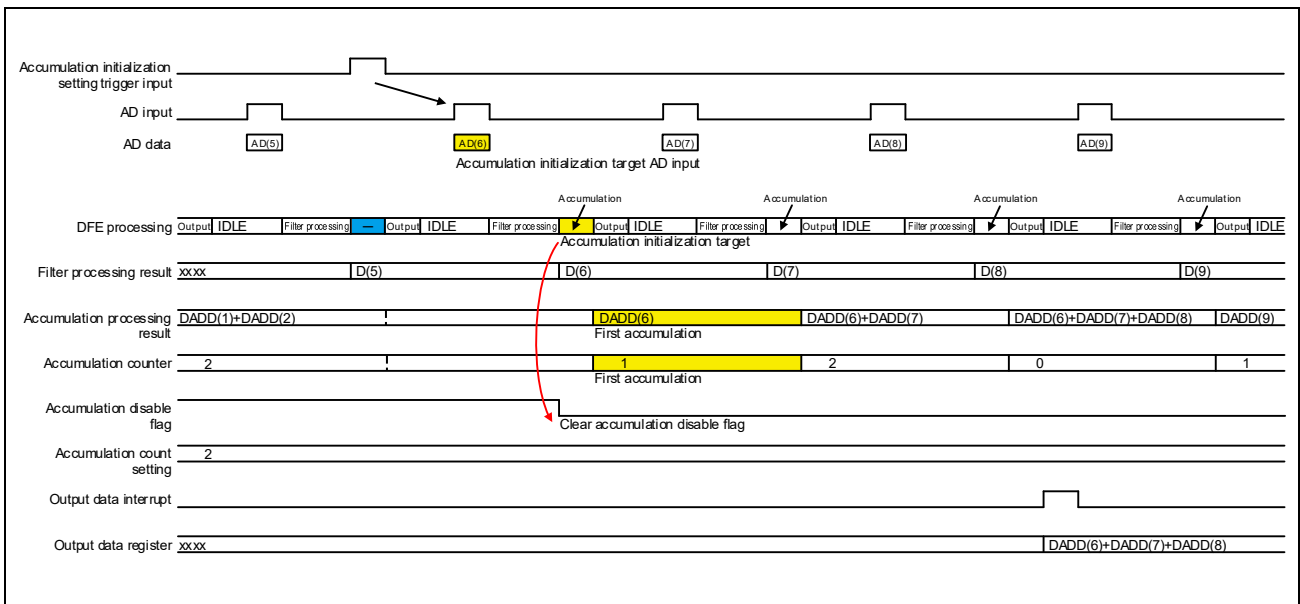


Figure 37.56 Accumulation Processing (Disable to Enable)

37.5.10.5 Summary of Decimation Processing

In the decimation processing, filter processing result data that is input to the accumulation circuit is decimated. The decimation processing count can be set to up to 511. (Data is output once every 512 times.) After processing for the specified decimation count has been completed, decimation processing is executed again.

When the PRCSA bits in DFEjCTLBCHn are 10_B, decimation processing is executed. Specify the decimation processing count by setting the SELA bits in DFEjCTLBCHn to one of the four accumulation/decimation count setting registers (ACA to ACD).

37.5.10.6 Operation for Decimation Processing

The operation for decimation processing is explained using an example when ACA = 0_H, ACB = 1_H, ACC = 2_H, and the PRCSA bits in DFEjCTLBCHn = 10_B.

When the SELA bits in DFEjCTLBCHn are set to 00_B, ACA is selected and decimation processing is executed with a decimation count value of 0.

In decimation processing with a decimation count value of 0, the filter processing result is output as it was input and an interrupt request is issued. This processing is repeated continuously.

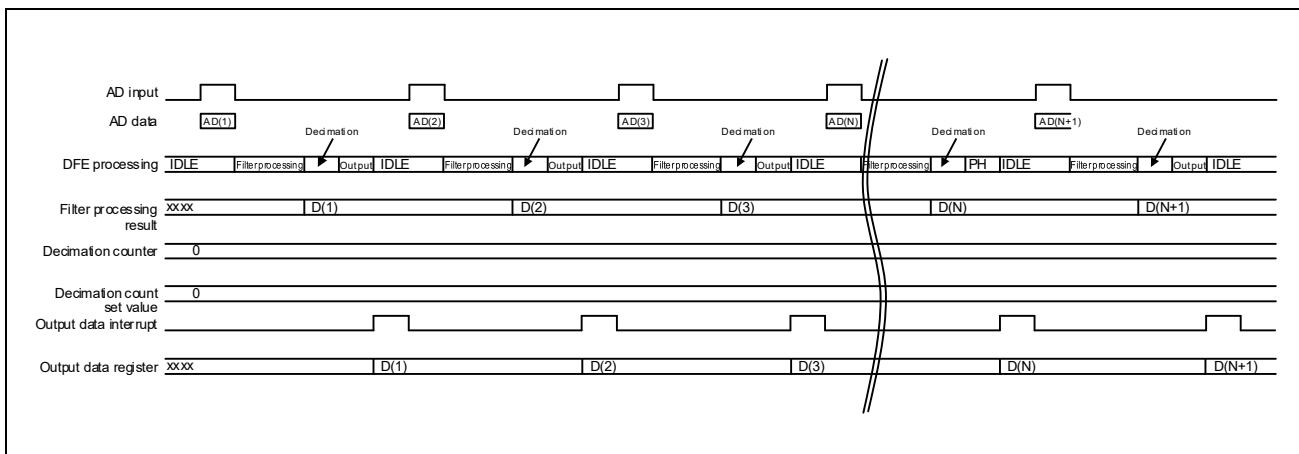


Figure 37.57 Decimation Processing (Count Value = 0)

When the SELA bits in DFEjCTLBCHn are set to 01_B, the ACB register is selected and decimation processing is executed with a decimation count value of 1.

When the filter processing result data (D (1)) is input to the accumulation circuit, the decimation counter is incremented. At this time, the accumulation circuit does not output the input filter processing result data (D (1)) to the output circuit because the decimation counter value differs from the specified decimation count value.

When the filter processing result data (D (2)) is input in to the accumulation circuit, the accumulation circuit outputs the input filter processing result data (D (2)) to the output circuit because the decimation counter value is equal to the specified decimation count value. The output circuit writes the data (D (2)) that is input from the accumulation circuit to the output data register and issues an output data interrupt request.

When the decimation count is set to 1, an interrupt request is issued once every two filter processing cycles. This processing is repeated continuously.

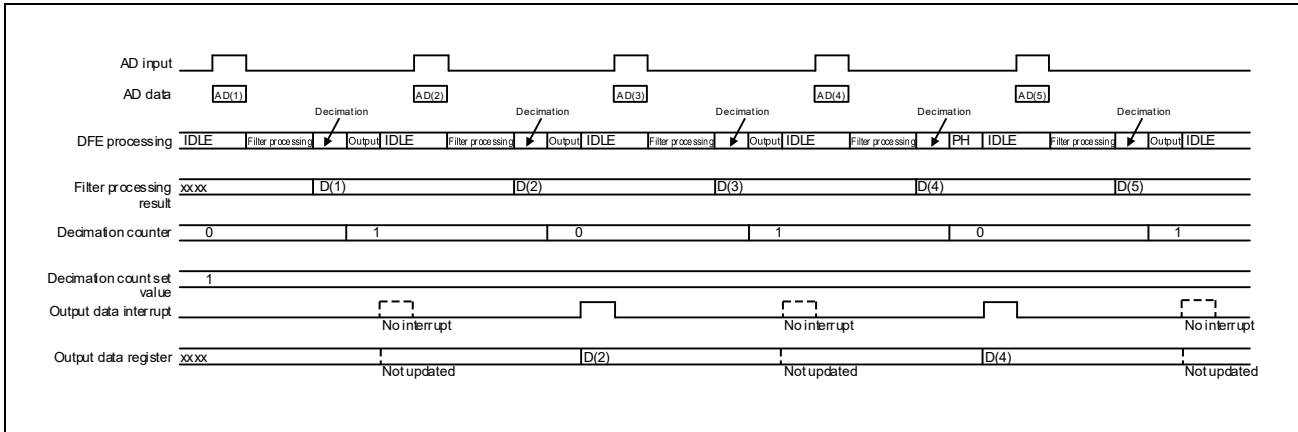


Figure 37.58 Decimation Processing (Count Value = 1)

When the SELA bits in DFEjCTLBCHn are set to 10_B, the ACC register is selected and decimation processing is executed with a set decimation count value of 2.

When the filter processing result data (D (1)) is input to the accumulation circuit, the decimation counter is incremented. At this time, the accumulation circuit does not output the input filter processing result data (D (1)) to the output circuit because the decimation counter value differs from the specified decimation count value.

Then, when the filter processing result data (D (2)) is input to the accumulation circuit, the decimation counter is incremented. At this time, the accumulation circuit does not output the input filter processing result data (D (2)) to the output circuit because the decimation counter value differs from the specified decimation count value.

When the filter processing result data (D (3)) is input to the accumulation circuit, the accumulation circuit outputs the input filter processing result data (D (3)) to the output circuit because the decimation counter value is equal to the set decimation count value. The output circuit writes the data (D (3)) that is input from the accumulation circuit to the output data register and issues an output data interrupt request.

When the decimation count is set to 2, an interrupt request is issued once for every three filter processing cycles. This processing is repeated continuously.

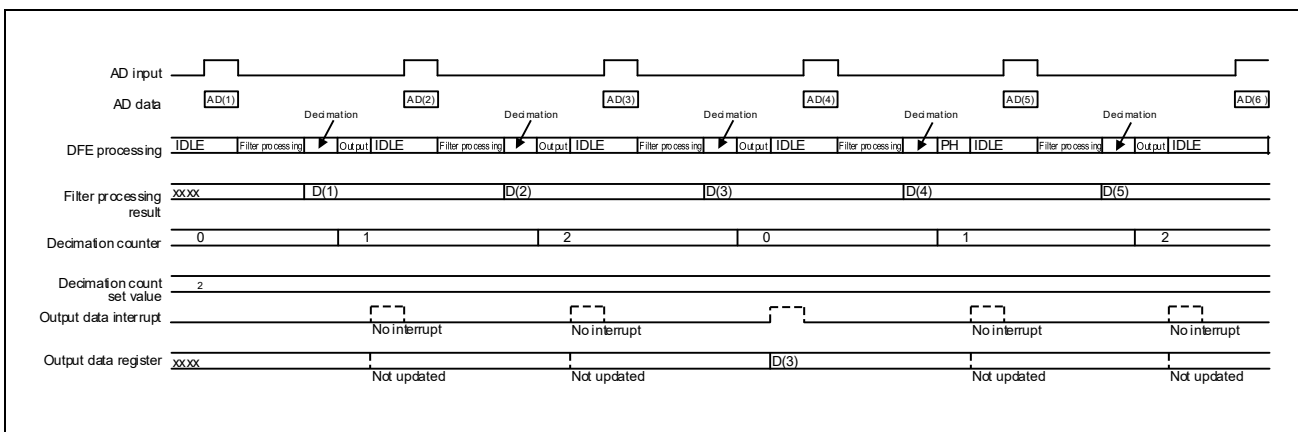


Figure 37.59 Decimation Processing (Count Value = 2)

37.5.10.7 Operation of Decimation Processing (with Trigger Flag)

The following describes the operation of decimation processing when the decimation initialization flag and the decimation disable flag are enabled. Alternate between input to the decimation initialization and decimation disable flags. Operations for decimation processing are not guaranteed in case of consecutive input to either flag. When the decimation initialization flag is enabled by a timer trigger or a software trigger while decimation is specified (PRCSA = 10_B in DFEjCTLBCHn), the decimation counter is initialized and decimation processing is executed on the first data of the target channel that immediately follows.

When the decimation initialization flag is enabled and the decimation count is set to 2, for example, when data (D (3)) is input to the accumulation circuit for the third time, the third data (D (3)) is not output from the accumulation circuit, but is processed as the first input data. Subsequently, the accumulation circuit does not output data (D (4)) that is input to the accumulation circuit, but does output data (D (5)) to the output circuit that is input to the accumulation circuit.

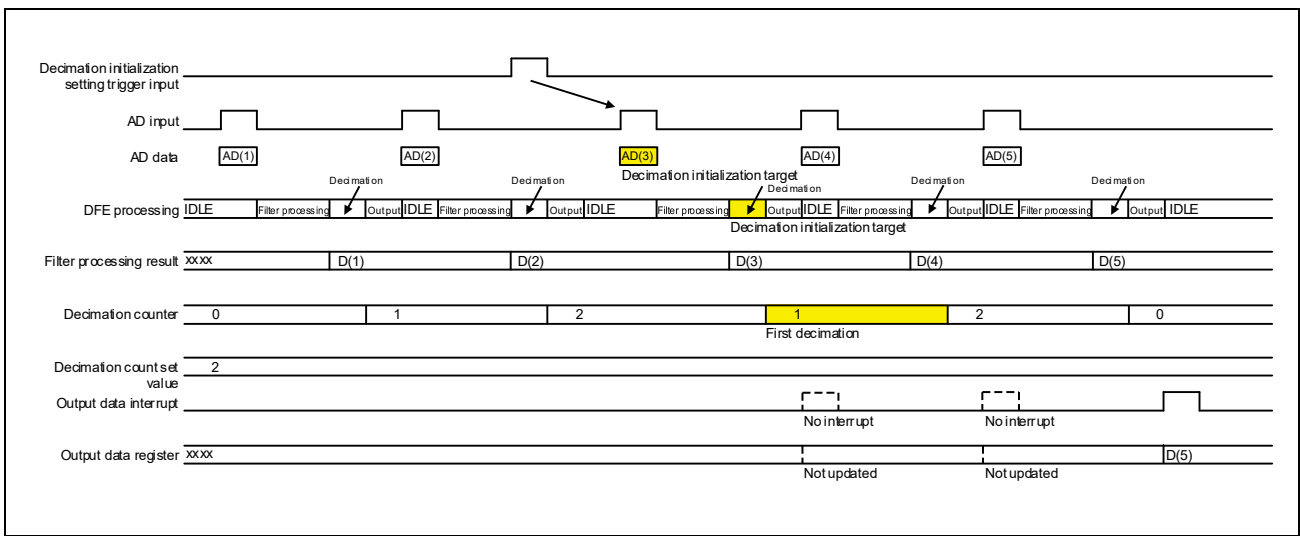


Figure 37.60 Decimation Processing (with the Initialization Flag Enabled)

When decimation is specified (PRCSA = 10_B in DFEjCTLBCHn) and the decimation disable flag is enabled, the internal decimation disable flag is asserted. In the subsequent processing, the decimation counter is disabled. While decimation is disabled, no output data interrupt request is issued.

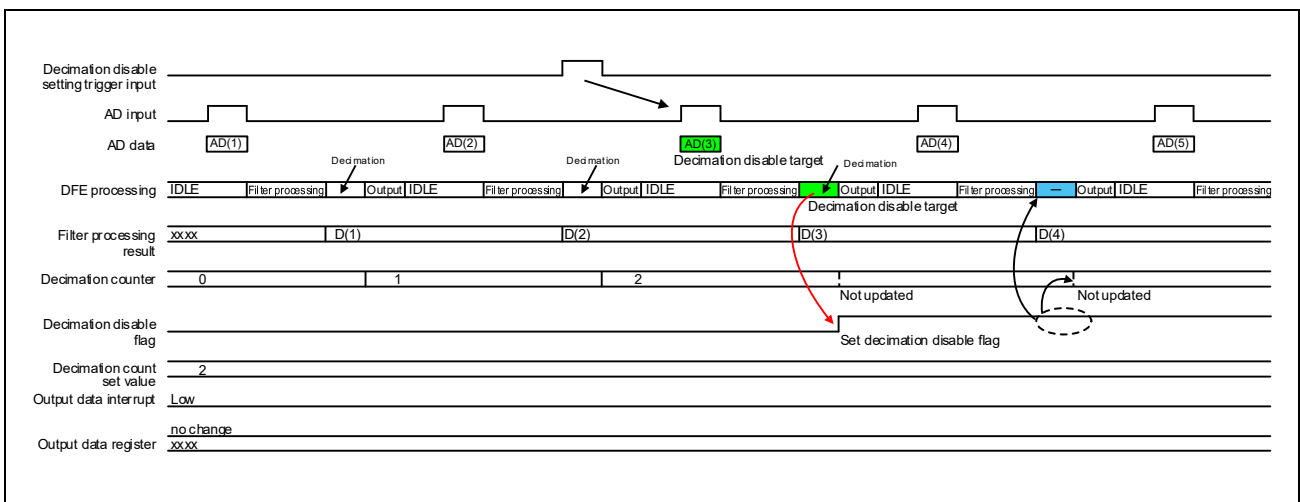


Figure 37.61 Decimation Processing (with the Decimation Disable Flag Enabled)

To execute the decimation processing again while the internal decimation disable flag is asserted, enable the decimation initialization flag.

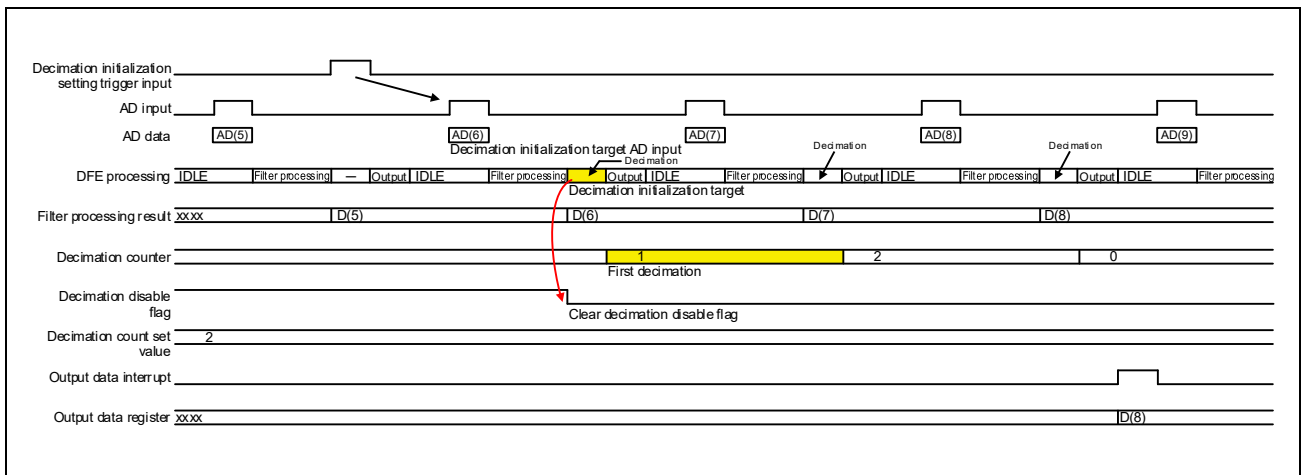


Figure 37.62 Decimation Processing (Disable to Enable)

37.5.11 Subtraction Circuit

After the end of processing by the filtering circuit, the subtraction circuit performs subtraction processing by using the filtering process result data for any two channels. After subtraction processing, the subtraction result data is stored in the subtraction result register. A subtraction data interrupt request is issued, when the subtraction result data has been stored. A subtraction error interrupt request is also issued, if an error has occurred in the subtraction circuit.

If the cascade function is specified to the subtraction channel, the subtraction circuit output data is input to the dispatch circuit.

The minuend channel and the subtrahend channel must not be set to the same channel.

Use the same format as the filtering process result for the minuend channel and the subtrahend channel. Subtraction processing of the filter processing result of different formats is prohibited. In this case, the data will not be guaranteed.

Subtraction is irrelevant to the specification of accumulation/decimation. Be aware that subtraction among accumulation results (S9.22) is not possible if the input data is in a fixed-point format.

The format of the input data uniquely determines the format of the subtraction result.

When the input data is in the fixed-point (S.15) format, the subtraction result is in the S.31 format.

When the input data is in the integer (S.15) format, the subtraction result is in the S31 format.

Filtering Process Result Format	Floating-point conversion	Execution (Result)	Cascade (Output)
S.31 (Fixed-Point)	Conversion specified	OK (floating-point)	OK (S.15)
	Conversion not specified	OK (S.31)	OK (S.15)
S31 (Integer)	Conversion specified	NO*1	OK (S15)
	Conversion not specified	OK (S31)	OK (S15)

Note 1. Setting is prohibited for NO combinations.

37.5.11.1 Summary of the Subtraction Circuit

If subtraction is specified (DFEjSUBCTLCHn.SEN = 1), after the end of processing by the filtering circuit, subtraction processing is executed on the basis of the filtering process results for the channels set with the minuend channel selection bit (DFEjSUBCTLCHn.MINCH, hereafter called MINCH) and the subtrahend channel selection bit (DFEjSUBCTLCHn.SUBCH, hereafter called SUBCH).

The subtraction result is stored in the subtraction result data register (DFEjSUBDOCHn.SDO, hereafter called SDO).

If 1 is set to the subtraction data interrupt request enabling bit (DFEjSUBCTLCHn.SIEO), the subtraction data interrupt request is issued when the data is stored in the subtraction result data register.

In subtraction processing, three error flags are used.

- (1) "Subtraction result register overwrite error (DFEjSUBSTCHn.SDOOW)" that detects overwrite of the subtraction result data.
- (2) "Subtraction circuit cascading rounding error (DFEjSUBSTCHn.SCER)" that detects an overflow that occurs when 32-bit data is rounded to 16-bit data, and when the subtraction result data is used for cascade processing.
- (3) "Subtraction guard error (DFEjSUBSTCHn.SGER)" that detects an overflow of the subtraction result data.

If one of these errors occurs, the error interrupt request resulting from the subtraction circuit is issued.

The conceptual image of the subtraction circuit is shown below.

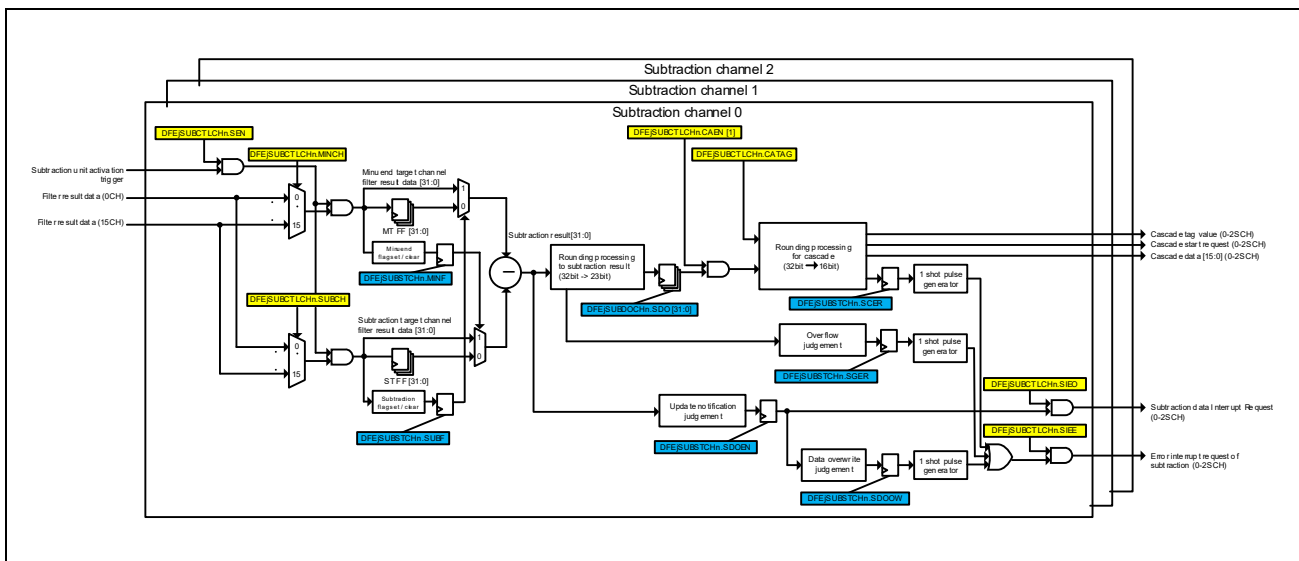


Figure 37.63 Configuration of Subtraction Circuit

37.5.11.2 Operation of the Subtraction Circuit

(1) Operation when Filtering Processing for Minuend Channel Is Executed Earlier

When the filtering process result for the channel set as a minuend is output, the filtering process result is held in the internal FF for storing minuend filtering result (hereafter called MTFF), and the minuend channel data input enabling bit (DFEjSUBSTCHn.MINF, hereafter called MINF) is set to 1. When the filtering process result for the channel set with SUBCH is output after 1 is set to MINF, the subtraction circuit performs subtraction processing of the data stored in MTFF and the filtering process result of the channel set with SUBCH, and stores the subtraction result in SDO. It also clears MTFF to ALL0, and clears MINF to 0. If subtraction is specified, this operation is executed repeatedly.

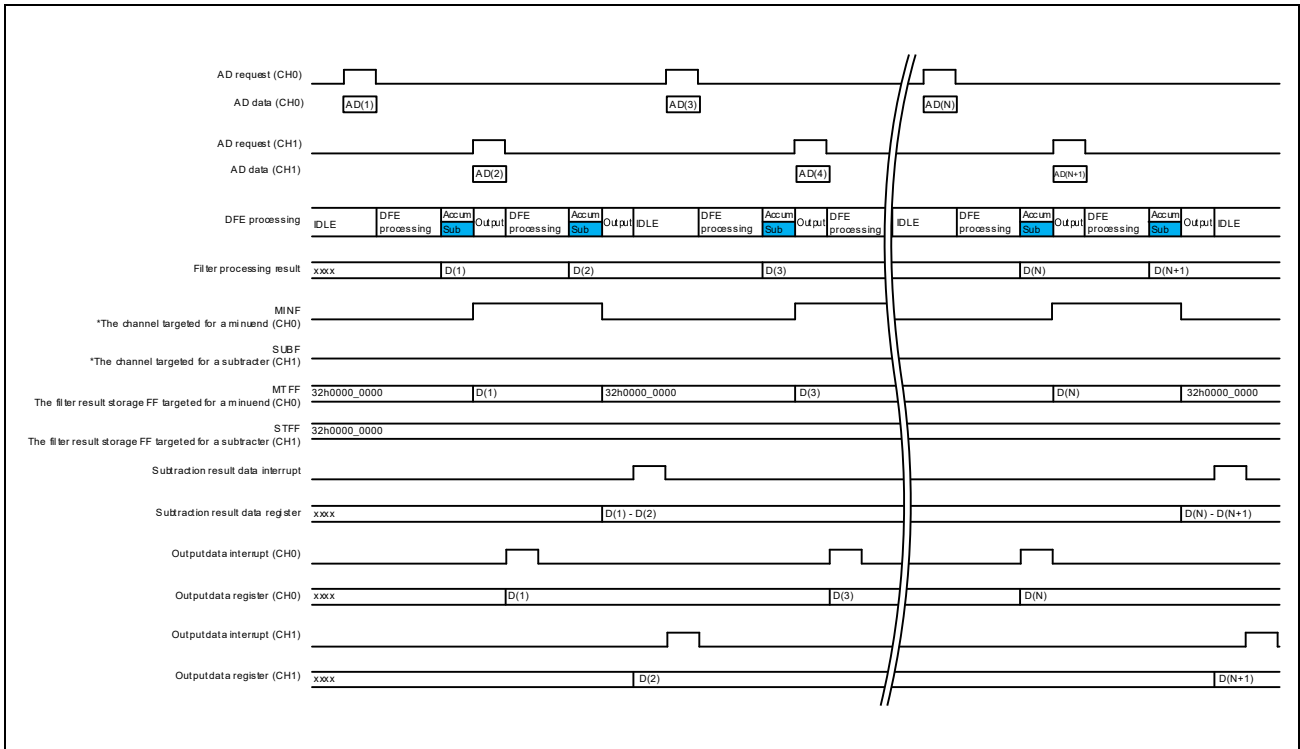


Figure 37.64 Subtraction Processing (When Processing for a Minuend Channel Is Executed Earlier)

(2) Operation when Filtering Processing for Subtrahend Channel is Executed Earlier

When the filtering process result for the channel set as a subtrahend is output, the filtering process result is held in the internal FF for storing subtrahend filtering result (hereafter called STFF), and the subtrahend channel data input enabling bit (DFEjSUBSTCHn.SUBF, hereafter called SUBF) is set to 1. When the filtering process result for the channel set with MINCH is output after 1 is set to SUBF, the subtraction circuit performs subtraction processing of the data stored in STFF and the filtering process result of the channel set with MINCH, and stores the subtraction result in SDO. It also clears STFF to ALL0, and clears SUBF to 0. If subtraction is specified, this operation is executed repeatedly.

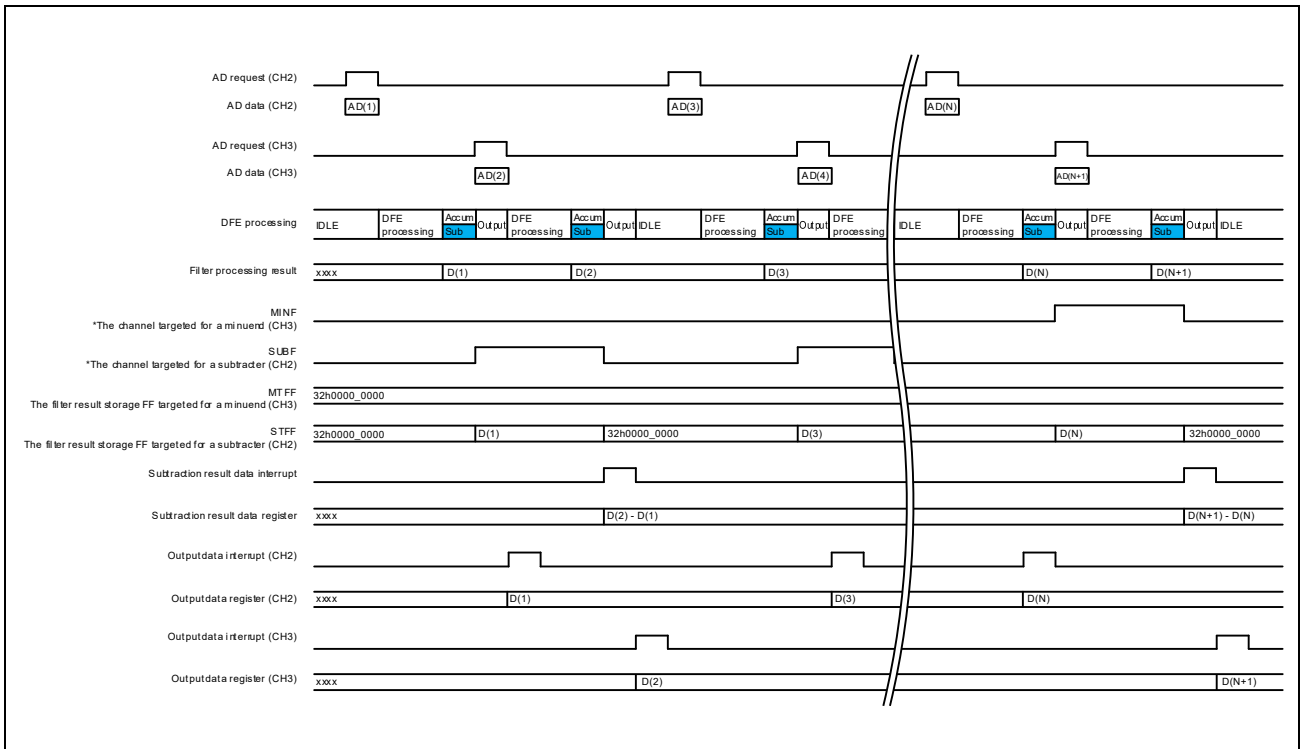


Figure 37.65 Subtraction Processing (when Processing for a Subtrahend Channel is Executed Earlier)

(3) Operation when Filtering Processing Order of Subtrahend and Minuend Channels is Not Constant

The subtraction circuit can perform subtraction processing regardless of the filter input order of the specified MINCH and SUBCH. The following example shows the operation for MINCH = channel 5 and SUBCH = channel 4.

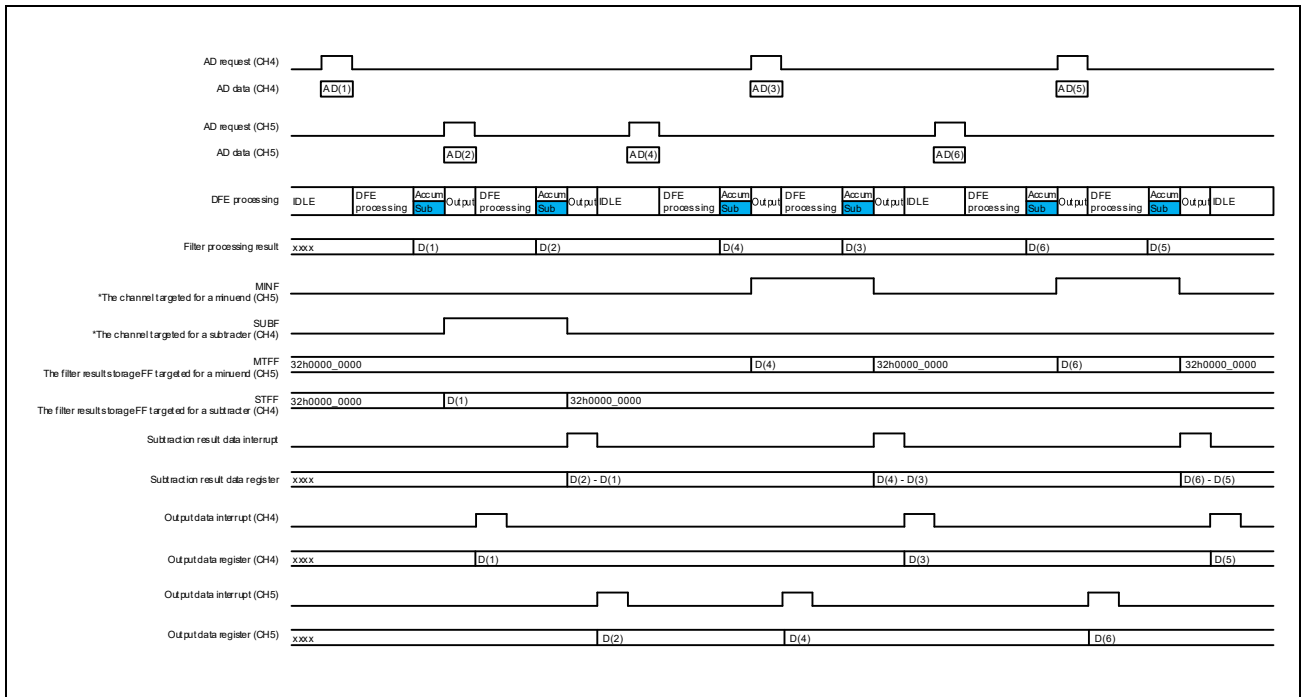


Figure 37.66 Subtraction Processing (when Order of Processing for Subtrahend/Minuend Channels is Not Constant)

(4) Operation when Filtering Processing for One Channel is Executed Continuously

When one channel set as a subtrahend or a minuend is processed continuously, the latest filtering process result is held in STFF or MTFF. When one channel is processed continuously, SUBF or MINF keeps holding 1.

After that, subtraction processing is executed on the basis of the latest filtering process result for both the minuend channel and the subtrahend channel.

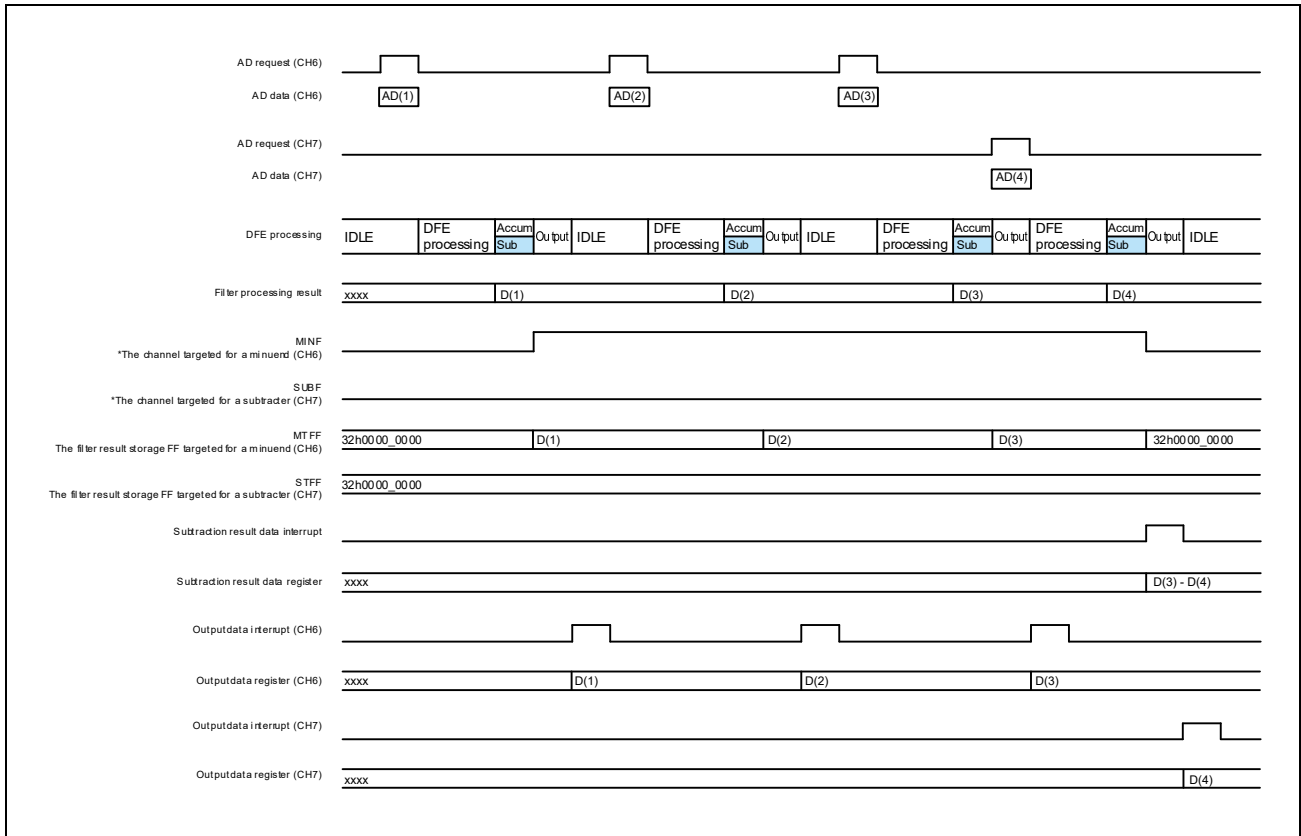


Figure 37.67 Subtraction Processing (when Processing for a Subtrahend Channel is Executed Continuously)

(5) Operation when Filtering Processing for Only One Channel is Executed and Finished

When one channel set as a subtrahend or a minuend is processed continuously and the other does not undergo filtering process, SUBF or MINF keeps holding 1.

Even if a subtraction ending flag is enabled and then subtraction processing ends, STFF, MTF, SUBF, and MINF keep holding the value.



Figure 37.68 Subtraction Processing (when Processing for Only a Subtrahend Channel is Executed and Finished)

37.5.11.3 Operation of Subtraction Circuit (with Subtraction Start Trigger)

To use subtraction processing, it is necessary to start filtering process after enabling the subtraction starting flag when subtraction is specified. Input the subtraction starting trigger by using a timer trigger or a software trigger, and enable the subtraction starting flag.

If the subtraction starting trigger is not input, subtraction processing is not executed during filter processing.

The subtraction circuit performs initialization during subtraction processing for subtrahend or minuend channel just after the subtraction starting flag is enabled. Specifically, the circuit initializes STFF or MTFF, which hold the filtering process result, subtrahend flag SUBF, and minuend flag MINF, and executes subtraction processing using the values as the initial data of subtraction processing.

In the following example, subtraction processing is not executed in filtering process of AD data AD (1) of CH6 and AD data AD (2) of CH7. The subtraction starting flag is enabled in filtering process of AD data AD (3) of CH6 after the subtraction starting trigger is input, and subtraction processing starts.

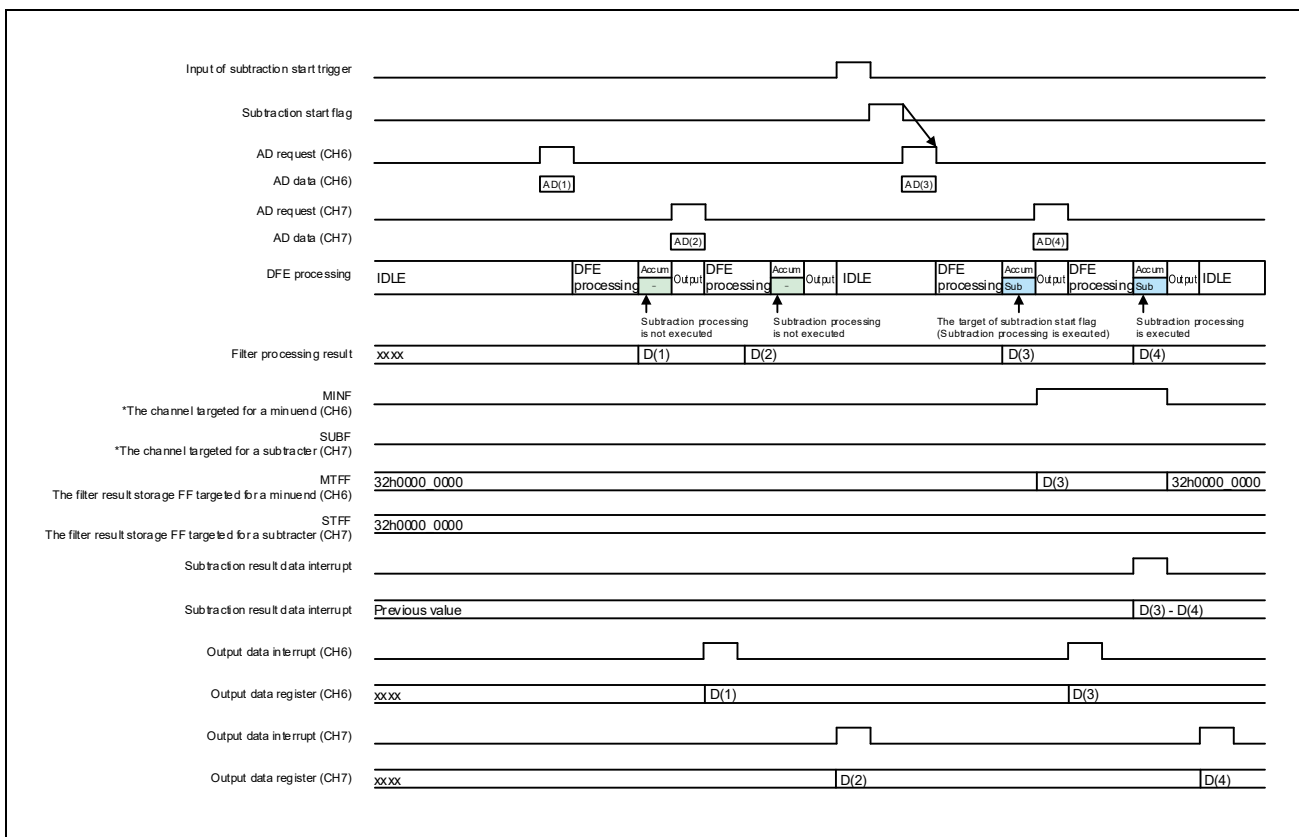


Figure 37.69 Subtraction Processing (Operation when Subtraction Starting Flag is enabled)

37.5.11.4 Operation of Subtraction Circuit (with Subtraction End Trigger)

To stop subtraction processing when subtraction is specified, it is necessary to start filtering process after enabling the subtraction ending flag. Input the subtraction ending trigger by using a timer trigger or a software trigger, and enable the subtraction ending flag.

Subtraction processing stops at the subtrahend or minuend channel just after the subtraction ending flag is enabled. After it stops, STFF, MTFF, which hold the filtering process result, subtrahend flag SUBF, and minuend flag MINF keep holding the value.

The following example shows the case where the subtraction ending trigger is input during filtering process of AD data AD (3) of CH6. The subtraction ending flag is enabled in filtering process of the AD data AD (4) of CH7, and subtraction processing of AD (4) is not executed.

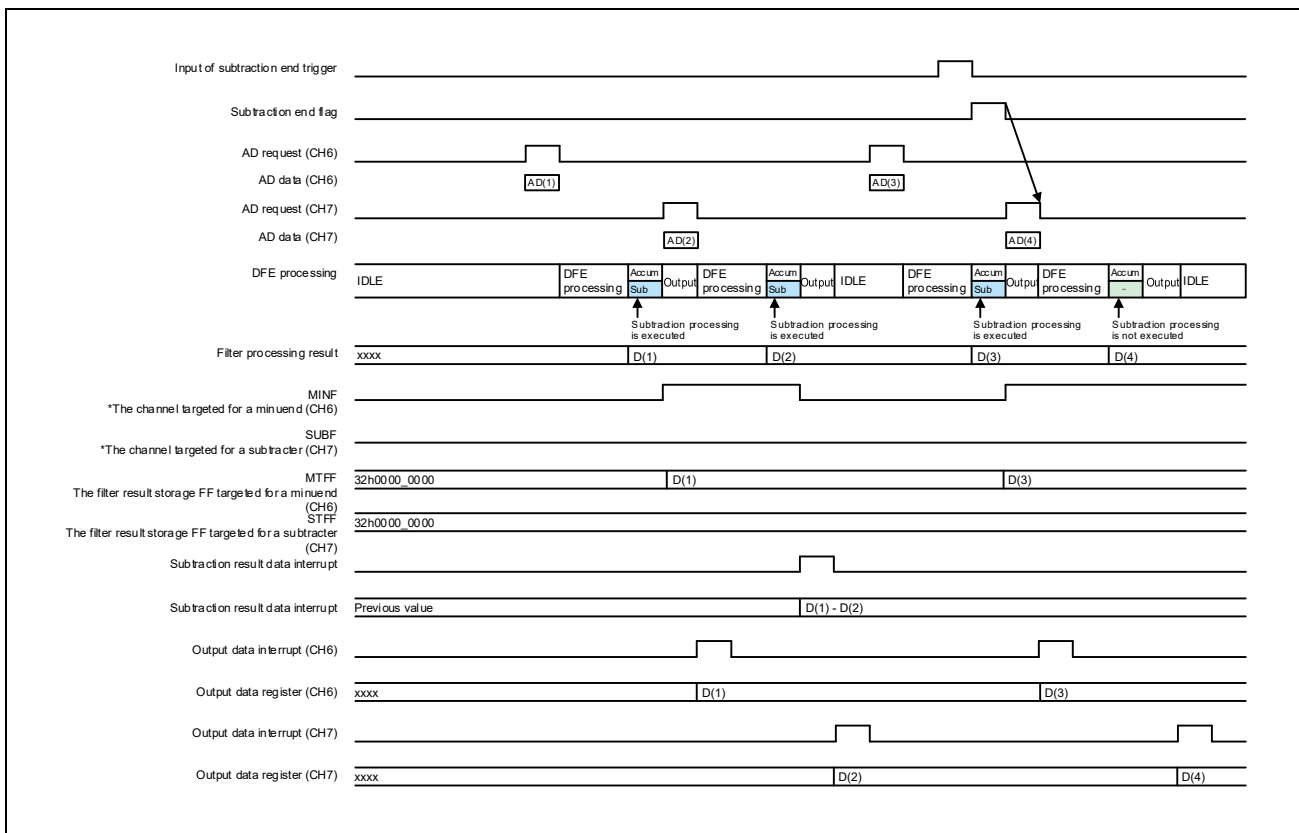


Figure 37.70 Subtraction Processing (Operation when Subtraction Ending Flag is enabled)

37.5.11.5 Operation of Subtraction Circuit (Stop by DFEjSUBCTLCHn.SEN)

By writing 0 to the subtraction channel enabling bit (DFEjSUBCTLCHn.SEN), it is possible to stop subtraction processing for the target of subtraction channel.

Before 0 is written to the subtraction channel enabling bit, subtraction processing is executed for the channel that has accepted an activation request or the channel undergoing filtering process. After 0 is written to the subtraction channel enabling bit, even if activation request of minuend/subtrahend channel has accepted, subtraction processing is not executed in this subtraction channel. The subtraction start/end flags which have been kept are discarded from this subtraction channel.

When subtraction processing in this subtraction channel is restarted, set the bit (DFEjSUBCTLCHn.SEN) to 1. The subtraction processing starts from filter processing after 1 is set to the subtraction channel enabling bit.

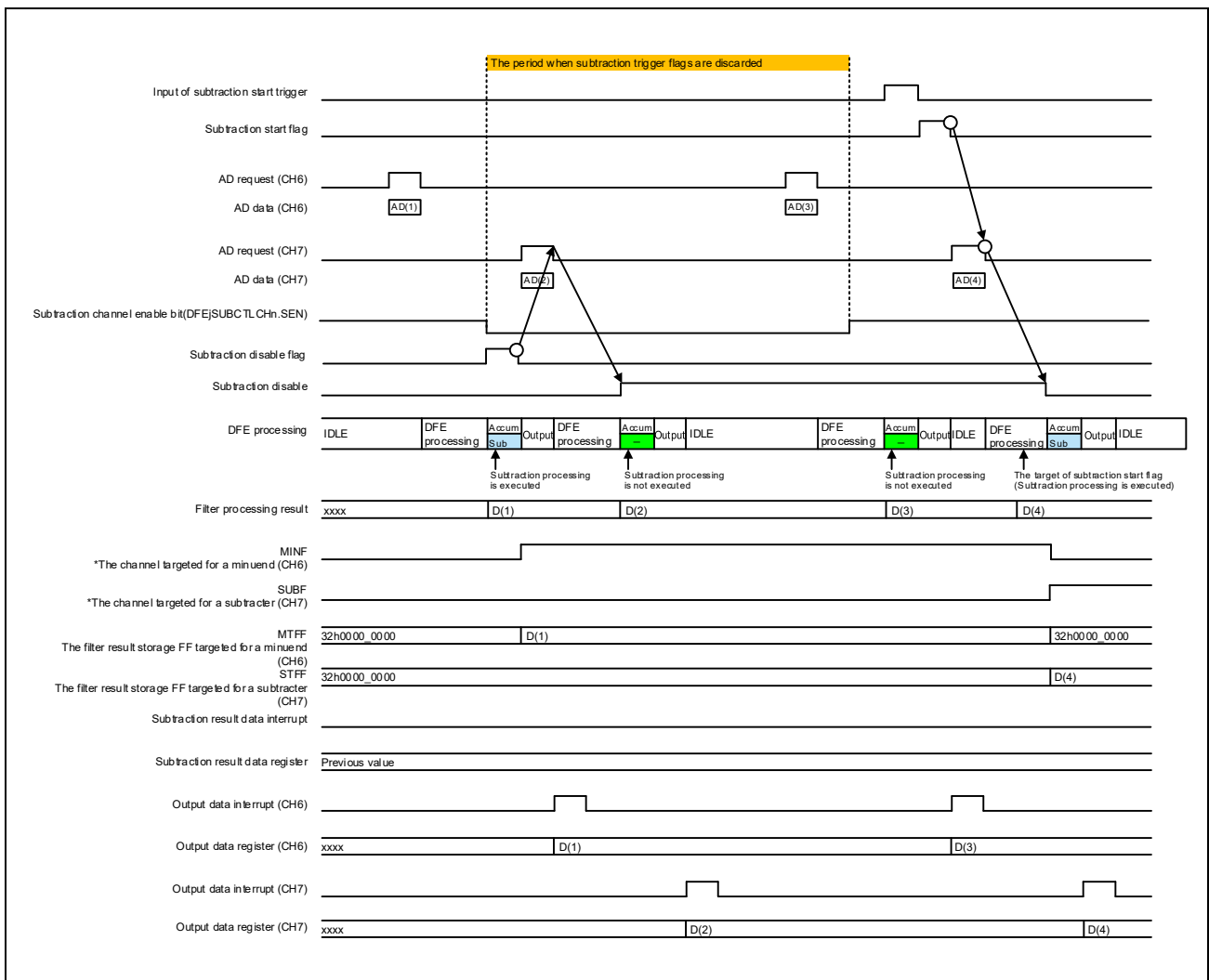


Figure 37.71 Subtraction Processing (Subtraction Processing Stopped by Clearing Subtraction Channel Enabling Bit (1/2))

In the following example, since the activation request for AD data AD (2) of CH7 is already accepted when 0 is written to the subtraction channel enabling bit, subtraction processing for this subtraction channel will be stopped after subtraction processing of AD (2) is executed.

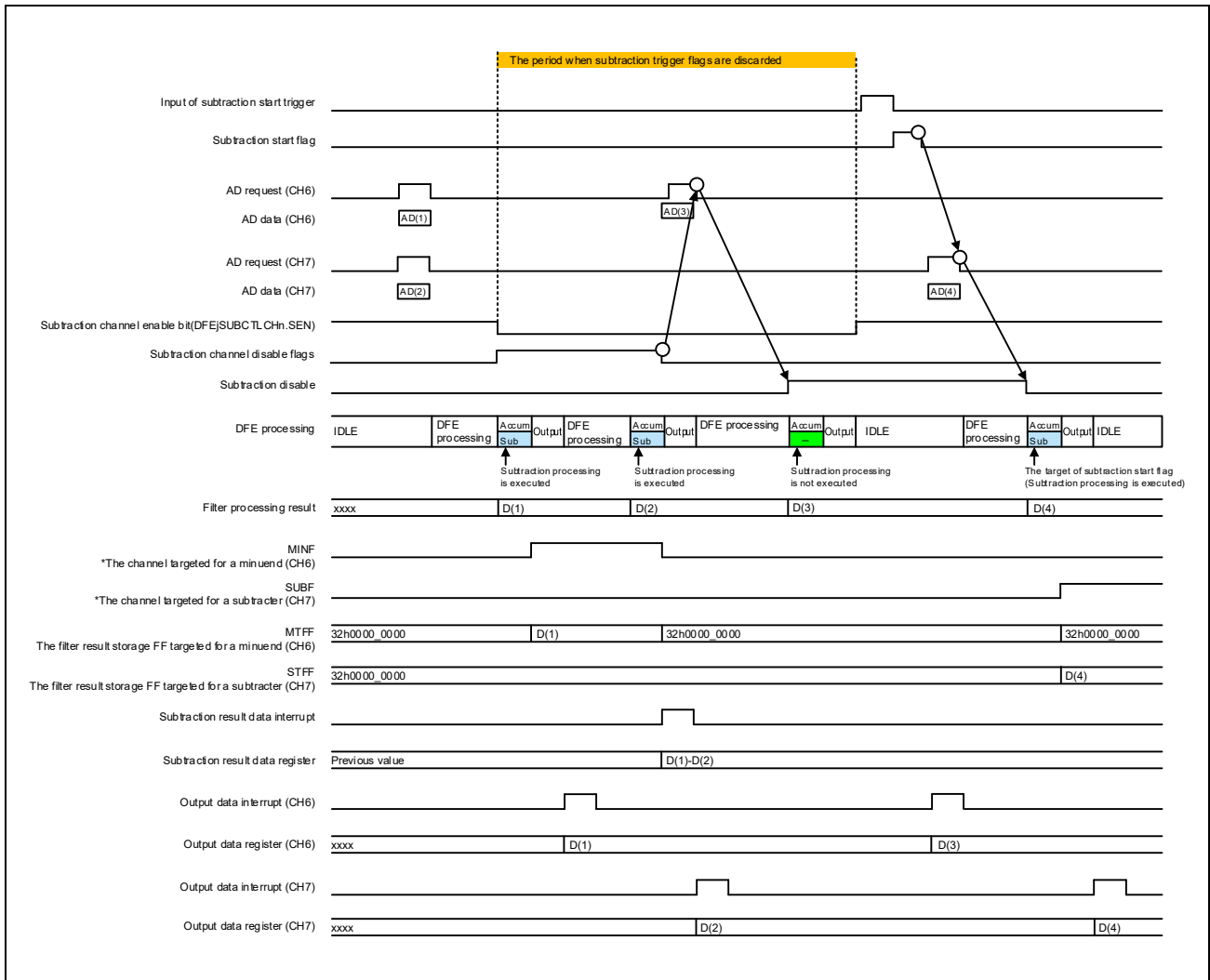


Figure 37.71 Subtraction Processing (Subtraction Processing Stopped by Clearing Subtraction Channel Enabling Bit (2/2))

37.5.12 Peak-Hold Circuit

The peak-hold circuit executes peak-hold processing and comparison processing for data from the accumulation circuit outputs.

According to the setting of DFEjCTLBCHn.PRC SB in the table below, three types of processing can be selected for the peak-hold circuit: (1) Only peak-hold processing executed, (2) Only comparison processing executed, or (3) Both peak-hold processing and comparison processing executed.

If DFEjCTLBCHn.PRC SB = 11_B and DFEjPH23CCTL0.PEN = 1 at the same time, the peak-hold 23 function, which is an extended function for PH processing, can be selected.

Peak-hold Circuit Setting	DFEjCTLBCHn.PRC SB	DFEjPH23CCTL0.PEN
Peak-hold processing and comparison processing prohibited	00 _B	—
Only peak-hold executed	01 _B	—
Only comparison executed	10 _B	—
Both peak-hold processing and comparison processing executed	11 _B	0
Peak-hold 23 function executed	11 _B	1*1

Note 1. To enable the peak-hold 23 function, it is necessary to set the target channel in DFEjPH23CCTL0.CHS.

The peak-hold circuit executes peak-hold processing or comparison processing for data that is output from the accumulation circuit. When the accumulation/decimation count is specified, peak-hold processing is executed on each specified count.

In peak-hold processing, peak-hold results are written to the peak-hold result register. A condition match interrupt is generated at the end of peak-hold processing. In comparison processing, accumulation circuit output data is compared with the values of the comparison target register. When the comparison result is true, a condition match interrupt is generated.

37.5.12.1 Summary of Peak-Hold Processing

In peak-hold processing, the upper peak or the lower peak can be selected by the peak-hold peak selection bit (DFEjCTLBCHn.PHPS).

When the upper peak is selected, data that is input to the peak-hold circuit is compared with the data that has been input to the peak-hold circuit for each channel, and the larger value is written to the peak-hold result register as a peak-hold result.

When the lower peak is selected, data that is input to the peak-hold circuit is compared with the data that has been input to the peak-hold circuit for each channel, and the smaller value is written to the peak-hold result register as a peak-hold result.

The initial peak-hold value can be selected by using the SELB2 bits in DFEjCTLBCHn. The peak-hold circuit executes peak-hold processing from the time the peak-hold initialization flag is valid until the peak-hold end flag is valid. The peak-hold circuit does not execute peak-hold processing from the time the peak-hold end flag is valid until the peak-hold initialization flag is valid. The peak-hold index register indicates how much data has been stored in the peak-hold result register since the start of peak-hold processing.

37.5.12.2 Operation of Peak-Hold Processing

Peak-hold processing proceeds when $DFEjCTLBCHn.PRCsB = 01_B$ (specifying peak-hold operation).

Peak-hold processing starts from the next input of data following the peak-hold initialization flag becoming valid.

Peak-hold processing ends from the next input of data following the peak-hold end flag becoming valid. After this, the peak-hold initialization flag must again become valid for peak-hold processing to be restarted.

Alternate between input to the peak-hold initialized and end flags. Operations for peak-hold processing are not guaranteed in case of consecutive input to either flag.

Values input to the peak-hold circuit are compared with the peak-hold initialization register or the peak-hold result register. The condition for updating the peak-hold value when (1) the upper peak is selected or (2) the lower peak is selected is shown below. When the condition is satisfied, the value input to the peak-hold circuit is stored in the peak-hold result register.

Notifications for peak-hold updating are issued according to the settings of the peak-hold updated notification setting register.

Peak	Function	Number Times of PH	Issue Condition
Upper peaks	PH update notification	First time*1	PH circuit input data*2 > The value of the peak-hold initial value setting register*3
		After the first time	PH circuit input data*2 > The value of the DFEjPHCHn*4
Lower peaks	PH update notification	First time*1	PH circuit input data*2 < The value of the peak-hold initial value setting register*3
		After the first time	PH circuit input data*2 < The value of the DFEjPHCHn*4

Note 1. The first time means when filter processing is executed after PH initialization flag is valid.

Note 2. PH input data is the output data of the filter circuit or accumulation/decimation circuit.

Note 3. The register means one of the DFEjPHIA-DFEjPHID registers.

Note 4. These values are compared in either fixed-point format or integer format regardless of the DFEjCTLBCHn.PFMT bit setting.

When the next value is input after the peak-hold initialization flag is valid, and when the value does not satisfy the condition for updating the peak-hold value, the value of the peak-hold initialization register is stored in the peak-hold result register, and the notification of peak-hold updating is not issued.

Peak	Function	Number Times of PH	Update Condition	Stored Data in DFEjPHCHn
Upper peaks	Peak-Hold Result register	First time* ¹	PH circuit input data* ² > The value of the peak-hold initial value setting register* ³	PH circuit input data
			PH circuit input data* ² ≤ The value of the peak-hold initial value setting register* ³	The value of the peak-hold initial value setting register* ³
		After the first time	PH circuit input data* ² > The value of the DFEjPHCHn* ⁴	PH circuit input data
			PH circuit input data* ² ≤ The value of the DFEjPHCHn* ⁴	Holds the value of DFEjPHCHn
Lower peaks	Peak-Hold Result register	First time* ¹	PH circuit input data* ² < The value of the peak-hold initial value setting register* ³	PH circuit input data
			PH circuit input data* ² ≥ The value of the peak-hold initial value setting register* ³	The value of the peak-hold initial value setting register* ³
		After the first time	PH circuit input data* ² < The value of the DFEjPHCHn* ⁴	PH circuit input data
			PH circuit input data* ² ≥ The value of the DFEjPHCHn* ⁴	Holds the value of DFEjPHCHn

Note 1. The first time means when filter processing is executed after PH initialization flag is valid.

Note 2. PH input data is the output data of the filter circuit or accumulation/decimation circuit.

Note 3. The register means one of the DFEjPHIA-DFEjPHID registers.

Note 4. These values are compared in either fixed-point format or integer format regardless of the DFEjCTLBCHn.PFMT bit setting.

37.5.12.3 Operation of Peak-Hold Processing (Trigger Flag is Enabled)

Operations for peak-hold initialization are shown below.

The execution for the peak-hold processing of values from the given ADC starts once the peak-hold initialization flag becomes valid.

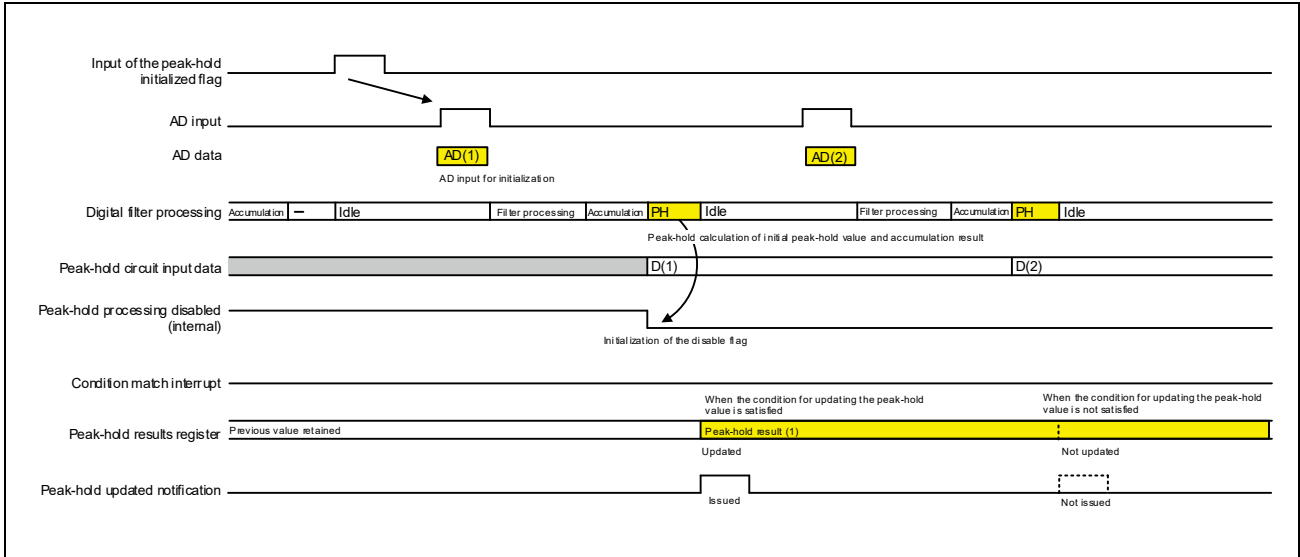


Figure 37.72 Operations for Peak-Hold Initialization

Operations for ending peak-hold are shown below. The execution for the peak-hold processing of values from the given ADC does not start when the peak-hold end flag becomes valid.

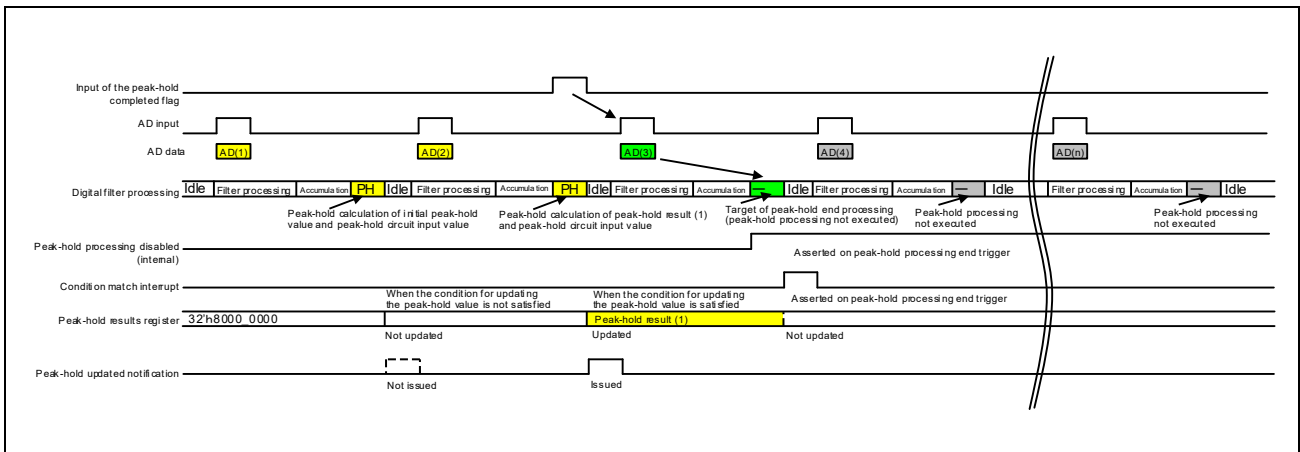


Figure 37.73 Operations for Ending Peak-Hold Processing

The effective range of peak-hold initialization/end flags in each mode of peak-hold circuit is explained below.

The peak-hold processing is always affected by these flags.

The comparison processing is not affected by these flags except for the peak-hold 23 mode. Comparison is performed without being affected by these flags for all modes except peak-hold 23.

For the peak-hold stop status in peak-hold 23 mode, neither the peak-hold processing nor the comparison processing is executed.

Peak-Hold Circuit Setting	Peak-Hold Initialization/End Flag	
	Peak-Hold Processing	Comparison Processing
Peak-hold processing and comparison processing prohibited	—	—
Only peak-hold executed	Valid	—
Only comparison executed	—	Invalid
Both peak-hold processing and comparison processing executed	Valid	Invalid
Peak-hold 23 function executed	Valid	Valid

The effective range of Peak-Hold Processing Disable bit (DFEjCTLBCHn.DISB) in each mode of peak-hold circuit is explained below.

The peak-hold processing is always affected by this bit.

The comparison processing is not affected by this bit except for the peak-hold 23 mode. Comparison is performed without being affected by this bit for all modes except peak-hold 23.

For the peak-hold stop status by this bit in peak-hold 23 mode, neither the peak-hold processing nor the comparison processing is executed.

Peak-Hold Circuit Setting	Start/Stop by DISB	
	Peak-Hold Processing	Comparison Processing
Peak-hold processing and comparison processing prohibited	—	—
Only peak-hold executed	Valid	—
Only comparison executed	—	Invalid
Both peak-hold processing and comparison processing executed	Valid	Invalid
Peak-hold 23 function executed	Valid	Valid

37.5.12.4 Operation of Peak-Hold Mask Processing (Trigger Flag is Enable)

The peak-hold mask start/end operation is described below. If the timer trigger set as the peak-hold mask start flag is input, it is held inside the DFE as the peak-hold mask start flag. In the next filtering process, the peak-hold mask start flag is enabled and peak-hold processing stops. Be aware that, during peak-hold mask, peak-hold processing is stopped but the internal peak-hold counter works without stopping.

If the timer trigger set as the peak-hold mask end flag is input, it is held inside the DFE as the peak-hold mask end flag. When peak-hold processing in the next filtering process is executed, the peak-hold mask end flag is enabled and peak-hold processing resumes.

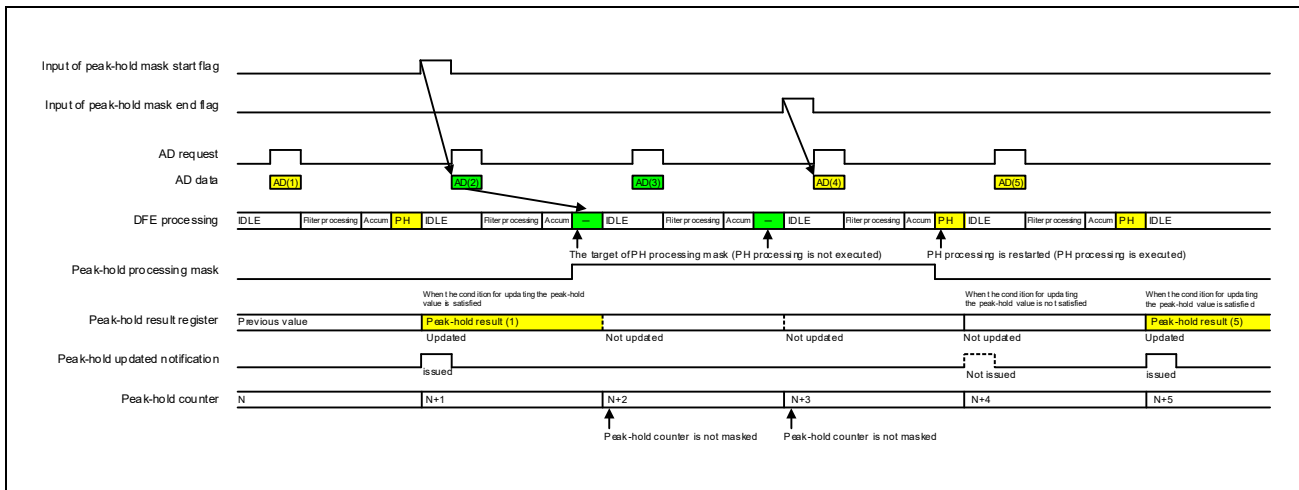


Figure 37.74 Peak-hold Masking Operation

To mask peak-hold processing for a specified period from the start of peak-hold processing, make the settings as follows. Before the filter processing where the peak-hold initialization flag is enabled is started, input the timer trigger set as the peak-hold mask start flag. At this time, both the peak-hold initialization flag and the peak-hold mask start flag are held inside the DFE. When filter processing is started with the peak-hold initialization flag and the peak-hold mask start flag held, peak-hold processing is not executed but remains stopped. Then, the peak-hold counter counts it but the peak-hold index register is hold previous value.

To resume peak-hold processing, enable the peak-hold mask end flag. The peak-hold initialization flag is enabled when the filter processing with the peak-hold mask end flag enabled is executed. Therefore, peak-hold processing is executed from this filter processing, and the value of peak-hold circuit input data and the value of peak-hold initial value register are compared with each other. Then, the peak-hold counter is updated, too.

In both examples (1) and (2) below, the peak-hold processing can be masked from the start of peak-hold processing.

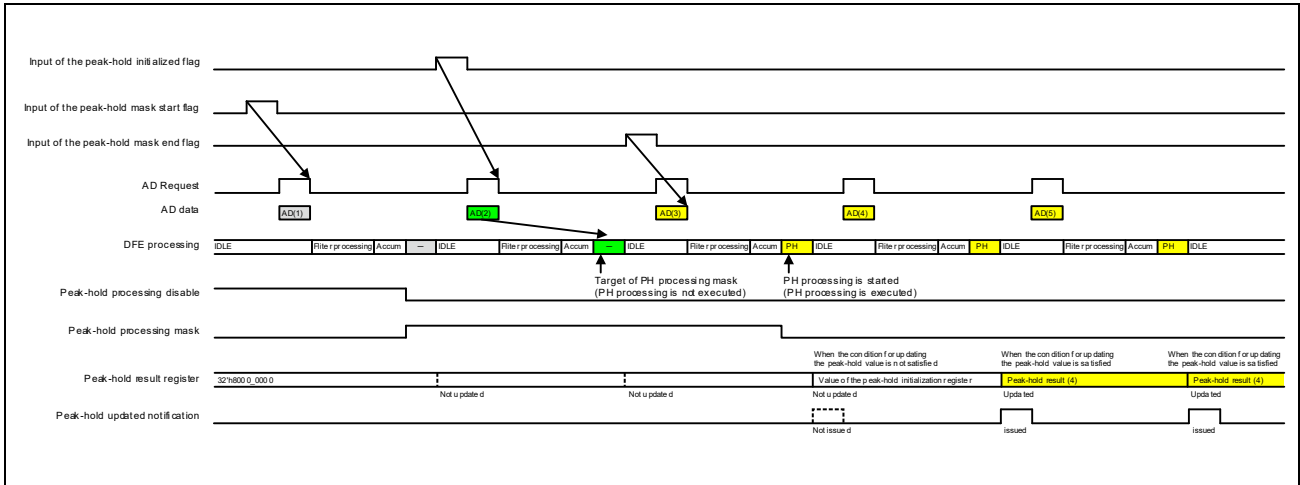


Figure 37.75 Peak-Hold Mask Processing for Specified Period from Start of Peak-Hold Processing (1/2)

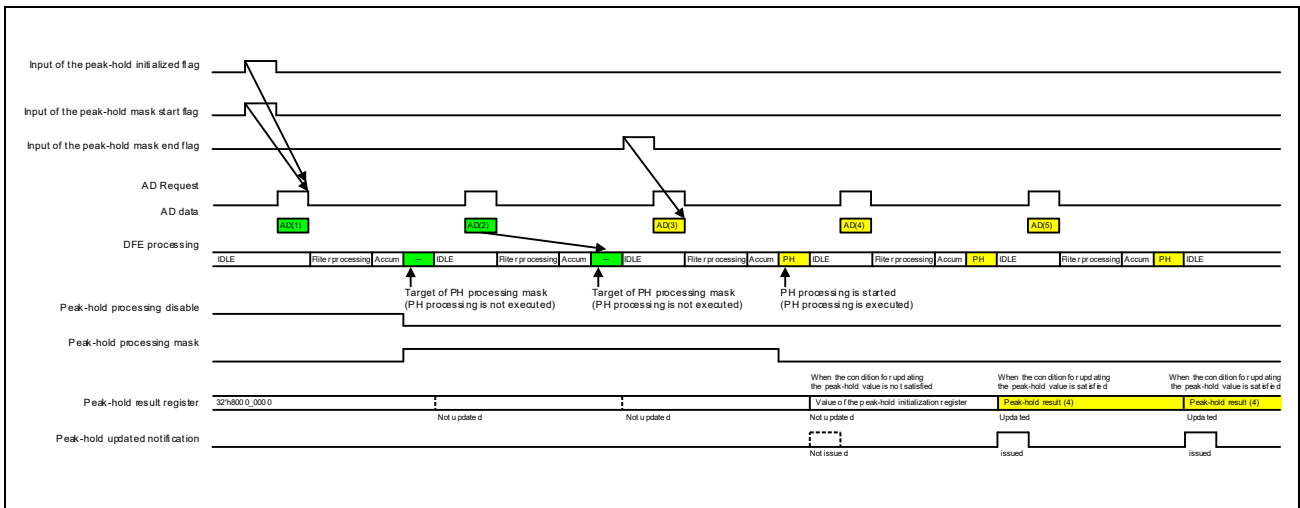


Figure 37.75 Peak-Hold Mask Processing for Specified Period from Start of Peak-Hold Processing (2/2)

The effective range of peak-hold mask start/end flags in each mode of peak-hold circuit is explained below.

The peak-hold processing is always affected by these flags.

Comparison is performed without being affected by these flags for all modes except peak-hold 23.

For the peak-hold mask status in peak-hold 23 mode, neither the peak-hold processing nor the comparison processing is executed.

Peak-Hold Circuit Setting	Peak-Hold Mask Start/End Flag	
	Peak-Hold Processing	Comparison Processing
Peak-hold processing and comparison processing prohibited	—	—
Only peak-hold executed	Valid	—
Only comparison executed	—	Invalid
Both peak-hold processing and comparison processing executed	Valid	Invalid
Peak-hold 23 function executed	Valid	Valid

37.5.12.5 Operation of Comparison Processing

In comparison processing, the peak-hold circuit input data is compared with the values of comparison target registers (CPA to CPD) for each channel. When the comparison result is true, the CND flag (condition match flag) is asserted and a condition match interrupt request is issued. Comparison processing is not affected by trigger flags except for the peak-hold 23 mode.

The following comparison calculations can be performed.

Comparison	Description
Equal to (=)	"Equal to (=)" calculation is performed between the peak-hold circuit input data and the value selected for comparison. When the comparison result is "equal to", it is true.
Equal to or less than (\leq)	"Equal to or less than (\leq)" calculation is performed between the peak-hold circuit input data and the value selected for comparison. When the comparison result is "equal to or less than", it is true.
Equal to or more than (\geq)	"Equal to or more than (\geq)" calculation is performed between the peak-hold circuit input data and the value selected for comparison. When the comparison result is "equal to or more than", it is true.
Less than (<)	"Less than (<)" calculation is performed between the peak-hold circuit input data and the value selected for comparison. When the comparison result is "less than", it is true.
More than (>)	"More than (>)" calculation is performed between the peak-hold circuit input data and the value selected for comparison. When the comparison result is "more than", it is true.

Once the comparison calculation result is true and the condition match bit (CND in DFEjSTCHn) is set to 1, the condition match bit remains 1 even if the subsequent comparison calculation result is false. Use the clear status register to clear the condition match bit to 0.

37.5.12.6 Operation of the Peak-Hold Index Register

When peak-hold processing is enabled (by setting DFEjCTLBCHn.PRCsB to 01_B), the peak-hold counter in the digital filter is cleared after the first round of peak-hold processing following input of the peak-hold initialization flag. After that, the peak-hold counter is incremented every time a value is peak-hold processed (whether or not it satisfies the condition for updating the peak-hold value).

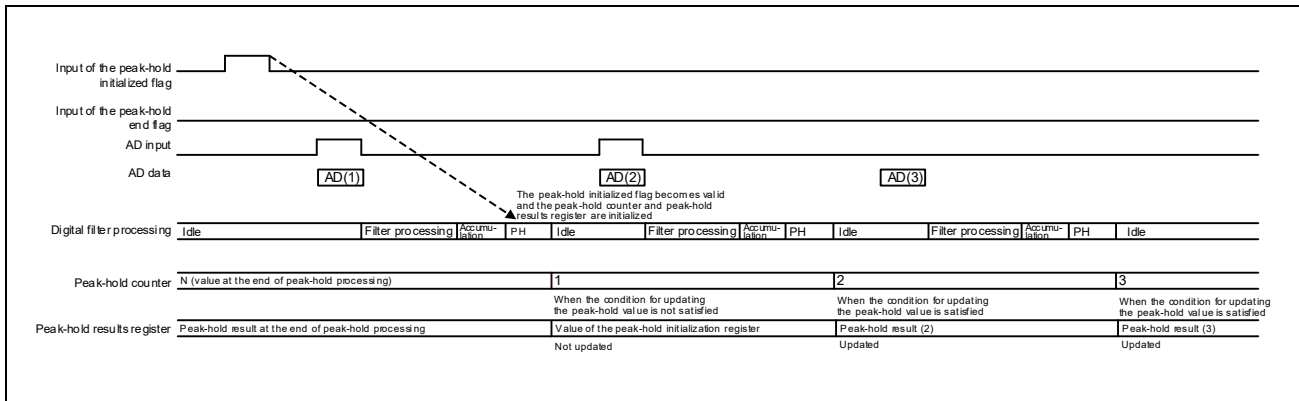


Figure 37.76 Operation of the Peak-Hold Counter at the Start of Peak-Hold Processing

Input of the peak-hold end flag leads to the end of peak-hold processing following input of the next value for peak-hold processing. If the peak-hold end flag becomes valid, the peak-hold counter and results register retain their previous values but do not update in response to data from the ADC. The values are retained until the peak-hold initialization flag becomes valid again.

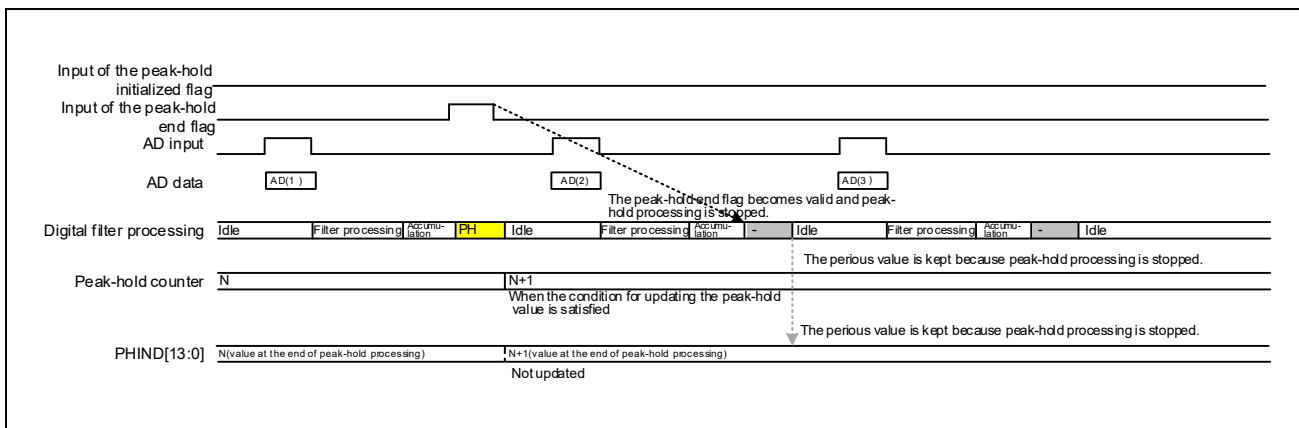


Figure 37.77 Operation of the Peak-Hold Counter at the End of Peak-Hold Processing

If the peak-hold update condition is satisfied, updating the peak-hold result register stores the peak-hold counter value to the PHIND bits of the peak-hold index register when the peak-hold counter is incremented.

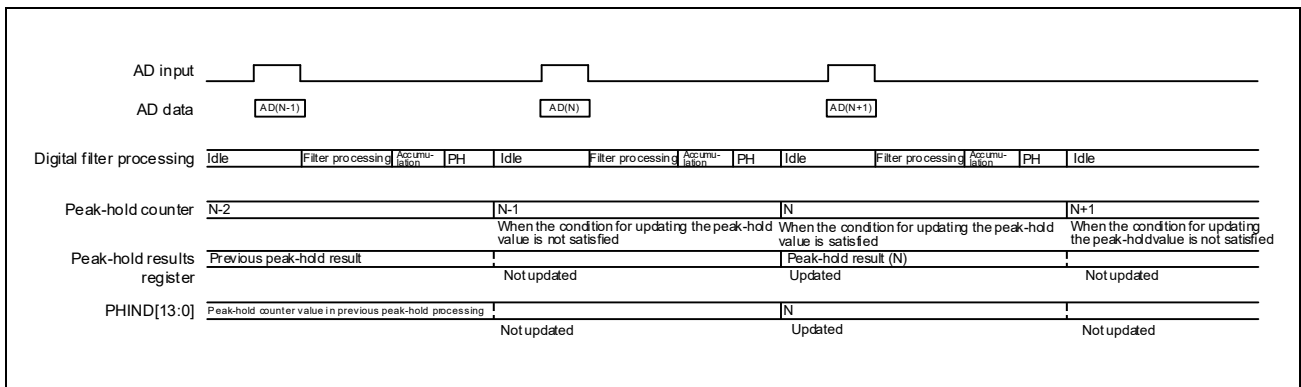


Figure 37.78 Peak-Hold Counter and Peak-Hold Index Register

When 16384 rounds of peak-hold processing are completed, the PHIOF bit in the peak-hold index register is set to 1. At this time, the value of the peak-hold counter becomes 0. Since the value of the peak-hold counter is stored in the PHIND bits when the input value satisfies the condition for updating the peak-hold value, the value of the PHIND bits becomes 0 if the condition is satisfied, but the previous value is retained if it is not.

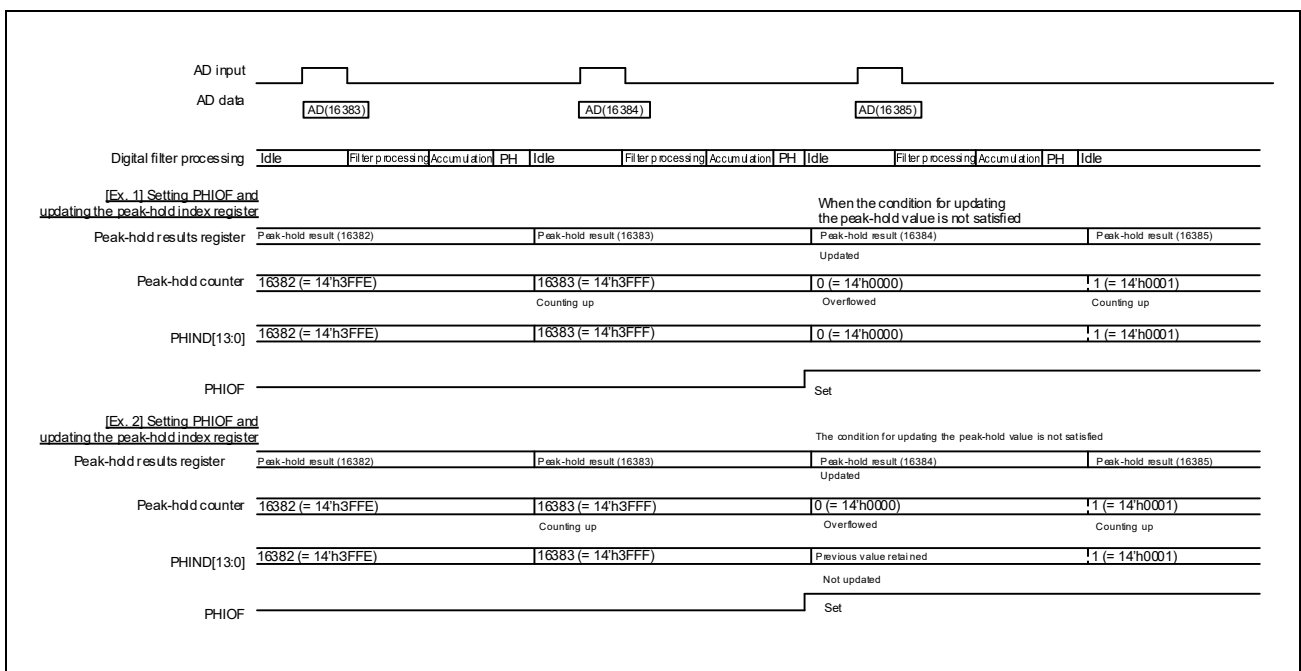


Figure 37.79 Setting of the Peak-Hold Index Overflow Bit

37.5.12.7 Reading of the Peak-Hold Index Register and Operation

Read the peak-hold index register between the end of peak-hold processing and the next time it starts (i.e. while peak-hold processing is disabled).

Operation of the peak-hold index register differs according to whether or not it is read.

Operations in peak-hold index updating and peak-hold retention modes also differ if the peak-hold index register is not read.

(1) When the Register is Read while Peak-Hold Processing is Disabled

The following applies whether the mode is peak-hold index updating or peak-hold index retention.

The PHIND and PHIOF bits are cleared when the first value for peak-hold processing is handled after peak-hold processing starts. The value of the PHIOW bit does not change.

The value of the peak-hold counter becomes 1, as does the value of the PHIND bits.

The peak-hold result register is not affected by whether reading or not reading peak-hold index register. Refer to **Section 37.5.12.2.** for the details.

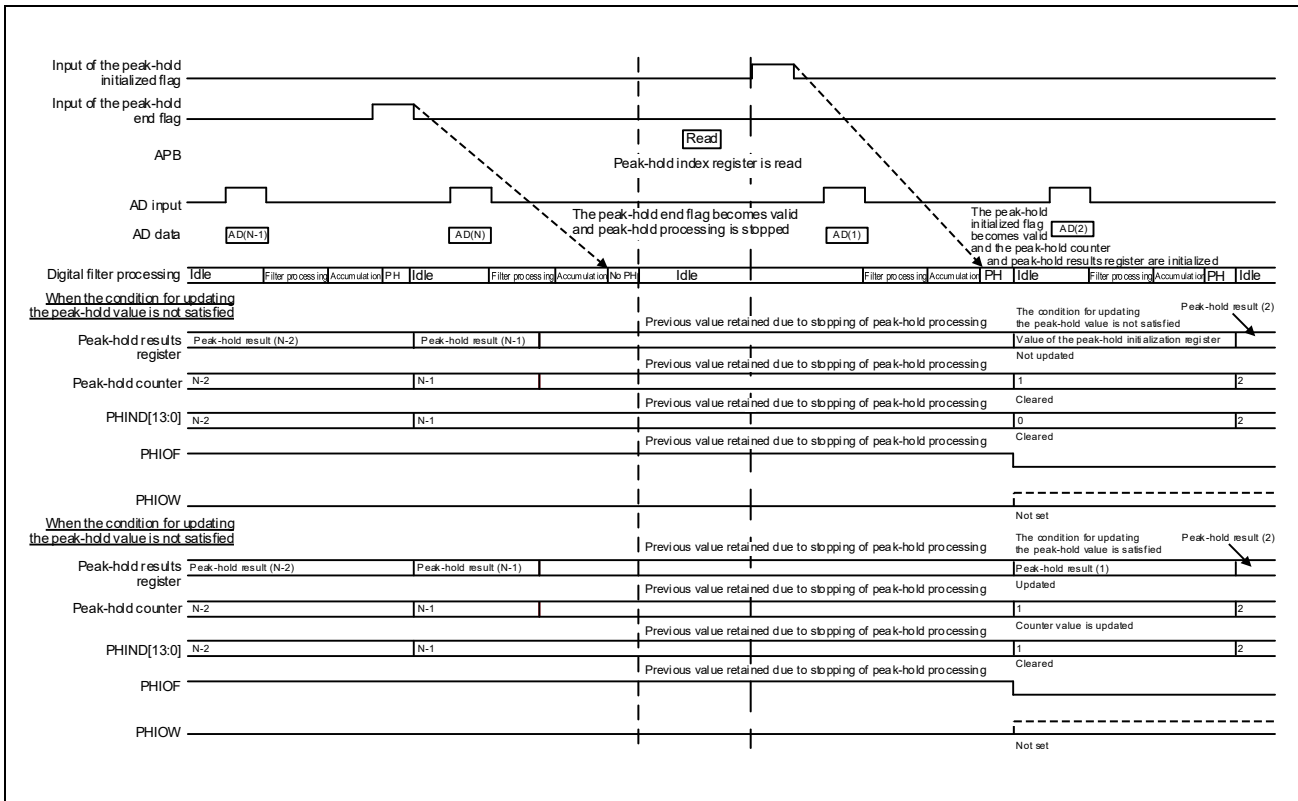


Figure 37.80 Operation of the Peak-Hold Index Register when it is Read

(2) When the Register is Not Read while Peak-Hold Processing is Disabled

- Peak-hold index update mode (DFEjCTLBCHn.PICS = 0):

The PHIND and PHIOF bits are cleared when the first value for peak-hold processing is handled after peak-hold processing starts and the PHIOW bit is set.

The value of the peak-hold counter becomes 1, as does the value of the PHIND bits.

To clear the PHIOW bit, write 1 to DFEjCLRSTCHn.CLRPHIOW.

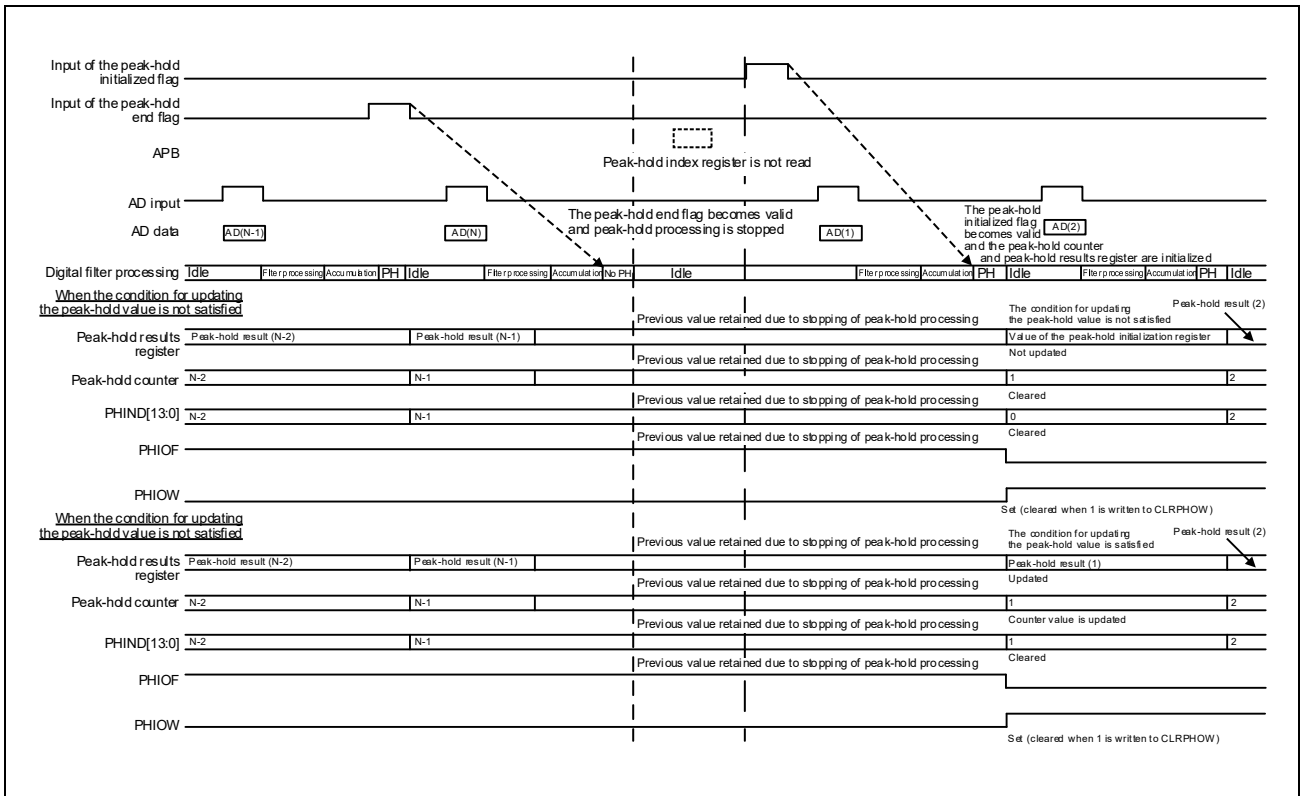


Figure 37.81 Operation of the Peak-Hold Index Register When It is Not Read (in Updating Mode)

- Peak-hold index retention mode (DFEjCTLBCHn.PICS = 1):

The values of the PHIND and PHIOF bits are retained the next time peak-hold processing starts. The value of PHIOW does not change.

If the register is read following the next and subsequent times peak-hold processing is ended, starting peak-hold processing again leads to clearing of the PHIND and PHIOF bits (refer to **(1) When the Register is Read while Peak-Hold Processing is Disabled**).

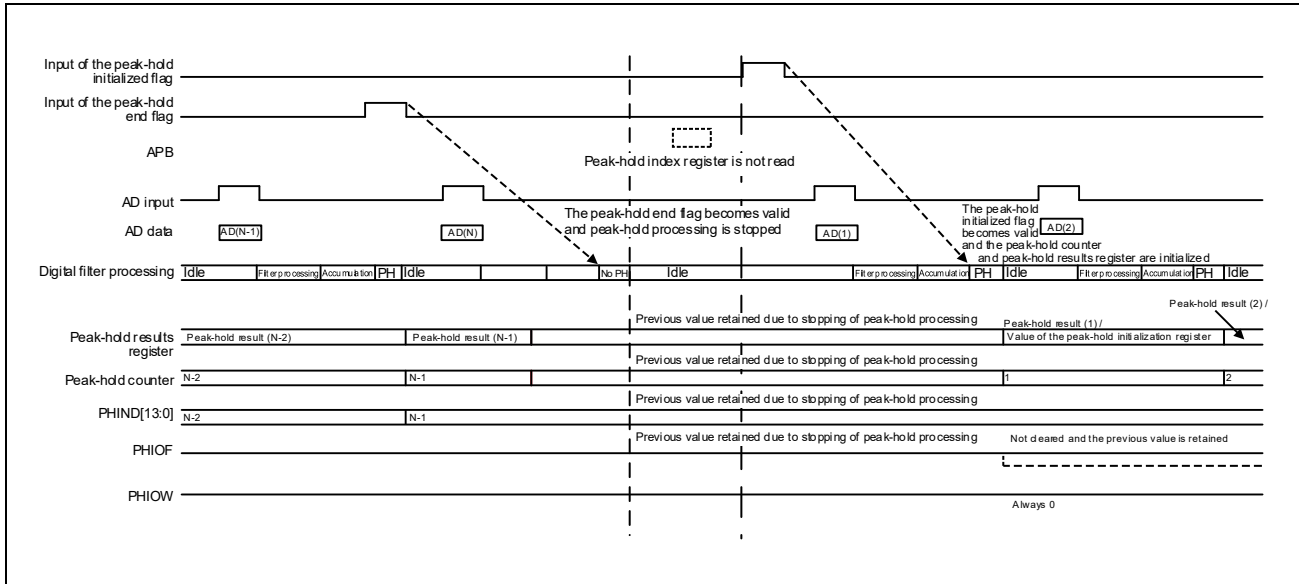


Figure 37.82 Operation of the Peak-Hold Index Register when it is Not Read (in Retention Mode)

37.5.12.8 Operation When Both Peak-Hold Processing and Comparison Processing Are Executed

When the peak-hold circuit processing select bit (DFEjCTLBCHn.PRC SB) is 11_B and the peak-hold 23 function enable bit (DFEjPH23CCTL0.PEN) is 0, peak-hold processing and comparison processing can be executed simultaneously.

In this mode, comparison processing is always executed. Peak-hold processing has a valid processing period based on the peak-hold initialization and end flags, and a peak-hold stop period based on the peak-hold mask start and end flags.

In peak-hold processing, the peak-hold end interrupt request is issued in the filter processing where the peak-hold end flag is enabled.

In comparison processing, the condition match interrupt request is issued when the comparison condition is true.

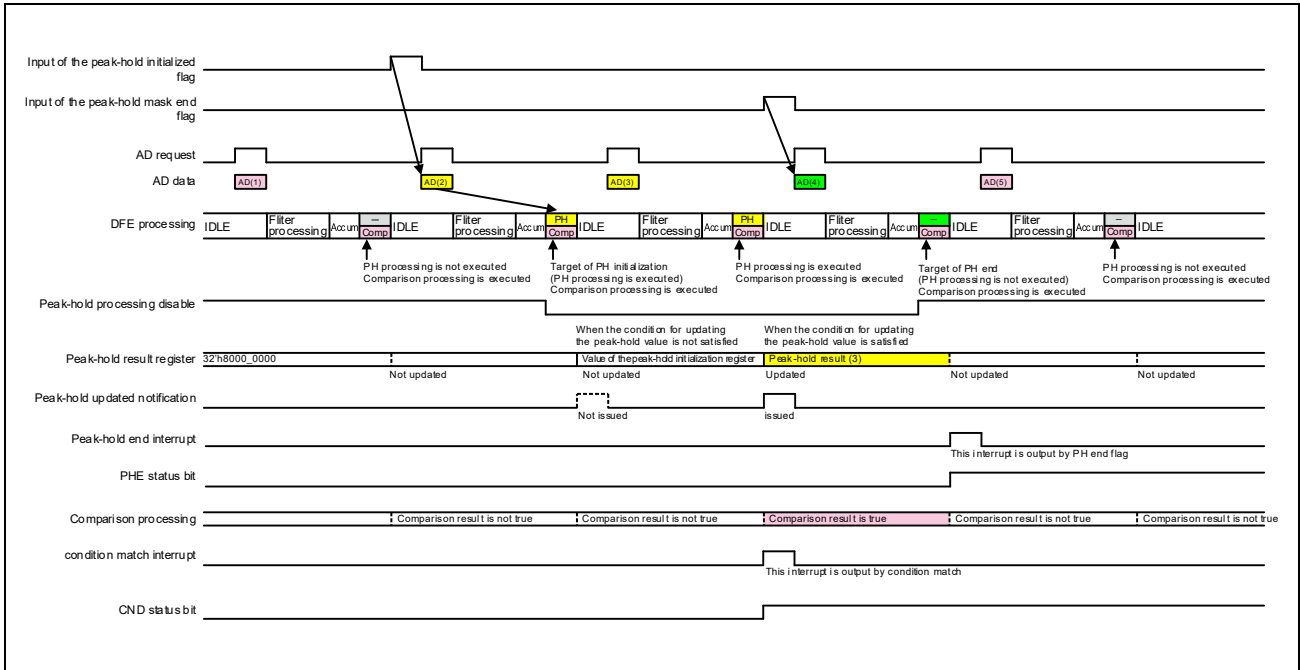


Figure 37.83 Operation when Peak-Hold Processing and Comparison Processing Are Executed Simultaneously (Peak-Hold Initialization/End Flag Enabled)

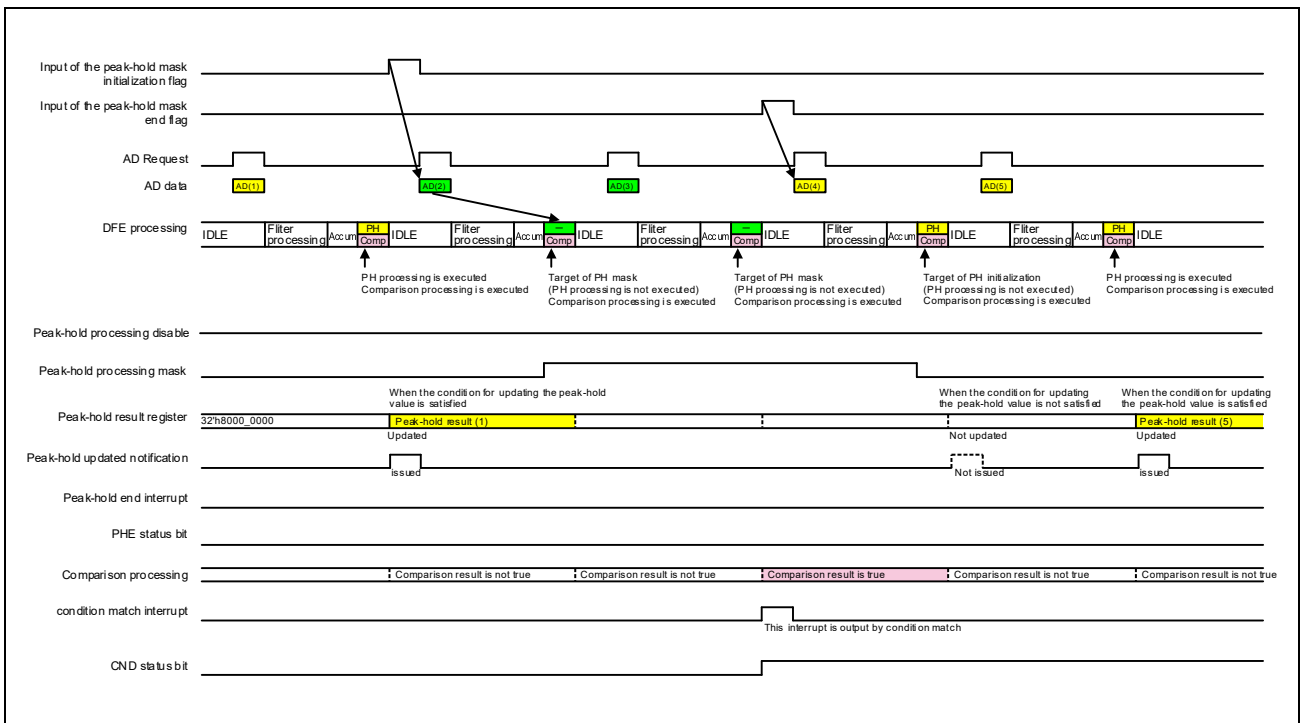


Figure 37.84 Operation when Peak-Hold Processing and Comparison Processing Are Executed Simultaneously (Peak-Hold Mask Start/End Flag Enabled)

37.5.12.9 Summary of Peak-Hold-23 Processing

The peak-hold-23 function is an extended function for peak-hold processing. The peak-hold-23 function allows you to detect the three peak points of the filtering process result.

Peak-hold processing is executed for the channel targeted for the peak-hold-23 function. Transitions from peak-hold-1 processing to peak-hold-2 processing and from peak-hold-2 processing to peak-hold-3 processing are realized in comparison processing. In this way, it is possible to hold the three peak-hold results by switching between peak-hold-1 processing, peak-hold-2 processing, and peak-hold-3 processing.

Table 37.84 Overview of Each Processing of Peak-Hold-23 Function

Category	Peak-hold Processing	Comparison Processing
Peak-hold-1 processing	<ul style="list-style-type: none"> • The upper/lower peak is detected. • The peak value is held in the peak-hold result register. • When the peak is detected, the notification of Peak-hold updates can be output. 	<ul style="list-style-type: none"> • When the comparison computation result for peak-hold-1 processing is true, the transition to peak-hold-2 processing occurs. • The condition match interrupt is not output.
Peak-hold-2 processing	<ul style="list-style-type: none"> • The upper/lower peak is detected. • The peak value is held in the peak-hold-2 result register 0. • When the peak is detected, the notification of peak-hold updates can be output. 	<ul style="list-style-type: none"> • When the comparison computation result for peak-hold-2 processing is true, the transition to peak-hold-3 processing occurs. • The condition match interrupt is not output.
Peak-hold-3 processing	<ul style="list-style-type: none"> • The upper/lower peak is detected. • The peak value is held in the peak-hold-3 result register 0. • When the peak is detected, the notification of peak-hold updates can be output. 	<ul style="list-style-type: none"> • When the comparison computation result for peak-hold-3 processing is true, the condition match interrupt is output, and peak-hold processing and comparison processing stop.

37.5.12.10 Operation of Peak-Hold-23 Function

With the settings described in **Section 37.5.12.14**, for the target channel, the peak-hold-23 function becomes available in peak-hold processing.

Peak-hold processing is executed by each of peak-hold-1 processes, peak-hold-2 processes, and peak-hold-3 processes using the data input to the peak-hold circuit. When the comparison computation result is true in each peak-hold processing, the transitions to peak-hold-2 and peak-hold-3 processing occur.

The status of the peak-hold-23 function can be checked by the peak-hold-23 function status bits (DFEjSTCHn.PH23ST).

Peak-hold-1 processing starts first for the filter processing where the peak-hold initialization flag has been enabled. When the condition for PH updates is matched during peak-hold-1 processing, the data input to the peak-hold circuit is stored in the peak-hold result register of the channel, and the notification of updates is issued. When the comparison computation result for peak-hold-1 processing is true, the transition to peak-hold-2 processing occurs in the next filtering process.

During peak-hold-2 processing, as with peak-hold-1 processing, the peak value is stored in the peak-hold-2 result register 0, and the notification of updates is issued. When the comparison condition for peak-hold-2 processing is true, the transition to peak-hold-3 processing occurs in the next filtering process.

During peak-hold-3 processing, as with peak-hold-1 processing, the peak value is stored in the peak-hold-3 result register 0, and the notification of updates is issued. When the comparison condition for peak-hold-3 processing is true, the condition match interrupt request is issued, and both peak-hold processing and comparison processing stop in the next filter process.

During the stop period, peak-hold processing and comparison processing are not executed. The peak-hold end interrupt request is issued in the filter process where the peak-hold end flag has been enabled, and the operation of the peak-hold-23 function ends.

When the peak-hold initialization flag is enabled again, peak-hold-1 processing starts in that filter process. Then, this operation is repeated. Operation of the peak-hold-23 function is shown below.

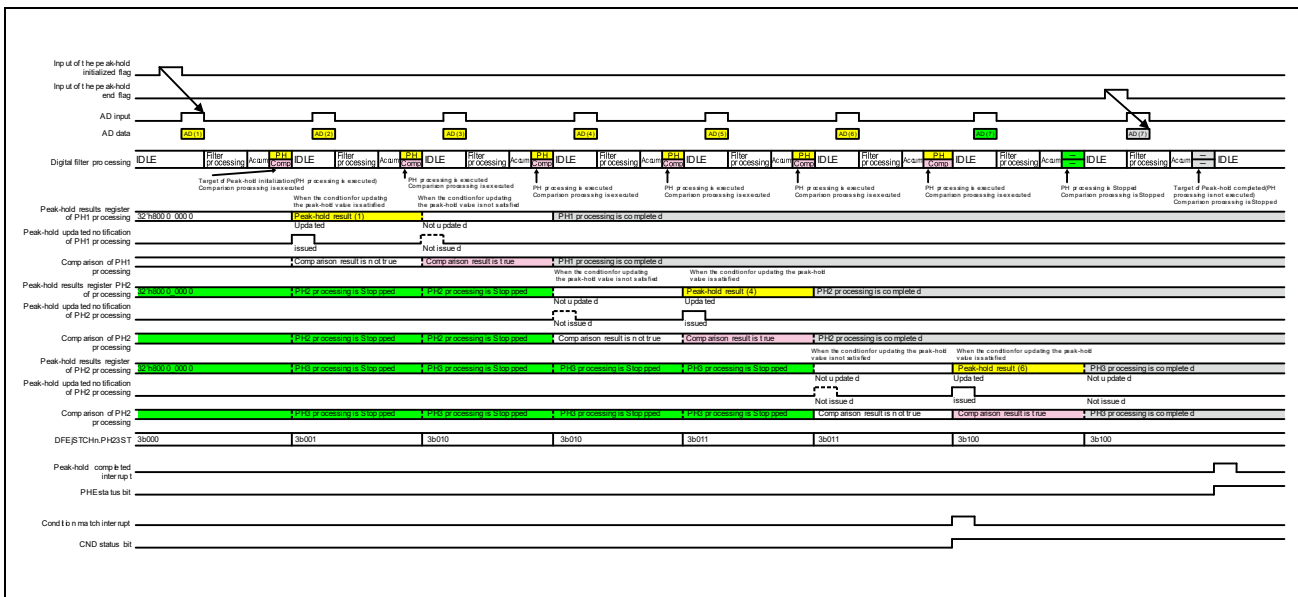


Figure 37.85 Operation of the Peak-Hold-23 Function

(1) Settings of Peak-Hold-23 Function

With the following settings, the peak-hold-23 function can be used for the target channel.

Category	Register Bit Name	Value	Remarks
Setting mode	DFEjPH23CCTL0.PEN	1	Peak-hold-23 Function is enabled.
	DFEjPH23CCTL0.CHS	Channel number	The target channel of peak-hold-23 function is selected.

(2) Settings and Operation of Peak-Hold-1 Processing

(a) Settings

The register settings required for peak-hold-1 processing are shown below.

To make the settings for peak-hold processing, it is necessary to select the upper or lower peak and the initial values of peak-hold processing.

To make the settings for comparison processing, it is necessary to select the comparison computation, the comparison offset value setting register, and the comparison offset value.

Make sure that the data format of the comparison offset value α is the same as the format of the filter processing result. By setting the comparison offset value with a two's complement, it is possible to execute the comparison computation with a comparison target value as "Value of peak-hold result register $n - \alpha$ ". The setting of the comparison offset value α is the same as that for peak-hold-2 processing and peak-hold-3 processing.

Category	Register Bit Name	Value	Remarks
Setting mode	DFEjCTLBCHn.PRC SB	11 _B	Indispensable setting
Peak-hold processing	DFEjCTLBCHn.PHPS	x	Selects the upper or lower peak.
	DFEjCTLBCHn.PHSLB2	xxx _B	Selects the PH initial value register.
Comparison processing	DFEjCTLBCHn.CPCS	1	Indispensable setting
	DFEjCTLBCHn.SELB1	00 _B	Selects the value of PH result register $n + \alpha$.
	DFEjCTLBCHn.SELB2	xxx _B	Selects the comparison computation.
	DFEjCTLBCHn.OFSL	xx _B	Selects the comparison offset value setting register to use.
	DFEjCPOFSTn.CPOFST	xxxx xxx _H	Sets the comparison offset value; two's complement when $-\alpha$ is set.

(b) Operation

In peak-hold processing for the peak-hold-1 process, the notification of peak-hold updates is issued and the data input to the peak-hold circuit is stored in the peak-hold result register. Conditions for peak-hold updates of peak-hold-1 process are as follows.

Peak	Function	Number Times of PH	Issue Condition
Upper peaks	PH update notification	First time* ¹	PH circuit input data* ² > The value of the peak-hold initial value setting register* ³
		After the first time	PH circuit input data* ² > The value of the DFEjPHCHn* ⁴
Lower peaks	PH update notification	First time* ¹	PH circuit input data* ² < The value of the peak-hold initial value setting register* ³
		After the first time	PH circuit input data* ² < The value of the DFEjPHCHn* ⁴

Peak	Function	Number Times of PH	Update Condition	Stored Data in DFEjPHCHn
Upper peaks	Peak-Hold Result register	First time* ¹	PH circuit input data* ² > The value of the peak-hold initial value setting register* ³	PH circuit input data
			PH circuit input data* ² ≤ The value of the peak-hold initial value setting register* ³	The value of the peak-hold initial value setting register* ³
		After the first time	PH circuit input data* ² > The value of the DFEjPHCHn* ⁴	PH circuit input data
			PH circuit input data* ² ≤ The value of the DFEjPHCHn* ⁴	Holds the value of DFEjPHCHn
Lower peaks	Peak-Hold Result register	First time* ¹	PH circuit input data* ² < The value of the peak-hold initial value setting register* ³	PH circuit input data
			PH circuit input data* ² ≥ The value of the peak-hold initial value setting register* ³	The value of the peak-hold initial value setting register* ³
		After the first time	PH circuit input data* ² < The value of the DFEjPHCHn* ⁴	PH circuit input data
			PH circuit input data* ² ≥ The value of the DFEjPHCHn* ⁴	Holds the value of DFEjPHCHn

Note 1. The first time means when filter processing is executed after PH initialization flag is valid.

Note 2. PH input data is the output data of the filter circuit or accumulation/decimation circuit.

Note 3. The register means one of the DFEjPHIA-DFEjPHID registers.

Note 4. These values are compared in either fixed-point format or integer format regardless of the DFEjCTLBCHn.PFMT bit setting.

- Comparison processing of peak-hold-1 processing is used as the condition for the transition to peak-hold-2 processing. In comparison processing, the following comparison computation is performed. In the following comparison processing, a “less-than operation (<)” is selected as an example.

Comparison processing of peak-hold-1	Value of input data to peak-hold circuit < value of peak-hold result register n + α.
--------------------------------------	--

(3) Settings and Operation of Peak-Hold-2 Processing

(a) Settings

The register settings required for peak-hold-2 processing are shown below.

To make the settings for peak-hold processing, it is necessary to select the upper or lower peak and the initial values of peak-hold processing.

To make the settings for comparison processing, it is necessary to select the comparison computation, the comparison offset value setting register, and the comparison offset value.

Category	Register Bit Name	Value	Remarks
Peak-hold processing	DFEjPH2CTL0.PHPS	x	Selects the upper or lower peak.
	DFEjPH2CTL0.PH2SLB2	xxx _B	Selects the PH initial value register.
Comparison processing	DFEjPH2CTL0.CN2SLB2	xxx _B	Selects the comparison computation.
	DFEjPH2CTL0.OFSL	xx _B	Selects the comparison offset value setting register to use.
	DFEjCPOFSTn.CPOFST	xxxx xxx _H	Sets the comparison offset value; two's complement when $-\alpha$ is set.

(b) Operation

In peak-hold processing of peak-hold-2 process, the notification of peak-hold updates is issued and the data input to the peak-hold circuit is stored in the peak-hold-2 result register 0. Conditions for peak-hold updates of peak-hold-2 process are as follows.

Peak	Function	Number Times of PH	Issue Condition
Upper peaks	PH update notification	The first time of transition to PH2 processing	PH circuit input data* ¹ > The value of the peak-hold initial value setting register* ²
		After the first time	PH circuit input data* ¹ > The value of the DFEjPH20* ³
Lower peaks	PH update notification	The first time of transition to PH2 processing	PH circuit input data* ¹ < The value of the peak-hold initial value setting register* ²
		After the first time	PH circuit input data* ¹ < The value of the DFEjPH20* ³

Peak	Function	Number Times of PH	Update Condition	Stored Data in DFEjPH20
Upper peaks	Peak-Hold Result register	The first time of transition to PH2 processing	PH circuit input data* ¹ > The value of the peak-hold initial value setting register* ⁴	PH circuit input data
			PH circuit input data* ¹ ≤ The value of the peak-hold initial value setting register* ⁴	The value of the peak-hold initial value setting register* ⁴
		After the first time	PH circuit input data* ¹ > The value of the DFEjPH20* ³	PH circuit input data
			PH circuit input data* ¹ ≤ The value of the DFEjPH20* ³	Holds the value of DFEjPH20
Lower peaks	Peak-Hold Result register	The first time of transition to PH2 processing	PH circuit input data* ¹ < The value of the peak-hold initial value setting register* ⁴	PH circuit input data
			PH circuit input data* ¹ ≥ The value of the peak-hold initial value setting register* ⁴	The value of the peak-hold initial value setting register* ⁴
		After the first time	PH circuit input data* ¹ < The value of the DFEjPH20* ³	PH circuit input data
			PH circuit input data* ¹ ≥ The value of the DFEjPH20* ³	Holds the value of DFEjPH20

Note 1. PH input data is the output data of the filter circuit or accumulation/decimation circuit.

Note 2. The register means one of the DFEjPHIA-DFEjPHID registers.

Note 3. These values are compared in either fixed-point format or integer format regardless of the DFEjPH23CCTL0.PFMT bit setting.

Note 4. For detail of the peak-hold initial value setting, refer to Note of DFEjPH2CTL0.PH2SLB2 in **Section 37.3.36, DFEjPH2CTL0 — Peak-Hold 2 Control Register 0.**

- Comparison processing of peak-hold-2 process is used as the condition for the transition to peak-hold-3 processing. In comparison processing, the following comparison computation is performed. In the following comparison processing, a “greater-than operation (>)” is selected as an example.

Comparison processing of peak-hold-2	value of input data to peak-hold circuit > value of peak-hold-2 result register 0 + α.
--------------------------------------	--

(4) Settings and Operation of Peak-Hold-3 Processing

(a) Settings

The register settings required for peak-hold-3 processing are shown below.

To make the settings for peak-hold processing, it is necessary to select the upper or lower peak and the initial values of peak-hold processing.

To make the settings for comparison processing, it is necessary to select the comparison computation, the comparison offset value setting register, and the comparison offset value.

Category	Register Bit Name	Value	Remarks
Peak-hold processing	DFEjPH3CTL0.PHPS	x	Selects the upper or lower peak.
	DFEjPH3CTL0.PH3SLB2	xxx _B	Selects the PH initial value register.
Comparison processing	DFEjPH3CTL0.CN3SLB2	xxx _B	Selects the comparison computation.
	DFEjPH3CTL0.OFSL	xx _B	Selects the comparison offset value setting register to use.
	DFEjCPOFSTn.CPOFST	xxxx xxx _H	Sets the comparison offset value; two's complement when -α is set.

(b) Operation

In peak-hold processing of peak-hold-3 process, the notification of peak-hold updates are issued and the data input to the peak-hold circuit is stored in the peak-hold-3 result register 0. Conditions for peak-hold updates of peak-hold-3 process are as follows.

Peak	Function	Number Times of PH	Issue Condition
Upper peaks	PH update notification	The first time of transition to PH3 processing	PH circuit input data* ¹ > The value of the peak-hold initial value setting register* ²
		After the first time	PH circuit input data* ¹ > The value of the DFEjPH30* ³
Lower peaks	PH update notification	The first time of transition to PH3 processing	PH circuit input data* ¹ < The value of the peak-hold initial value setting register* ²
		After the first time	PH circuit input data* ¹ < The value of the DFEjPH30* ³

Peak	Function	Number Times of PH	Update Condition	Stored Data in DFEjPH30
Upper peaks	Peak-Hold Result register	The first time of transition to PH3 processing	PH circuit input data* ¹ > The value of the peak-hold initial value setting register* ⁴	PH circuit input data
			PH circuit input data* ¹ ≤ The value of the peak-hold initial value setting register* ⁴	The value of the peak-hold initial value setting register* ⁴
		After the first time	PH circuit input data* ¹ > The value of the DFEjPH30* ³	PH circuit input data
			PH circuit input data* ¹ ≤ The value of the DFEjPH30* ³	Holds the value of DFEjPH30
Lower peaks	Peak-Hold Result register	The first time of transition to PH3 processing	PH circuit input data* ¹ < The value of the peak-hold initial value setting register* ⁴	PH circuit input data
			PH circuit input data* ¹ ≥ The value of the peak-hold initial value setting register* ⁴	The value of the peak-hold initial value setting register* ⁴
		After the first time	PH circuit input data* ¹ < The value of the DFEjPH30* ³	PH circuit input data
			PH circuit input data* ¹ ≥ The value of the DFEjPH30* ³	Holds the value of DFEjPH30

Note 1. PH input data is the output data of the filter circuit or accumulation/decimation circuit.

Note 2. The register means one of the DFEjPHIA-DFEjPHID registers.

Note 3. These values are compared in either fixed-point format or integer format regardless of the DFEjPH23CTL0.PFMT bit setting.

Note 4. For detail of the peak-hold initial value setting, refer to Note of DFEjPH3CTL0.PH3SLB2 in **Section 37.3.37, DFEjPH3CTL0 — Peak-Hold 3 Control Register 0.**

- Comparison processing of peak-hold-3 process is used as the condition for ending peak-hold-3 processing. In comparison processing, the following comparison computation is performed. In the following comparison processing, a “less-than operation (<)” is selected as an example.

In peak-hold-3 processing, the condition match interrupt request is issued when the comparison computation result is true. In the subsequent filter processing, both peak-hold processing and comparison processing stop.

The peak-hold end interrupt request is issued in the filter process where the peak-hold end flag has been enabled, and the operation of the peak-hold-23 function for that channel ends.

Comparison processing of peak-hold-3	value of input data to peak-hold circuit < value of peak-hold-3 result register 0 + α .
--------------------------------------	--

37.5.12.11 Operation of the Peak-Hold Index Register When Peak-Hold-23 is Enabled

When the peak-hold-23 function is used, the peak-hold count value is stored in each peak-hold index register for each peak-hold process. The operations of the peak-hold-2 index register 0 (DFEjPH2IND0) and the peak-hold-3 index register 0 (DFEjPH3IND0) are the same as those of the peak-hold index register (DFEjPHINDCHn) of the channel targeted for the peak-hold-23 function.

Category	Target Register	Remarks
Peak-hold 1 processing	DFEjPHINDCHn	This register operates only during peak-hold-1 processing. It holds the value and stops when the transition to peak-hold-2 processing occurs.
Peak-hold 2 processing	DFEjPH2IND0	This register operates only during peak-hold-2 processing. It holds the value and stops when the transition to peak-hold-3 processing occurs.
Peak-hold 3 processing	DFEjPH3IND0	This register operates only during peak-hold-3 processing. It holds the value and stops when peak-hold-3 processing ends.

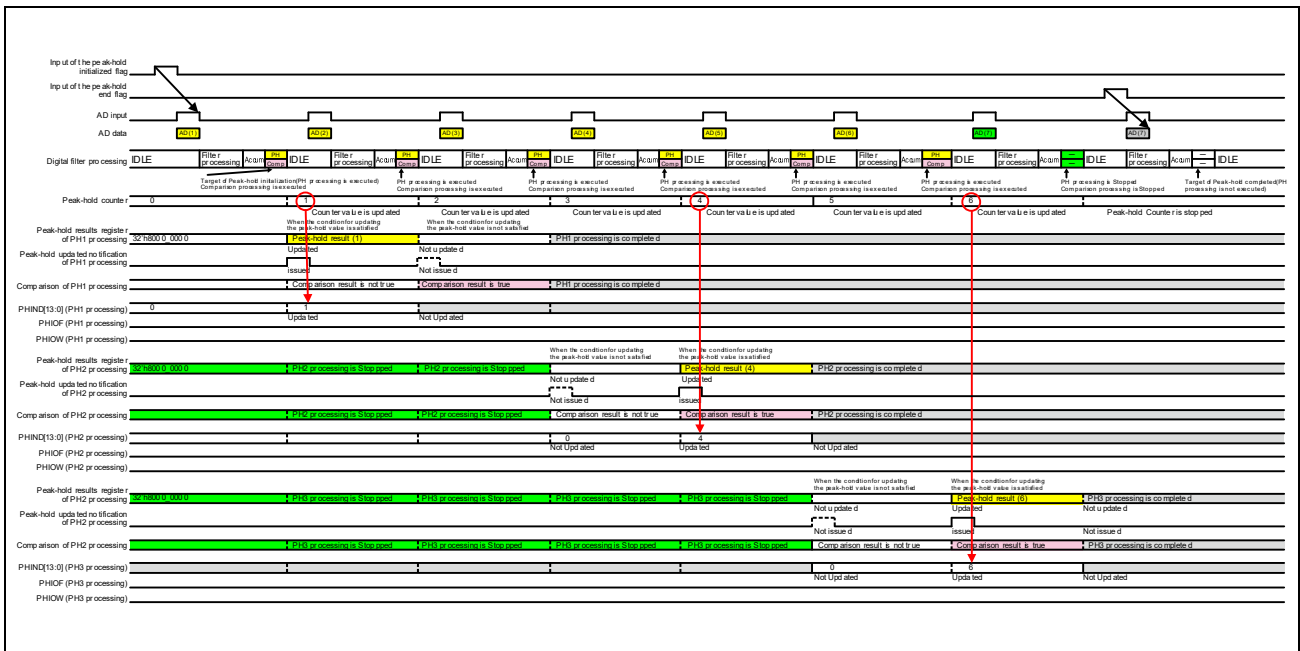


Figure 37.86 Operation of Peak-Hold Index Register when Peak-Hold-23 Function is used

37.5.12.12 Status of Peak-Hold-23 Function

The condition match interrupt request is issued when the comparison computation result is true in peak-hold-3 processing. The peak-hold end interrupt request is issued when the filter processing where a peak-hold end flag that is enabled is executed.

The status of the peak-hold 23 function can be checked by reading the peak-hold 23 function status bit (DFEjSTCHn.PH23ST). For details, see **Section 37.3.8, DFEjSTCHn — Status Register n (n = 0 to 11, for DFE0, n = 0 to 3 for DFE1)**.

When enabling the peak-hold initialization flag again, clear the status bit to 000_B in advance if the status bit indicates neither 000_B nor 100_B.

37.5.12.13 Operation of Peak-Hold-23 Function (Trigger Flag is enable)

The peak-hold-23 function is affected by the function of peak-hold mask because this is an extended function for peak-hold processing. After the peak-hold mask start flag was set valid, the peak-hold-23 processing is kept stopped until the peak-hold mask end flag is set valid.

And the peak-hold-23 function is affected by the auto initialization flag. After the auto initialization flag which is the peak-hold 23 processing target channel was valid, target registers of the auto initialization are initialized (Target registers are DFEjSTCHn.PH23ST, DFEjPHCHn, DFEjPHINDCHn, DFEjPH20, DFEjPH2IND0, DFEjPH30 and DFEjPH3IND0) .

Then, when the peak-hold processing is invalid, the PH23ST bit is set 000_B. After that, when peak-hold processing is valid, the PH23ST bit is set 001_B.

When the auto initialization flag which is the peak-hold 23 processing target channel is valid, and when the peak-hold initialization flag is valid and when the peak-hold mask start flag is valid, target registers of the auto initialization are initialized. The PH23ST bit keeps 000_B until the peak-hold mask is set invalid.

After the peak-hold mask is set invalid, PH23ST bit is set 001_B and the peak-hold 1 processing is started.

37.5.12.14 Setting Example of Peak-Hold-23 Function

The peak-hold-23 function is used to detect the three peak points of the filtering process result for the channel. The following two cases are assumed.

- (1) To Detect the Three Peak Points Starting from the Rising Edge
- (2) To Detect the Three Peak Points Starting from the Falling Edge

The following paragraphs describe how to make the settings for each of the two cases.

(1) To Detect the Three Peak Points Starting from the Rising Edge

This paragraph exemplifies how to set the peak-hold-23 function when you detect the three peak points of filter output values starting from the rising edge.

For the channel targeted for the peak-hold 23 function, make the settings as shown in the example below.

Table 37.85 Setting Example of Peak Detect Starting from the Rising Edge

	Category	Register Bit Name	Value	Remarks
Peak-hold-23 common setting	Mode setting	DFEjPH23CCTL0.PEN	1	
		DFEjPH23CCTL0.CHS	Channel number	Channel targeted for peak-hold-23
	Initial value setting	DFEjPHIA.PHI	8000 0000 _H	Negative minimum
		DFEjPHIB.PHI	7FFF FFFF _H	Positive maximum
Peak-hold-1 processing	Mode setting	DFEjCTLBCHn.PRCSB	11 _B	Indispensable setting
	Peak-hold processing	DFEjCTLBCHn.PHPS	0	Upper peak
		DFEjCTLBCHn.PHSLB2	000 _B	Selects DFEjPHIA.
	Comparison processing	DFEjCTLBCHn.CPCS	1	Indispensable setting
		DFEjCTLBCHn.SELB1	00 _B	Selects the value of peak-hold result register n + α.
		DFEjCTLBCHn.SELB2	001 _B	Less than or equal (≤)
		DFEjCTLBCHn.OFSL	00 _B	Selects DFEjCPOFST0.
DFEjCPOFST0.CPOFST	Value of α1	Two's complement		
Peak-hold-2 processing	Peak-hold processing	DFEjPH2CTL0.PHPS	1	Lower peak
		DFEjPH2CTL0.PH2SLB2	001 _B	Selects DFEjPHIB.
	Comparison processing	DFEjPH2CTL0.CN2SLB2	010 _B	Greater than or equal (≥)
		DFEjPH2CTL0.OFSL	01 _B	Selects DFEjCPOFST1.
DFEjCPOFST1.CPOFST	Value of α2			
Peak-hold-3 processing	Peak-hold processing	DFEjPH3CTL0.PHPS	0	Upper peak
		DFEjPH3CTL0.PH3SLB2	000 _B	DFEjPHIA selected
	Comparison processing	DFEjPH3CTL0.CN3SLB2	001 _B	Less than or equal (≤)
		DFEjPH3CTL0.OFSL	10 _B	Selects DFEjCPOFST2.
DFEjCPOFST2.CPOFST	Value of α3	Two's complement		

(2) To Detect the Three Peak Points Starting from the Falling Edge

This paragraph exemplifies how to set the peak-hold-23 function when you detect the three peak points of filter output values starting from the rising edge.

For the channel targeted for the peak-hold 23 function, make the settings as shown in the example below.

Table 37.86 Setting Example of Peak Detect Starting from the Falling Edge

	Category	Register Bit Name	Value	Remarks
Peak-hold-23 common setting	Mode setting	DFEjPH23CCTL0.PEN	1	
		DFEjPH23CCTL0.CHS	Channel number	Channel targeted for peak-hold-23
	Initial value setting	DFEjPHIA.PHI	7FFF FFFF _H	Positive maximum
		DFEjPHIB.PHI	8000 0000 _H	Negative minimum
Peak-hold-1 processing	Mode setting	DFEjCTLBChn.PRCsB	11 _B	Indispensable setting
	Peak-hold processing	DFEjCTLBChn.PHPS	1	Lower peak
		DFEjCTLBChn.PHSLB2	000 _B	Selects DFEjPHIA.
	Comparison processing	DFEjCTLBChn.CPCS	1	Indispensable setting
		DFEjCTLBChn.SELB1	00 _B	Selects the value of peak-hold result register n + α .
		DFEjCTLBChn.SELB2	010 _B	Greater than or equal (\geq)
		DFEjCTLBChn.OFSL	00 _B	Selects DFEjCPOFST0.
DFEjCPOFST0.CPOFST	Value of α 1			
Peak-hold-2 processing	Peak-hold processing	DFEjPH2CTL0.PHPS	0	Upper peak
		DFEjPH2CTL0.PH2SLB2	001 _B	DFEjPHIB selected
	Comparison processing	DFEjPH2CTL0.CN2SLB2	001 _B	Less than or equal (\leq)
		DFEjPH2CTL0.OFSL	01 _B	Selects DFEjCPOFST1.
		DFEjCPOFST1.CPOFST	Value of α 2	Two's complement
Peak-hold-3 processing	Peak-hold processing	DFEjPH3CTL0.PHPS	1	Lower peak
		DFEjPH3CTL0.PH3SLB2	000 _B	DFEjPHIA selected
	Comparison processing	DFEjPH3CTL0.CN3SLB2	010 _B	Greater than or equal (\geq)
		DFEjPH3CTL0.OFSL	10 _B	Selects DFEjCPOFST2.
		DFEjCPOFST2.CPOFST	Value of α 3	

37.5.13 Output Circuit

37.5.13.1 Floating-Point Conversion Circuit

This circuit executes IEEE754 floating-point conversion of (S.31) format data that is input to the output circuit and (S9.22) format data after accumulation. Setting the PRSC bit of the control register B (DFEjCTLBCHn) to 1 converts data to be stored in the output data register to the floating-point format.

37.5.13.2 Output Register Circuit

The output register circuit has 32-bit registers for each channel. Input calculation result data or floating-point conversion results are stored in the output register (DFEjDOCHn.DO[31:0]) for each channel.

When the results of calculation with DFEjCTLBCHn.PRCSA = 01_B (accumulation) or DFEjCTLBCHn.PRCSA = 10_B (decimation) are written to an output register, the DOEN bit in DFEjSTCHn is set and an interrupt request is generated. When filter processing is executed with DFEjCTLBCHn.PRCSA = 00_B, the filtering results are stored in an output register but the DOEN bit is not set and no interrupt is output.

To execute filter processing without accumulation/decimation and output an interrupt, set the PRCSA bit in DFEjCTLBCHn to 10_B (decimation) and set the decimation count to 0.

The DOEN flag in DFEjSTCHn is automatically cleared to 0 by reading the output register. If another calculation result is written to the output data register with the DOEN flag in DFEjSTCHn set to 1, the value in the output data register is overwritten. At this time, the DOOW bit in DFEjSTCHn is set to 1.

The DOEN bit in DFEjSTCHn is cleared to 0 by reading DFEjDOCHn (output data register) from the peripheral bus or setting the CLRDOOW bit in DFEjCLRSTCHn to 1.

37.5.13.3 Floating Point Conversion Circuits for the Intermediate Value Output Registers L and H

An intermediate value circuit produces 38-bit values, which can be processed by floating-point conversion to conform with the IEEE 754 standard in the case of even number channels. If the value of the HOFS bit in control register B (DFEjCTLBCHn) is 1, the contents of the given intermediate value output register L/H pair are converted from 38- to 32-bit floating-point format. Only even number channels have intermediate value output register L/H.

37.5.13.4 Intermediate Value Output Register Circuits

The intermediate values output register circuits for even number channels include 38-bit registers. They are in service when FIR filtering is selected. Values from the results of calculation to which rounding has not been applied or the results of floating-point conversion are stored in the respective intermediate values output registers L and H.

The values become fixed point if 0 has been written to the HOFS bit in control register B (DFEjCTLBCHn). In this case, for each 38-bit value, the six higher-order bits from 2 to 7 (guard bits) are stored in DFEjHOHCHn, and the highest-order bit (the sign bit) and 31 lower-order bits (the fractional part) are stored in DFEjHOLCHn.

If 1 has been written to the HOFS bit, the values are floating-point converted and the 32-bit values thus produced are stored in the given DFEjHOLCHn register. The value of the corresponding DFEjHOHCHn register is fixed to 0.

The timing for updating of the intermediate value register is the same as that for the corresponding output registers.

The intermediate value output registers become invalid if IIR filtering is selected, and their values are fixed to 0.

37.5.14 Capture Circuit

The capture circuit stores the filter results stored in the output circuit in the capture result register in response to a capture trigger generated by the timer.

Filtering results for a maximum of three channels can be stored from the DFE channel.

The data format for the capture result register conforms to the data format for the output data register for a selected channel.

37.5.14.1 Summary of Capture Circuit

Capture trigger input from the timer is enabled when a capture-object channel is selected using a capture-object channel selection bit (DFEjCAPCTLCHn.CAPCH) and when 1 is written to the capture channel enabling bit (DFEjCAPCTLCHn.CEN).

In response to a capture trigger from the timer, the capture circuit stores a value of the output data register (DFEjDOCHn) in the corresponding capture result register (DFEjCAPDOCHn).

37.5.15 Buffer A Circuit

Buffer A is a 32-bit data buffer which is comprised of 8-stage FIFO × 6 channels.

By response to a FIFO capture trigger from the timer, the circuit stores a maximum of the circuit stores a maximum of six values in the output data register on the selected channel in the FIFO--DFE0 or DFE1--in response to a FIFO capture trigger generated by the timer.

The stored data can be read using the P-bus interface.

The circuit issues the buffer A capture end interrupt request indicating the completion of storing of data in the FIFO. For further details, see **Section 37.5.27.2(1), Buffer A Capture End Interrupt Request**.

If an overflow error occurs in the FIFO, the circuit issues an error interrupt request of buffer A.

37.5.15.1 Summary of Buffer A Circuit

The buffer A circuit is enabled by buffer A enable bit, and capture triggers from the timer are enabled.

The storage of data items in each FIFO channel is enabled by the buffer A channel n enable bits. For each of the FIFO channels, either DFE0 or DFE1 is selected by DFE select bits, and the channel number for the output data register is selected by capture-object channel select bits.

Register Bit Name	Remarks
DFBFACCTL.AEN	Enables or disables the buffer A circuit.
DFBFACCTL.AUNE	Enables or disables the capture result register update flag.
DFBFACTLCHn.ADSL	Selects DFE0 or DFE1.
DFBFACTLCHn.BFACH	Selects a channel.
DFBFACTLCHn.CHEN	Enables or disables the FIFO channel.
DFBFACTLCHn.CLRA	Initializes the buffer A circuit.

The data format for the buffer A result register conforms to the data format for the output data register for a selected channel. By using the buffer A data update flag enable bit, it is possible to assign a buffer A update flag to the LSB of the buffer A result register to indicate that the output data register has been updated. For further details, see **Section 37.3.57, DFBFADOCHn — Buffer A Output Register n (n = 0 to 5)**.

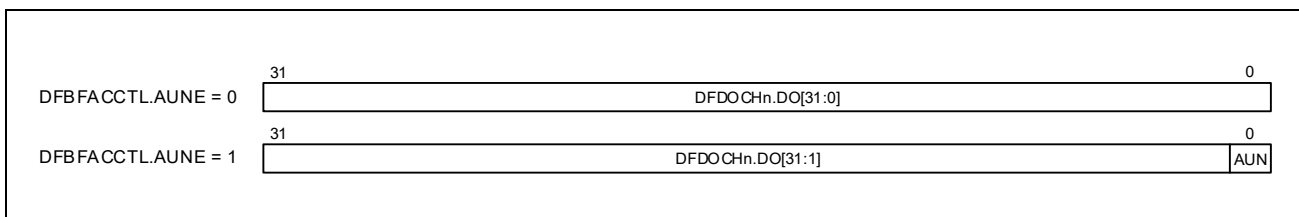


Figure 37.87 Buffer A Result Register Data Format Selection

In response to a FIFO capture trigger from the timer, the macro stores in batch the values of the DFE macro output data register in the enabled FIFO.

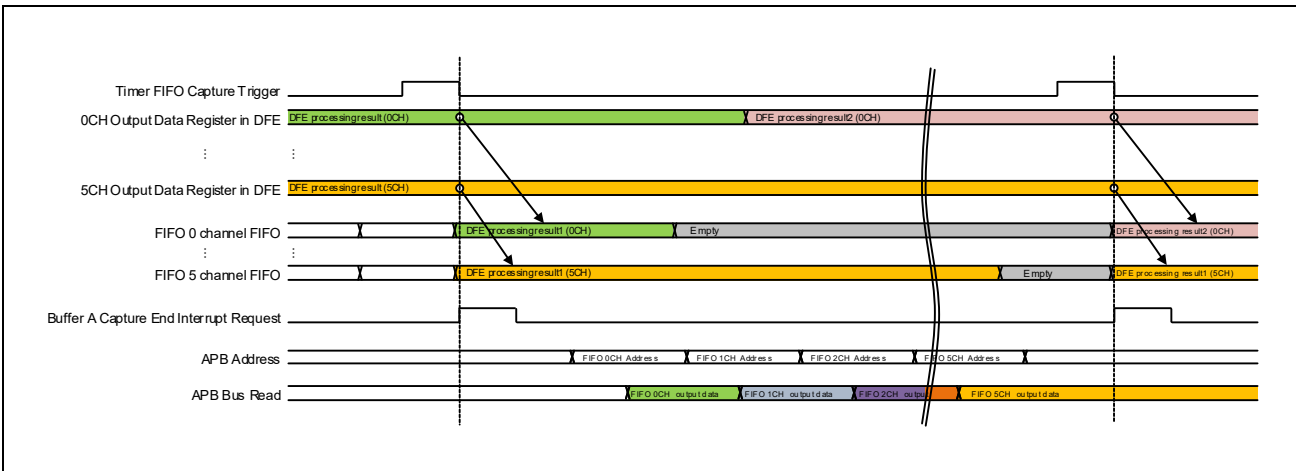


Figure 37.88 Operation of the Buffer A Circuit

37.5.15.2 Operation of Output Data Register Enable Flag and Buffer A Data Update Flag

When the buffer A circuit is selected, the output data register enable flags (DFEjSTCHn.DOEN) on the selected FIFO channels are cleared if data is stored in the FIFO.

When the value of updated output data is stored in the FIFO, the FIFO channel returns the FIFO capture ACK to the DFE macro; the DFE macro clears the DOEN bit on the pertinent channel to 0.

In this case, the buffer A data update flag (DFBFADOCHn.AUN) becomes 0.

Storing the same data, not updated, in the FIFO sets the buffer A data update flag to 1.

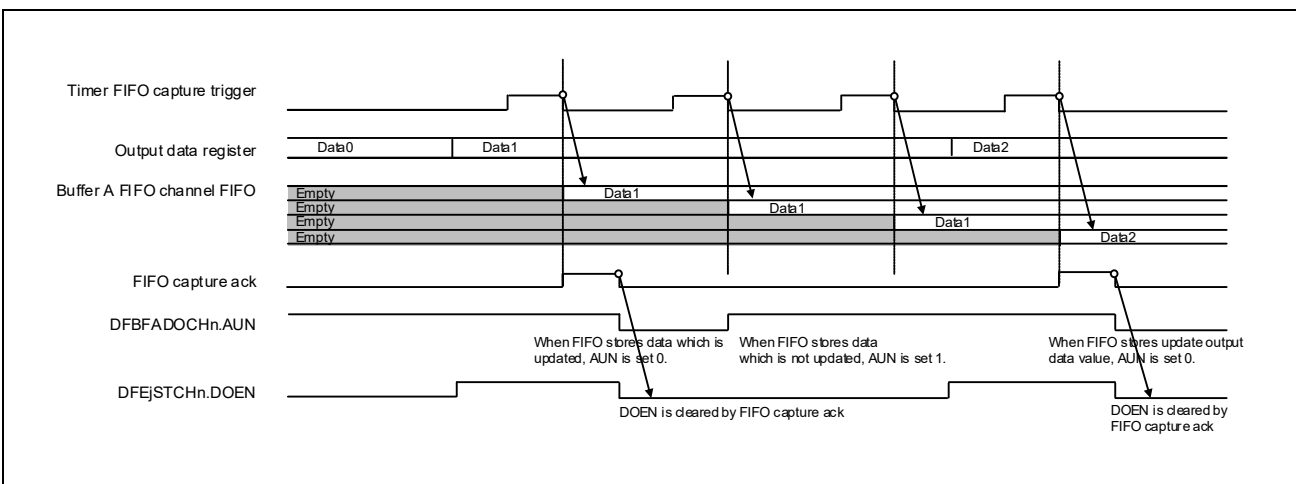


Figure 37.89 Operation of Output Data Register Enable Flag and Buffer A Data Update Flag

37.5.15.3 Initialization of the Buffer A Circuit

By means of the buffer A clear bit (DFBFACLR.CLRA), it is possible to initialize all FIFO channels. In this case, any invalid FIFO channels (DFBFACTLCHn.CHEN = 0) are also initialized.

For details, see **Section 37.3.58, DFBFACLR — Buffer A Clear Register**.

37.5.15.4 Overwriting of FIFO When There is an Overflow

When there is an overflow in a FIFO, no data is overwritten in the FIFO, which rule applies to all FIFOs in the FIFO macro (FIFO channel FIFO and buffer B circuit FIFO). In such a case, any data to store is discarded

Also, when an overflow occurs, the overflow error bit is set to 1. In the waveforms shown below, if an overflow occurs on FIFO channel 0 in the buffer A circuit, the FIFO overflow error bit (DFBFACST.OVFA0) is set to 1.

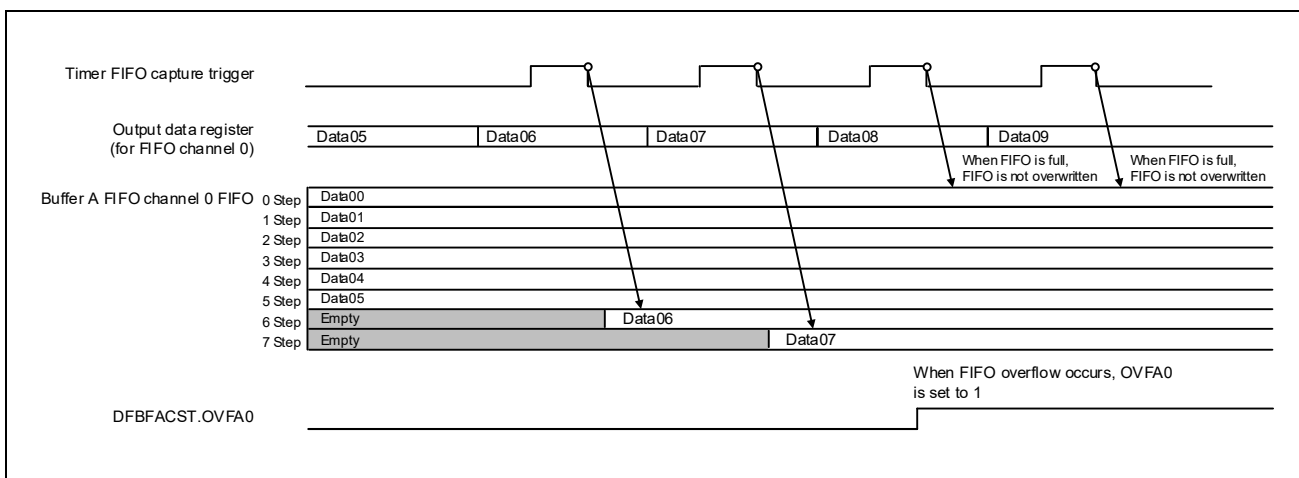


Figure 37.90 Overwrite Operation during FIFO Overflow

37.5.16 Buffer B Circuit

Buffer B is a 32-bit data buffer comprised of an 8-stage FIFO \times 1 channel.

When the value of the output data register on either a selected DFE0 or DFE1 channel is updated, the circuit stores the output data register value on the selected channel in one FIFO.

The stored data can be read by the P-bus interface.

The circuit issues the buffer B capture end interrupt request indicating the completion of storing of data in the FIFO. For further details, see **Section 37.5.27.2(2), Buffer B Capture End Interrupt Request**.

If an overflow error occurs in the buffer B FIFO, the circuit issues a buffer B interrupt request.

37.5.16.1 Summary of Buffer B Circuit

The buffer B circuit is enabled by buffer B enable bit. The circuit stores data in the FIFO when data of the target channel is updated.

Either DFE0 or DFE1 is selected by DFE select bits, and the channel number for the output data register is selected by capture-object channel select bits.

Register Bit Name	Remarks
DFBFBCTL.BEN	Enables or disables the buffer B circuit.
DFBFBCTL.BDSL	Selects DFE0 or DFE1.
DFBFBCTL.BFBCH	Selects a channel.
DFBFBCLR.CLRB	Initializes the buffer B circuit.

The data format for the buffer B result register conforms to the data format for the output data register for a selected channel. The circuit stores the output data register value in the FIFO when the output data of the target channel is updated.

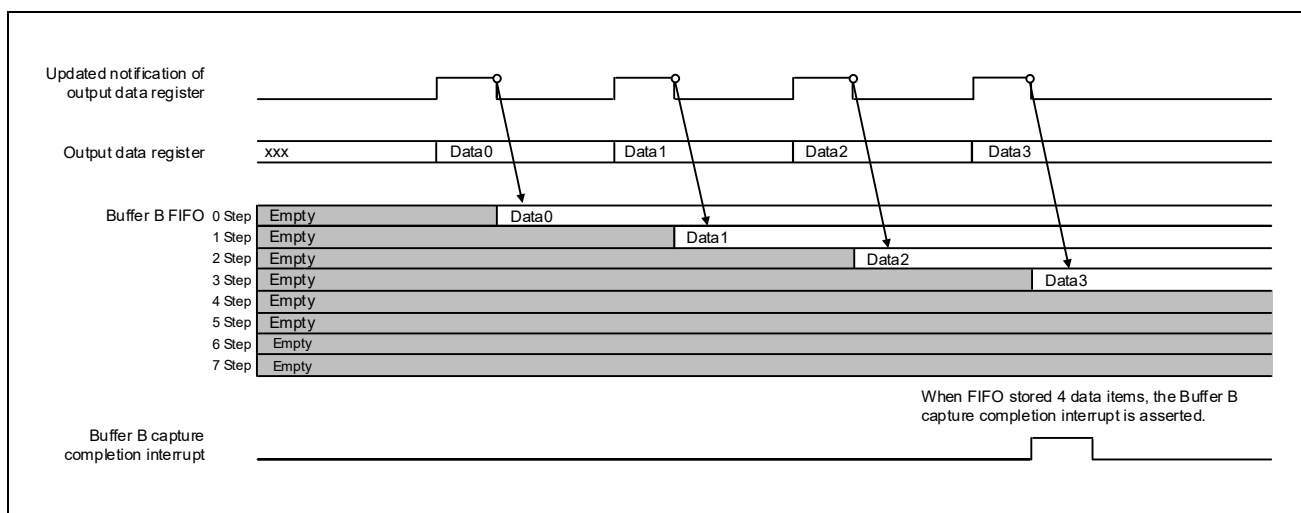


Figure 37.91 Operation of Buffer B Circuit

37.5.16.2 Operation of Output Data Register Enable Flag

When the buffer B circuit is selected, the output data register enable flag (DFEjSTCHn.DOEN) on the selected channel is cleared when data is stored in the buffer B FIFO.

At this time, the buffer B circuit returns the FIFO capture ACK to the DFE macro, and the DFE macro clears the DOEN bit of the target channel to 0.

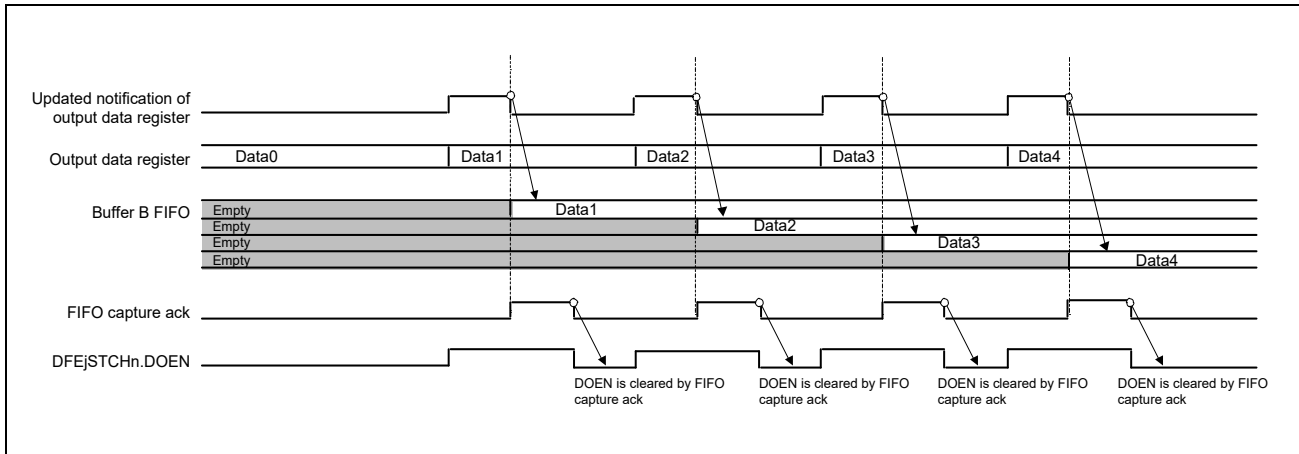


Figure 37.92 Clear Operation of Output Data Register Enabling Flag

37.5.16.3 Initialization for Buffer B Circuit

By means of buffer B clear bit (DFBFBCLR.CLRB), it is possible to initialize the buffer B circuit. For details, see **Section 37.3.62, DFBFBCLR — Buffer B Clear Register**.

37.5.17 Status Register

The DFE has a status register for each channel. The status register indicates the status shown below. Each flag except PH23ST and VALID can be cleared by using the corresponding clear status register. Only the DOEN flag can be automatically cleared by reading the output data register and the intermediate value output register L/H in addition to the clear status register.

Table 37.87 Status Register

Name	Description
PH23ST	<p>Peak-Hold-23 Processing Status</p> <p>When 0 is written to DFEjPH23CCTL0.PEN, PH23ST whose target channel is DFEjPH23CCTL0.CHS is set to 000_B. PH23ST[2:0] is set to 000_B, when the filter processing is executed after the auto initialization flag which is PH23 processing target channel is valid.</p> <p>PH23ST[2:0] is set to 001_B, when the filter processing whose peak-hold-1 initialization flag is valid is executed.</p> <p>PH23ST[2:0] is set to 010_B, when the comparison calculation result of peak-hold-1 is true.</p> <p>PH23ST[2:0] is set to 011_B, when the comparison calculation result of peak-hold-2 is true.</p> <p>PH23ST[2:0] is set to 100_B, when the comparison calculation result of peak-hold-3 is true.</p>
VALID	<p>Input Data Register Valid</p> <p>This bit is set to 1 when the input data register contains a valid value.</p> <p>This bit is cleared to 0 by a reset or at completion of accumulation/decimation processing.</p>
PHE	<p>Peak-Hold End</p> <p>When peak-hold and comparison are used simultaneously (DFEjCTLBCHn.PRCBSB = 11_B), and when the filter processing whose peak-hold end flag is valid is executed, this bit is set to 1.</p> <p>When peak-hold and comparison are not used simultaneously (DFEjCTLBCHn.PRCBSB ≠ 11_B), this bit is disabled (fixed 0).</p>
CND	<p>Condition Match</p> <p>When peak-hold is used (DFEjCTLBCHn.PRCBSB = 01_B), and when the filter processing whose peak-hold end flag is valid is executed, this bit is set to 1.</p> <p>When comparison is used (DFEjCTLBCHn.PRCBSB = 10_B) or peak-hold and comparison are used simultaneously (DFEjCTLBCHn.PRCBSB = 11_B), and when the comparison calculation result is true, This bit is set to 1.</p>
DOEN	<p>Output Data Register Enable</p> <p>Conditions for setting and clearing this flag are described later.</p>
DIOW	<p>Input Data Overwrite Error</p> <p>This bit is set to 1 when the input data register is overwritten during processing for the relevant channel. This bit is also set to 1 when the input data register for the process is overwritten by other input data with the same tag.</p>
DOOW	<p>Output Data Overwrite Error</p> <p>This bit is set to 1 when another calculation result is written while the output data register contains valid data.</p>
MER	<p>Multiplication Error</p> <p>This bit is set to 1 when calculation 8000_H × 8000_H is performed in the filtering circuit.</p>
GER	<p>Guard Error</p> <p>This bit is set to 1 when a rounding error (48 bits to 32 bits) occurs in the filtering circuit.</p>
CER	<p>Cascade Rounding Error</p> <p>This bit is set to 1 when a rounding error (32 bits to 16 bits) occurs during execution of cascade input.</p>
AER	<p>Absolute Value Error</p> <p>This bit is set to 1 when absolute value calculation of 8000 0000_H is performed during absolute value calculation in the accumulation circuit.</p>

37.5.17.1 Conditions for Setting and Clearing the DOEN Flag

The DOEN flag indicates whether values stored in output registers are valid. The flag is valid when settings are for accumulation or decimation. Its value stays unchanged as 0 for any other settings.

When even number channel is in use as an IIR filter, the output data register and intermediate value output registers (including the intermediate value output mirror register L) are all in the scope of operations.

When even number channel is in use as an IIR filter, although the values of the intermediate value output registers are fixed to 0, reading a register still leads to clearing of the DOEN flag. Accordingly, since only the output register is in scope for an IIR filter, reading of the intermediate value output registers (including the intermediate values output mirror register L) is prohibited.

Odd number channels do not have intermediate value output registers, so only the output data registers are in the scope of operations, regardless of the selected type of filter.

Conditions for setting and clearing of the DOEN flag differ with the channel, selected filter type, and selection or non-selection of floating point conversion for the intermediate values output registers. The conditions are summarized in the tables below.

Conditions for setting of and the timing of storage in the DFEjDOCHn, DFEjHOLCHn (or DFEjHOLMCHn), and DFEjHOHCHn bits do not vary.

In some case there are multiple conditions for clearing. In this case, the overall condition is the logical OR of the multiple conditions.

Table 37.88 Conditions for Setting and Clearing DOEN

Case 1: Even number channel with IIR filtering selected on an Odd number channel	
DOEN Setting Condition	DOEN Clearing Conditions
The result of filter calculations are stored in DFEjDOCHn	1 is written to DFEjCLRSTCHn.CLRDOEN
	DFEjDOCHn is read
	The data of target channels being stored in FIFO by Buffer A Circuit or Buffer B Circuit
Case 2: Even number channel with FIR filtering selected, and floating-point conversion of intermediate value is not selected	
DOEN Setting Condition	DOEN Clearing Conditions
The result of filter calculations are stored in DFEjDOCHn, DFEjHOLCHn, and DFEjHOHCHn	1 is written to DFEjCLRSTCHn.CLRDOEN
	DFEjDOCHn is read
	Both DFEjHOLCHn (or DFEjHOLMCHn) and DFEjHOHCHn is read (the order of reading does not create a problem)
	The data of target channels being stored in FIFO by Buffer A Circuit or Buffer B Circuit
Case 3: Even number channel with FIR filtering selected, and floating-point conversion of intermediate value is selected	
DOEN Setting Condition	DOEN Clearing Conditions
The result of filter calculations are stored in DFEjDOCHn and DFEjHOLCHn	1 is written to DFEjCLRSTCHn.CLRDOEN
	DFEjDOCHn is read
	DFEjHOLCHn (or DFEjHOLMCHn) is read
	The data of target channels being stored in FIFO by Buffer A Circuit or Buffer B Circuit

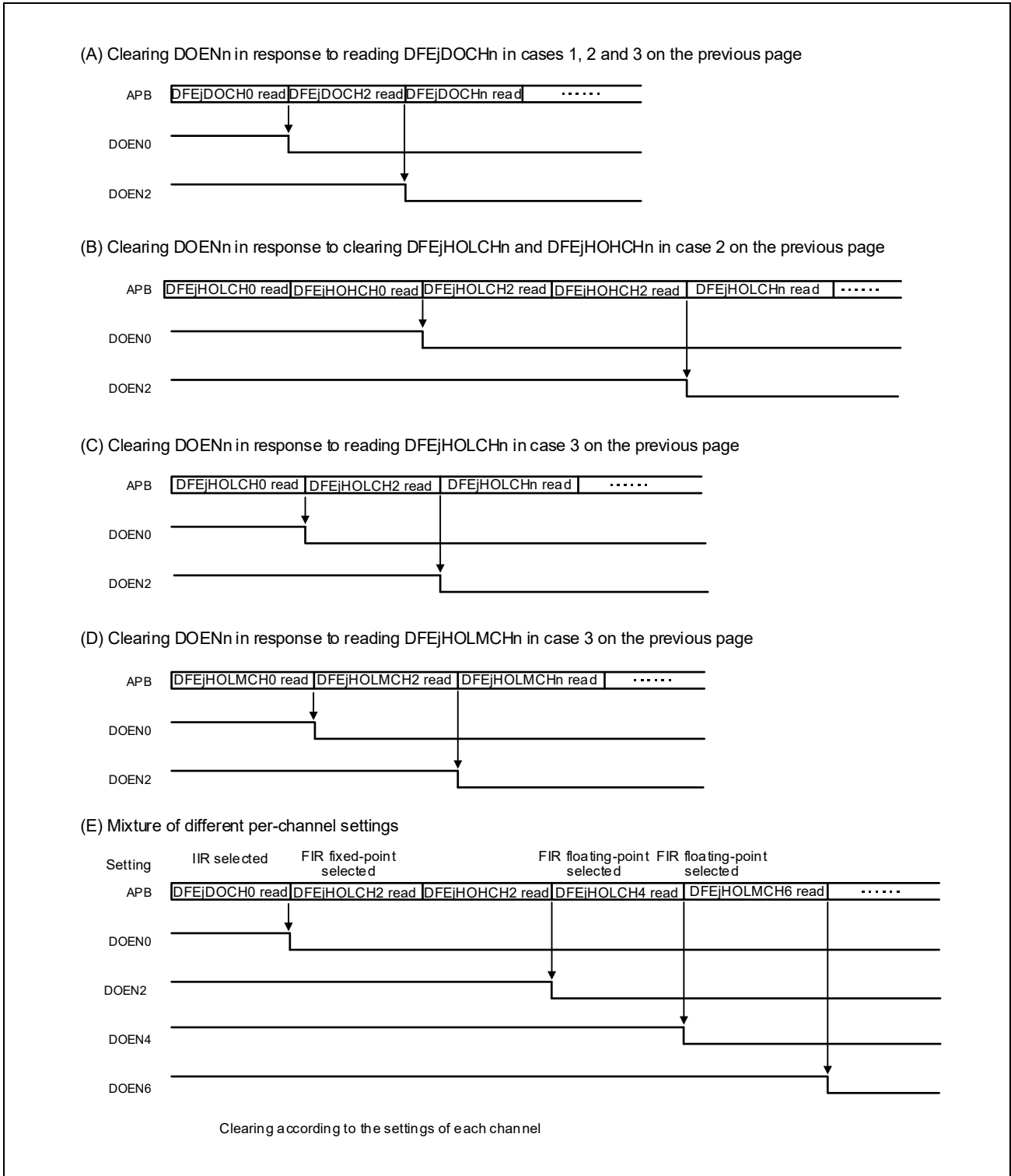


Figure 37.93 Operations for Clearing the DOEN Flag

37.5.18 Error Mask Registers

Functions of error mask registers are shown below. Masked error factors are excluded from error interrupt request conditions.

When the multiplication error mask flag is set to 1 in channel 0, for example, the multiplication error flag is not set to 1 even if a multiplication error occurs in channel 0. No error interrupt request is output in this case.

Table 37.89 Error Mask Registers

Flag	Description
MSKDIOW	Input data error (DIOW) mask bit
MSKDOOW	Output data error (DOOW) mask bit
MSKMER	Multiplication error (MER) mask bit
MSKGER	Guard error (GER) mask bit
MSKAER	Absolute value error (AER) mask bit
MSKCER	Cascade rounding error (CER) mask bit

37.5.19 Subtraction Status Register

For each subtraction channel, there is a subtraction status register. The subtraction status register indicates the status information described below. With the exception of the SUBF and MINF flags, flags are cleared through the use of the corresponding clear status register. In cases exclusively for the SDOEN flag, the flag is cleared automatically when the subtraction result register is read, in addition to the operation of the clear status register.

Table 37.90 Subtraction Status Register

Name	Description
SUBF	Subtrahend Channel Data Input Enable When MINF = 0, completion of the filtering of the subtrahend channel sets SUBF to 1. When SUBF = 1 and MINF = 0, performs subtraction upon completion of the filtering of the minuend channel. Storing subtraction result data in the subtraction result register clears SUBF to 0.
MINF	Minuend Channel Data Input Enable When SUBF = 0, completion of the filtering of the minuend channel sets MINF to 1. When MINF = 1 and SUBF = 0, MINF performs subtraction upon completion of the filtering of the subtrahend channel. Storing subtraction result data in the subtraction result register clears MINF to 0.
SDOEN	Subtraction Output Data Register Valid This bit is set to 1 when the DFE stores the subtraction results in the subtraction output data register. While SDOEN = 1, reading the DFEjSUBDOCHn register clears this bit to 0.
SDOOW	Subtraction Result Register Overwrite Error When SDOEN = 1, storing subtraction output data in the subtraction result register sets SDOOW to 1.
SCER	Subtraction Circuit Cascade Rounding Error When a cascade rounding error occurs in subtraction output data, this bit is set to 1.
SGER	Subtraction Guard Error When an overflow error of subtraction result, this bit is set to 1.

37.5.20 Subtraction Error Mask Registers

The following shows the functions of the subtraction error mask register. Any masked error events are excluded from the error interrupt request conditions.

For example, if the subtraction circuit cascade rounding error mask flag is set on subtraction channel 0, no subtraction circuit cascade rounding error will be set even when a subtraction circuit cascade rounding error condition arises on subtraction channel 0. Also, no error interrupt requests will be issued.

Table 37.91 Subtraction Error Mask Register

Name	Description
MSKSDOOW	Subtraction Result Register Overwrite Error (SDOOW) Mask
MSKSCER	Subtraction Circuit Cascade Rounding Error (SCER) Mask
MSKSGER	Subtraction Guard Error (SGER) Mask

37.5.21 Capture Status Register

For each capture channel, there is a capture status register. The capture status register indicates the status information described below. Flags are cleared through the use of the corresponding capture clear status register.

Table 37.92 Capture Status Register

Name	Description
CDOEN	Capture Output Data Register Valid This bit is set to 1 when the DFE stores the capture results in the capture output data register. While CDOEN = 1, reading the DFEjCAPDOCHn register clears this bit to 0.
CDOOW	Capture Result Register Overwrite Error When CDOEN = 1, storing capture data in the capture result register sets CDOOW to 1.

Note: Even if a capture process is executed, the DOEN for the capture-target of output data register is not cleared.

37.5.22 Capture Error Mask Register

The following shows the functions of the capture error mask register. Any masked error events are excluded from the error interrupt request conditions.

For example, if the capture result register overwrite error mask flag is set on capture channel 0, no capture result register overwrite error will be set even when a capture result register overwrite error condition arises on capture channel 0. Also, no error interrupt requests will be issued.

Table 37.93 Capture Error Mask Register

Name	Description
MSKCDOOW	Capture Result Register Overwrite Error (CDOOW) Mask

37.5.23 Buffer A Status Register

The buffer A status register is provided. The status register indicates the status information described below.

Table 37.94 Buffer A Status Register

Name	Description
EMPA _n	FIFO channel-n empty bit Indicates that no data is stored in the FIFO. The bit is cleared to 0 when data is stored in the FIFO. The bit is set to 1 when all remaining data in the FIFO is read. The bit is set to 1 when 1 is written in DFBFACLR.CLRA. EMPA _n of FIFO channel which is never used indicates 1.
OVFA _n	FIFO channel n FIFO overflow error bit Indicates that a FIFO overflow has occurred. The bit is set to 1 when new data is stored when the FIFO is full. The bit is cleared to 0 when 1 is written in DFBFACLR.CLRA. OVFA _n of FIFO channel which is never used indicates 0.

37.5.24 Buffer A Clear Register

Writing 1 to the buffer A clear bit of the buffer A clear register clears the status bits for all FIFO channels and any internal status (FIFO pointer values for buffer A), and resets them to their initial values.

37.5.25 Buffer B Status Register

The buffer B status register is provided. The status register indicates the status information described below.

Table 37.95 Buffer B Status Register

Name	Description
EMPB	Empty bit Indicates that not data is stored in the FIFO. The bit is cleared to 0 when data is stored in the FIFO. The bit is set to 1 when all remaining data in the FIFO is read. The bit is set to 1 when 1 is written in DFBFBCLR.CLRB. The bit indicates 1 when Buffer B is never used.
OVFB	FIFO overflow error bit Indicates that a FIFO overflow has occurred. The bit is set to 1 when new data is stored when the FIFO is full. The bit is cleared to 0 when 1 is written in DFBFBCLR.CLRB. The bit indicates 0 when Buffer B is never used.

37.5.26 Buffer B Clear Register

Writing 1 to the buffer B clear bit of the buffer B clear register clears the status bits and any internal status (FIFO pointer values for buffer B), and resets them to their initial values.

37.5.27 Interrupt Requests

37.5.27.1 DFE Macro

DFE macro has the interrupt request of the following list.

Table 37.96 List of Interrupt Request for the DFE Macro

Interrupt Request Function	Number	Remarks
Output Data Interrupt Request	DFE0: 12 DFE1: 4	Can be issued to each channel
Condition Match Interrupt Request 0	DFE0: 12 DFE1: 4	Can be issued to each channel
Condition Match Interrupt Request 1	DFE0: 4 DFE1: 4	Can be issued to CH0-CH3 channel
Subtraction Data Interrupt Request	DFE0: 3 DFE1: 3	Can be issued to each subtraction channel
Error Interrupt Request	DFE0: 1 DFE1: 1	Can be issued for all error sources commonly.

(1) Output Data Interrupt Requests

When a calculation result with $DFEjCTLBCHn.PRCSA = 01_B$ (accumulation) or $DFEjCTLBCHn.PRCSA = 10_B$ (decimation) is written to an output register, an output data interrupt request is output. (When the PRCSA is 00_B , a calculation result is stored in an output data register but no interrupt is output.)

Interrupt requests corresponding to each channel can be output.

Interrupt request output can be controlled by using the IEO bit in the control register A.

The relationship between the $DFEjSTCHn.DOEN$ bit and an output data interrupt request in addition to the relationship between the $DFEjSTCHn.DOEN$ flag and an output data interrupt request are shown below. For operation of the $DFEjSTCHn.DOEN$ bit, see **Section 37.5.17.1, Conditions for Setting and Clearing the DOEN Flag**.

If the DOEN flag has been cleared to 0, an output data interrupt is generated when the result of calculation is stored in the output data register again.

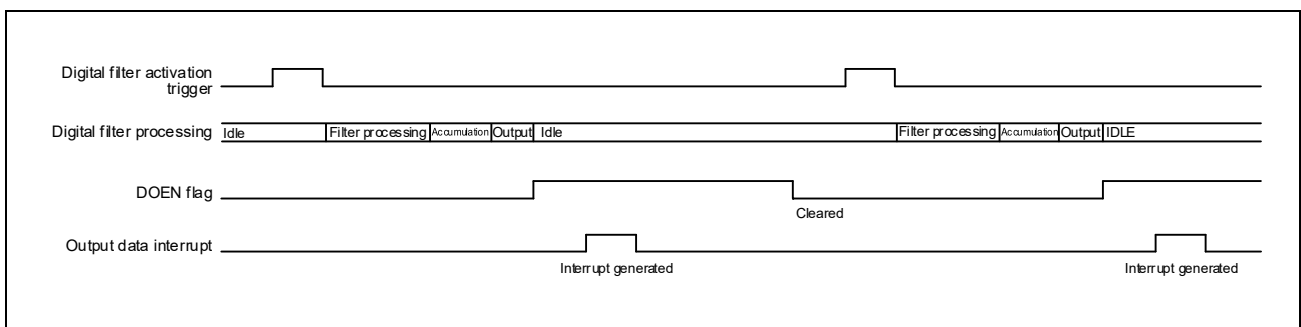


Figure 37.94 Relationship between the DOEN Bit and an Interrupt

When the DOEN bit in $DFEjSTCHn$ is not cleared by the software, an output data interrupt request is not output even if data is rewritten to the output data register. If the output data register value is overwritten, the DOOW bit in $DFEjSTCHn$ is set to 1.

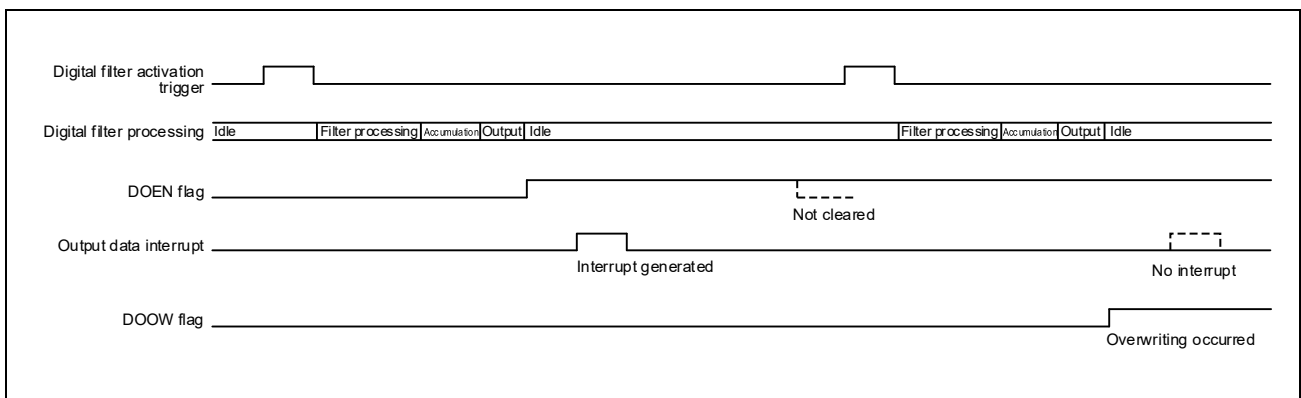


Figure 37.95 Relationship between the DOEN and DOOW Bits and an Interrupt

(2) Condition Match Interrupt Request

When the peak-hold end trigger flag is enabled during peak-hold processing, a condition match interrupt request indicating completion of peak-hold processing is output. When comparison calculation result is true during comparison calculation, a condition match interrupt request is output.

Interrupt request output can be controlled by using the IEC flag in the control register A.

There are two types of condition match interrupt requests: INTDFEjCND0 and INTDFEjCND01. The interrupt source (either DFEjSTCHn.CND or DFEjSTCHn.PHE) can be selected individually for each of these requests. As shown below, interrupt sources that can be assigned vary based on the peak-hold circuit settings (combination of DFEjCTLBCHn.PRCSB and DFEjPH23CCTL0.PEN). INTDFEjCND0 is given to all channels, and INTDFEjCND1 is given to channel 0 to 3. See **Table 37.97**.

Table 37.97 Correspondence between Condition Match Interrupt Requests and Interrupt Sources (1)

Peak-Hold Circuit Settings	PRCSB	PEN	INTDFEjCND0	INTDFEjCND1
Neither peak-hold processing nor comparison calculation processing is executed.	00 _B	—	No function	No function
Peak-hold processing is executed	01 _B	—	CND (peak-hold ended)	No function
Comparison processing is executed.	10 _B	—	CND (condition matched)	No function
Peak-Hold and comparison simultaneously	11 _B	0	*See Table 37.98 .	
Peak-hold-23 is executed.	11 _B	1		

When simultaneous execution of peak-hold processing and comparison processing is selected, or when the peak-hold-23 function is selected, by setting the DFEjCTLACHn.CNSLE and the DFEjCTLACHn.CNSL as shown below, the interrupt source (either DFEjSTCHn.CND or DFEjSTCHn.PHE) to assign to the INTDFEjCND0/1 can be changed.

Table 37.98 Correspondence between Condition Match Interrupt Requests and Interrupt Sources (2)

Peak-Hold Circuit Settings	CNSLE	CNSL	INTDFEjCND0	INTDFEjCND1
Peak-hold and comparison simultaneously	0	—	CND (condition matched)	PHE (peak-hold ended)
	1	0	CND (condition matched)	PHE (peak-hold ended)
		1	PHE (peak-hold ended)	CND (condition matched)
Peak-hold-23 is executed.	0	—	CND (condition matched)	PHE (peak-hold ended)
	1	0	CND (condition matched)	PHE (peak-hold ended)
		1	PHE (peak-hold ended)	CND (condition matched)

Please note that no condition match interrupt request is output after a condition match interrupt request is issued, unless the pertinent interrupt source status is cleared (see **Figure 37.96, item (2)**).

The condition match bit (DFEjSTCHn.CND) is cleared when 1 is written to DFEjCLRSTCHn.CLRCND. Also, peak-hold end bit (DFEjSETCHn.PHE) is cleared when 1 is written to DFEjCLRSTCHn.CLRPHE.

For the condition match interrupt request based on a DFEjSTCHn.CND source, output of an interrupt request can be controlled by means of DFEjCTLACHn.IEC. Also, for the condition match interrupt request based on DFEjSTCHn.PHE, output of an interrupt request can be controlled by means of DFEjCTLACHn.IEP.

Figure 37.96 shows the relationship between the CND bit in the status register (DFEjSTCHn (n = 0 to 11 for DFE0, n = 0 to 3 for DFE1)) and condition match interrupt request. When the peak-hold end flag is enabled, a condition match interrupt request is issued at the same time.

The DISB bit (peak-hold processing disable bit) in DFEjCTLBCHn disables peak-hold processing in the same way as the peak-hold end flag is enabled, but does not output a condition match interrupt. In addition, even if a peak-hold end trigger is input with peak-hold processing, no condition match interrupt is output.

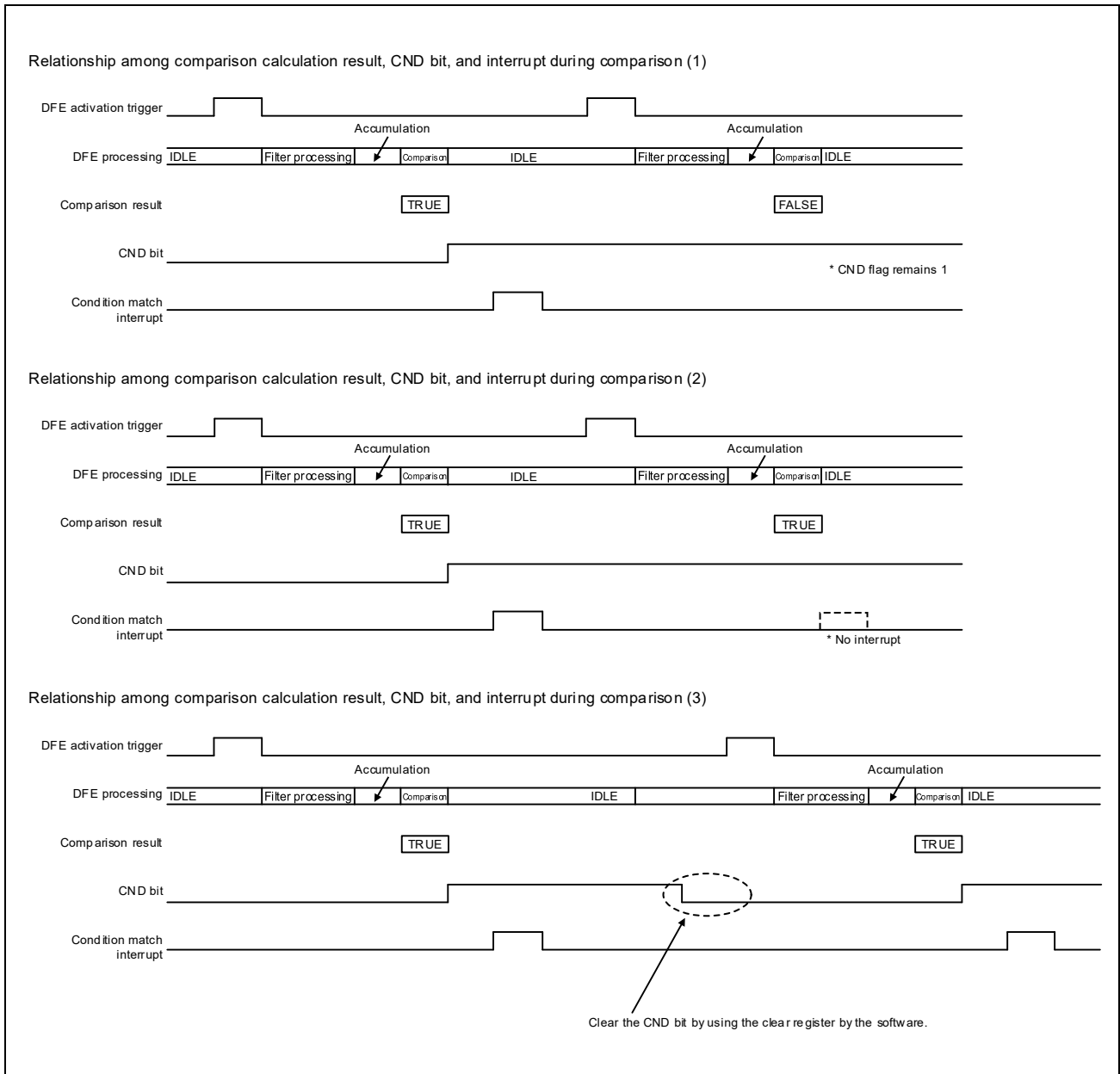


Figure 37.96 Relationship between CND Bit and Condition Match Interrupt Request

(3) Subtraction Data Interrupt Request

Upon storing subtraction results in the subtraction result register, the subtraction circuit issues the subtraction data interrupt request.

The interrupt requests can be issued for corresponding subtraction channels, and by using the DFEjSUBCTLCHn.SIEO bits, each interrupt request output can be controlled.

The figure below shows the relationship between the DFEjSUBSTCHn.SDOEN flags and the subtraction data interrupt requests.

Clearing the SDOEN flag to 0 and storing the subtraction results again in the subtraction result register issues a subtraction data interrupt request.

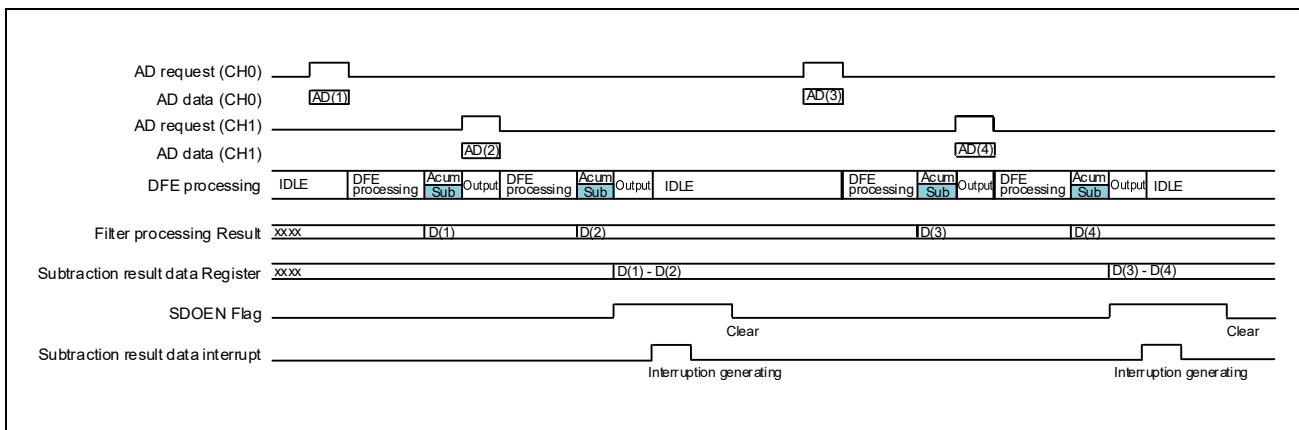


Figure 37.97 Relationship between SDOEN and Subtraction Data Interrupt Request

If the DFEjSUBSTCHn.SDOEN flag is not cleared to 0, no subtraction data interrupt request is issued even when the subtraction results are stored again in the subtraction result register. In this case, because the subtraction result register is overwritten, the DFEjSUBSTCHn.SDOOW flag is set to 1.

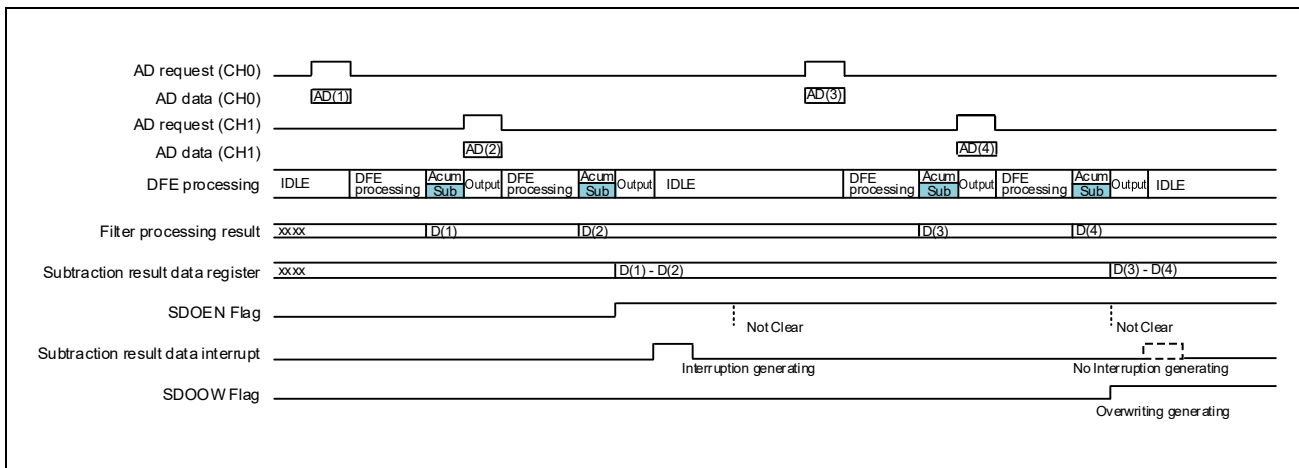


Figure 37.98 Relationship Between SDOEN, SDOOW and Subtraction Data Interrupt Request

(4) Capture Data Overwrite Error

Having stored the value of the output data register in the capture result register, the capture circuit sets DFEjCAPSTCHn.CDOEN to 1; it does not issue an interrupt request that indicates the storage of data in the capture result register.

If the DFEjCAPSTCHn.CDOEN flag is not cleared to 0, storing capture data again in the capture result register causes the overwriting of the capture result register; consequently, the DFEjCAPSTCHn.CDOOW flag is set to 1, enabling an error interrupt to be issued.

For further details on error interrupts, see **Section 37.5.27.1(5)**.

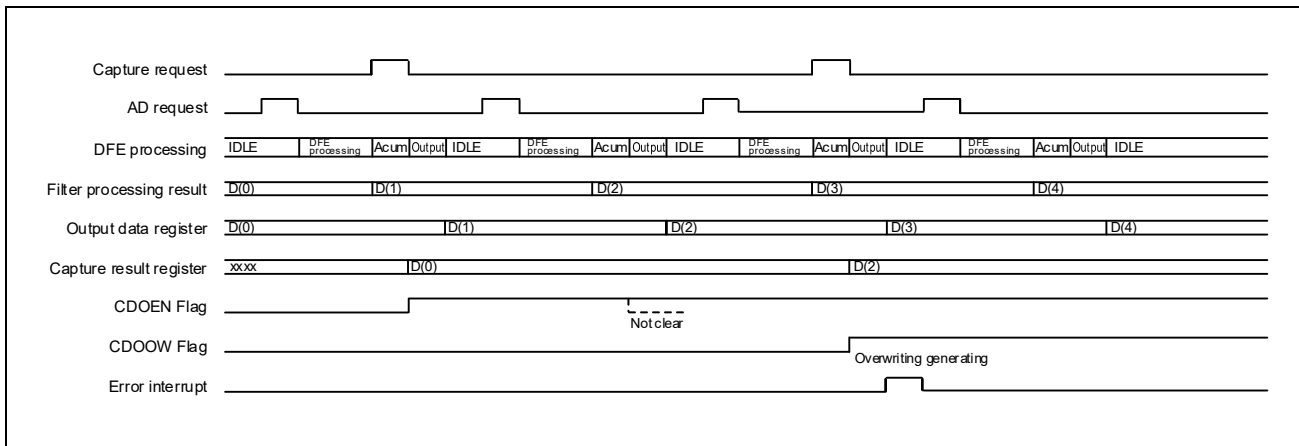


Figure 37.99 Relationship Between CDOEN, CDOOW and Error Interrupt Request

(5) Error Interrupt Requests

Error interrupt request is composed of the 'OR' for the error sources on all channels, all subtraction channels, and all capture channels. All the DFE macro error sources can be summarized as follows.

Error interrupt request output can be controlled by channel, by subtraction channel, and by capture channel. For the meanings of error sources, see **Section 37.5.17** to **Section 37.5.22**.

Table 37.99 Summary of DFE Macro Error Factors

Category	Error Status Bit	Remarks
Channel	DFEjSTCHn.DIOW	Input Data Overwrite Error
	DFEjSTCHn.DOOW	Output Data Overwrite Error
	DFEjSTCHn.MER	Multiplication Error
	DFEjSTCHn.GER	Guard Error
	DFEjSTCHn.CER	Cascade Rounding Error
	DFEjSTCHn.AER	Absolute Value Error
Subtraction Channel	DFEjSUBSTCHn.SDOOW	Subtraction Result Register Overwrite Error
	DFEjSUBSTCHn.SCER	Subtraction Circuit Cascade Rounding Error
	DFEjSUBSTCHn.SGER	Subtraction Guard Error
Capture Channel	DFEjCAPSTCHn.CDOOW	Capture Result Register Overwrite Error

For example, in the case of error interrupt requests for channel 0 to channel 7 are enabled and error interrupt requests for channel 8 to channel 11 are masked, if an error occurs in any channel from 0 to 7, an error interrupt request is output. However, if an error occurs in any channel from 8 to 11 in this state, no error interrupt request is output.

Since error interrupts are composed of the OR of error sources, if an error interrupt request has already been issued, no error interrupts are issued when a new error source occurs. After an error interrupt request is issued, if an error bit occurs again after the error bit is cleared using the clear register, an error interrupt request is issued.

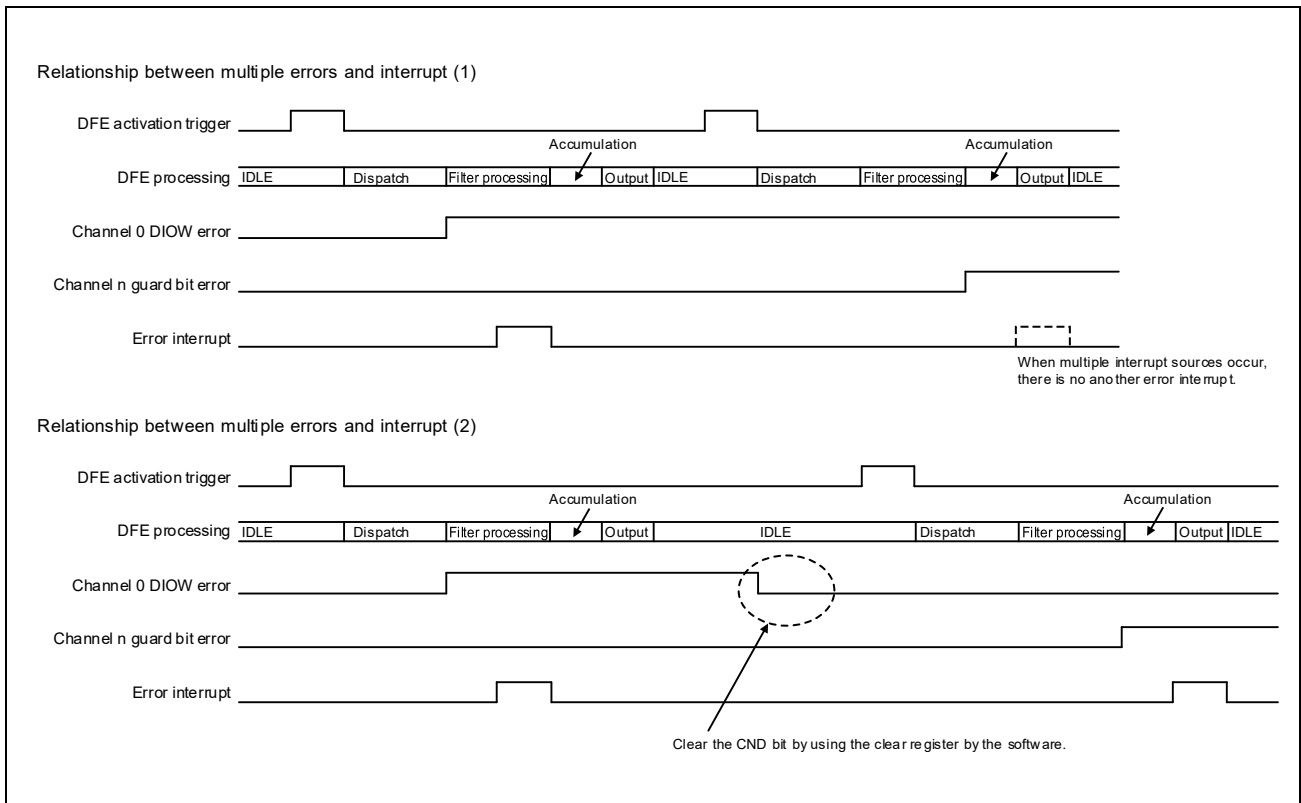


Figure 37.100 Relationship between Error Bit and Error Interrupt

37.5.27.2 FIFO Macro

FIFO macro has the interrupt request in the following list.

Table 37.100 List of Interrupt Request of FIFO Macro

Interrupt Request Function	Number	Remarks
Buffer A capture end interrupt request	1	Can be issued to all FIFO channels commonly
Buffer B capture end interrupt request	1	Can be issued to buffer B simple substance
FIFO Error Interrupt Request	1	Can be issued to all error sources commonly

(1) Buffer A Capture End Interrupt Request

The output of the buffer A capture end interrupt request can be controlled by setting the AIEO bit of the buffer A common control register (DFBFACCTL) to 1.

When the FIFO data, which is the object of reading in the buffer A output data register (DFBFADOCHn) is updated, the buffer A capture end interrupt request is issued.

If data is stored in the FIFO in the FIFO empty state, for example, the buffer A capture end interrupt request is issued. Also, the buffer A capture end interrupt request is issued when all buffer A output data registers are read on the enabled FIFO channels when multiple data items are stored on each of the FIFO channels.

To issue a buffer A capture end interrupt request again, read all data from the enabled FIFO channels. No buffer A capture end interrupt request will be issued if all of the data is not read on the enabled FIFO channels.

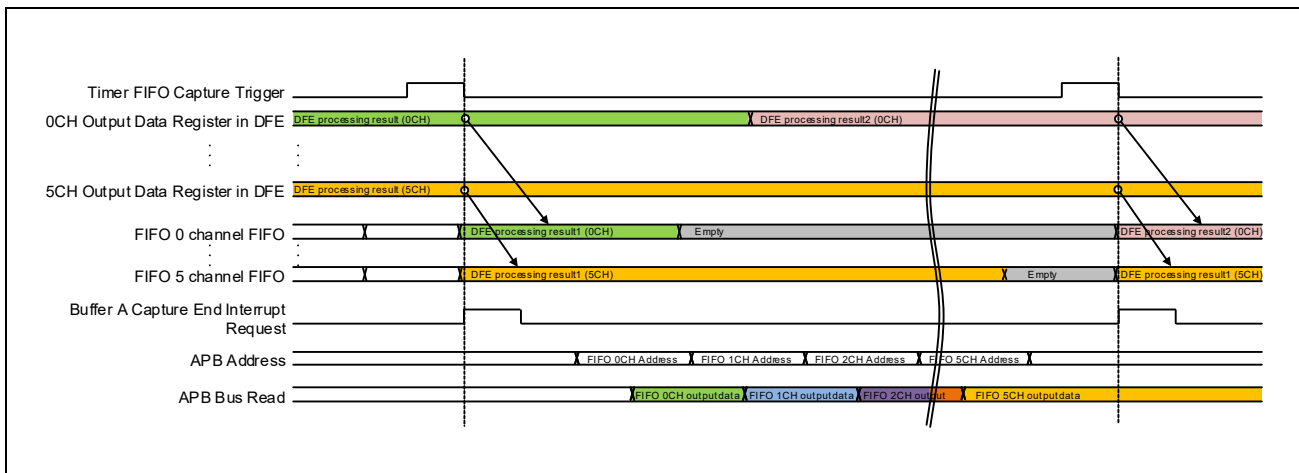


Figure 37.101 FIFO Macro Buffer A Capture End Interrupt Requests

(2) Buffer B Capture End Interrupt Request

The output of the buffer B capture end interrupt request can be controlled by setting the BIEO bit of the buffer B control register (DFBFBCTL) to 1.

When data is stored in the buffer B four times, the buffer B capture end interrupt request is issued. To issue a buffer B capture end interrupt request again, read all the four data items. Otherwise, no buffer B capture end request will be issued even if four data items are stored in the buffer B again.

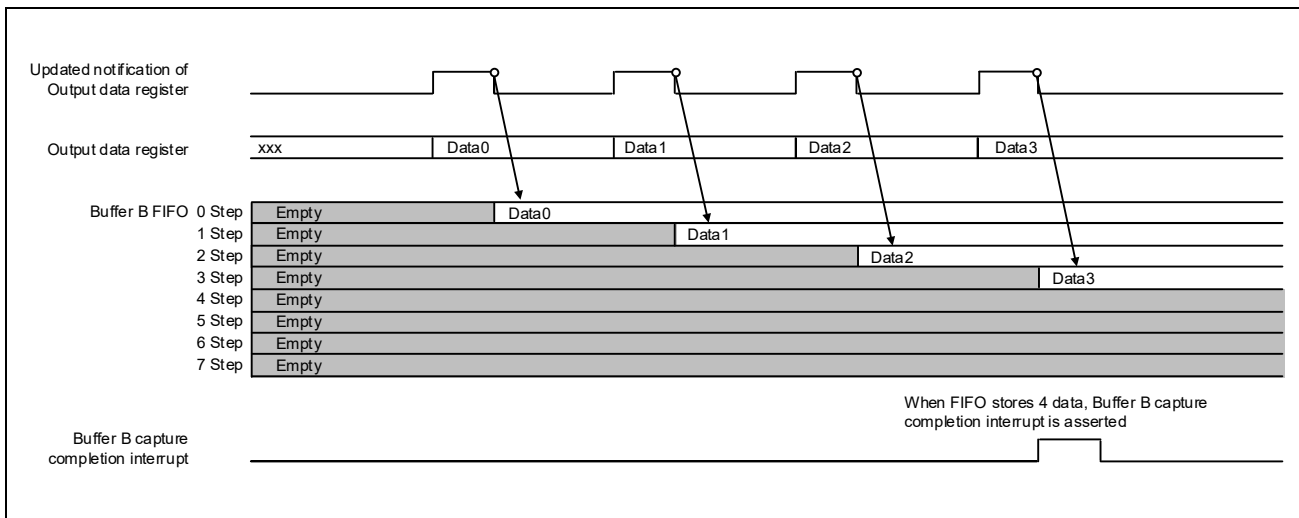


Figure 37.102 FIFO Macro Buffer B Capture End Interrupt Requests

(3) Error Interrupt Request

The error interrupt request of the FIFO macro indicates that an overflow was detected in the FIFO.

The error interrupt request is the OR of a buffer A FIFO channel overflow error and a buffer B overflow error.

Table 37.101 Summary of FIFO Macro Error Factors

Category	Error Status Bit	Remarks
Buffer A Circuit	DFBFACST.OVFAn (n = 0, 1, ..., 5)	FIFO Over Flow Error in Buffer A
Buffer B Circuit	DFBFBST.OVFB	FIFO Over Flow Error in Buffer B

The output of the FIFO macro error interrupt request can be controlled by setting the AIEE bit of the buffer A common control register (DFBFACCTL) to 1. The FIFO macro error interrupt request is issued if a FIFO overflow error occurs on a FIFO channel.

The output of the FIFO macro error interrupt request can be controlled by setting the BIEE bit of the buffer B control register (DFBFBCTL) to 1. The FIFO macro error interrupt request is issued if a FIFO overflow error occurs in the buffer B FIFO.

The FIFO macro error interrupt request is composed of the OR of error sources. Consequently, if an error interrupt request is issued once and then an error occurs due to another source, no new error interrupts will be issued.

37.6 Notes and Restrictions

37.6.1 Processing Time

37.6.1.1 Processing Time for One Channel

The digital filter executes a one-time FIR or IIR processing at a maximum of 0.7 μs . When there is no memory access competition. If memory access competition occurs, the processing time is a maximum of 1.2 μs .

37.6.1.2 Termination Processing Time

When the EN bit in DFEjCTLACHn is set to 0, the digital filter executes processing for all the input data that has been accepted and terminates processing. The maximum processing time up to the termination is obtained by “the number of channels for which input data has been accepted (number of channels for which processing has not been done) \times processing time”. For example, if five channels have not been processed, the termination processing time is a maximum of 3.5 μs .

37.6.2 DFE Activation Input Interval

An interval of DFE activation signal inputs should be a minimum of 0.7 μs if there is no memory access competition or a minimum of 1.2 μs if memory competition occurs.

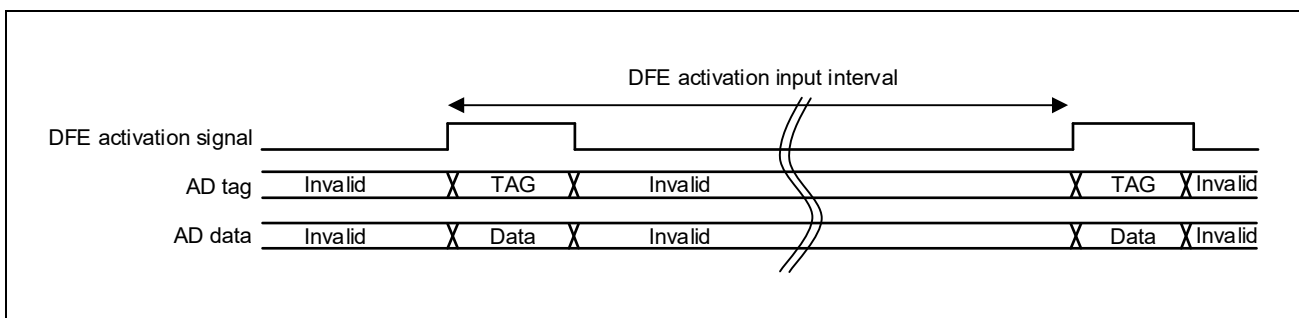


Figure 37.103 DFE Activation Input Interval (AD Input)

37.6.2.1 DFE Processing Time

The following two tables list the DFE processing time, i.e. the cycle from the input of a startup trigger from the AD until the next startup trigger on the same channel is accepted.

Table 37.102 Processing Time of FIR and IIR (with no initialization)

		Cycle
FIR (normal case)	8TAP	22 cycle
	16TAP	26 cycle
	24TAP	30 cycle
	32TAP	34 cycle
	64TAP	50 cycle
IIR (normal case)	Secondary biquad 1 stage	28 cycle
	Secondary biquad 2 stages	36 cycle
	Secondary biquad 3 stages	44 cycle
IIR (with gain adjustment function)	Secondary biquad 2 stages	42 cycle
	Secondary biquad 3 stages	56 cycle

Table 37.103 Processing Time of FIR and IIR (with Initialization)

		Time
FIR (with initialization)	8TAP	28 cycle
	16TAP	36 cycle
	24TAP	44 cycle
	32TAP	52 cycle
	64TAP	84 cycle
IIR (with initialization)	Secondary biquad 1 stage	38 cycle
	Secondary biquad 2 stages	54 cycle
	Secondary biquad 3 stages	70 cycle
IIR (with gain adjustment function)	Secondary biquad 2 stages	60 cycle
	Secondary biquad 3 stages	82 cycle

When the coefficient memory or data memory is accessed by the CPU during the DFE processing period, the access increases the total time for processing. Specifically, it takes more time per access.

37.6.3 Trigger Input

The following figure shows an example for an input of timer triggers 0 to 3 and a software trigger. When timer trigger 0 is used, for example, it is assumed that the DFE activation signal is input once or more times and timer trigger 0 is input again after the trigger flag function is enabled by the timer trigger 0 input.

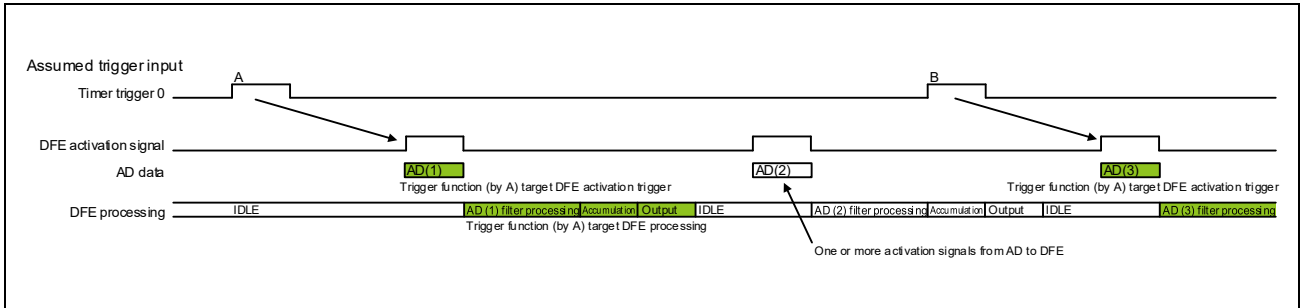


Figure 37.104 Example of Trigger Input

Input of a trigger from a timer or from software leads to setting of the corresponding trigger flag and the start of digital filter processing, but repeating input of the same trigger is prohibited.

In the example below, the trigger function by B is not enabled in some cases.

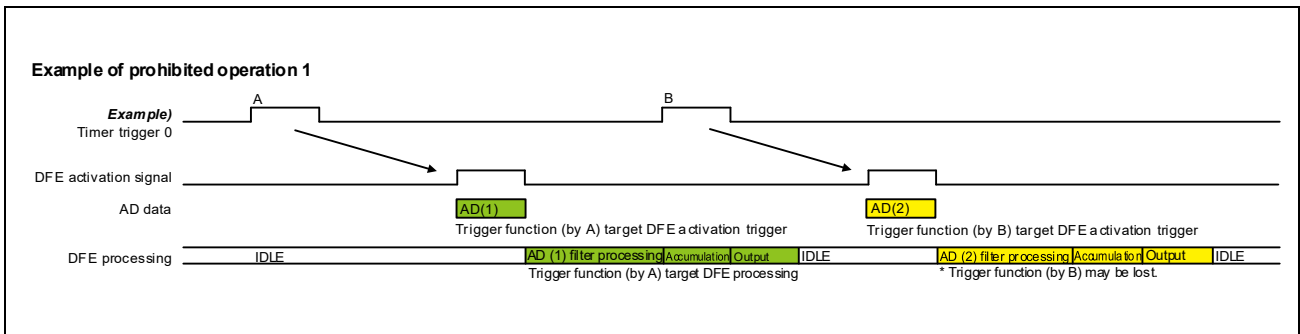


Figure 37.105 Operation When the Same Trigger is Prohibited 1

Before a timer trigger or a software trigger is input and the DFE activation signal is input, the same timer trigger or software trigger must not be input again. An example is shown below. In this case, two consecutive triggers are input at A and B, but are only effective as a single trigger from AD (2).

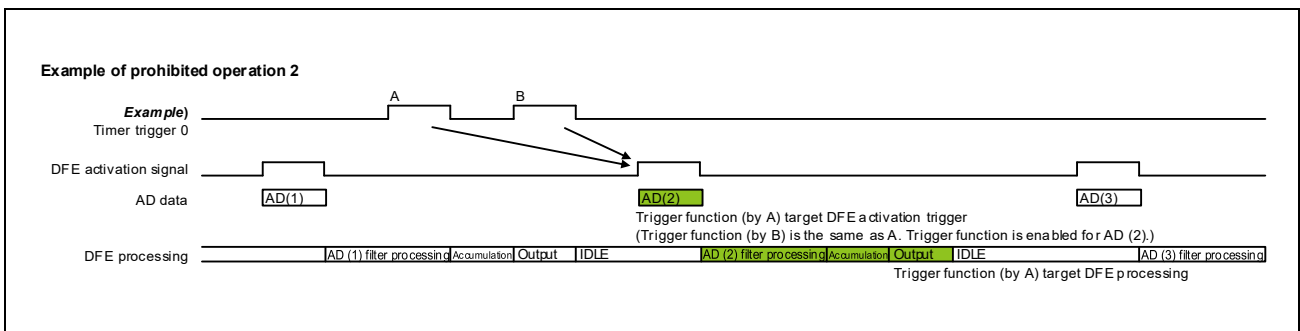


Figure 37.106 Operation When the Same Trigger is Prohibited 2

37.6.4 Channel Tag and AD Tag

If a tag value (AD tag) that the AD inputs to the digital filter and a tag value (channel tag) that the software sets for the digital filter are different, correct processing is disabled. For example, the AD inputs an AD tag value of 0001_B to the digital filter and there is no channel with a digital filter's channel tag value 00001_B, the DFE does not execute filter processing.

37.6.5 Input Data Format and Calculation Restrictions

Restrictions on available calculations are provided for the digital filter. For details, see **Section 37.5.6, Setting Control Registers**.

37.6.6 Restrictions on Cascade Processing

Cascade processing is a function to round 32-bit accumulation circuit output data or 32-bit subtraction circuit output data to 16-bit data to be input as data of other channels. The following restrictions are provided for cascade processing. Observe these restrictions together with the descriptions in **Section 37.5.6, Setting Control Registers**.

1. When cascade is enabled, do not set the same value for cascade tag and channel tag on the same channel. Assign a channel that is different from the channel for which cascade is specified as a cascade destination channel. If the same tag value is set, the filter processing result is input to the same channel again and calculation is repeated.
2. When cascade is enabled, the data format of cascade enabled channels must be the same as the data format of cascade destination channels. For example, when the fixed-point input format is specified for cascade enabled channels, the fixed-point input format is also specified for cascade destination channels.
3. Cascade input can be enabled for up to 15 channels on 16 channels. If cascade input is enabled for all channels, calculation is repeated for filter calculation results that are input in cascade.
4. The CAEN bits in the control register A (DFEjCTLACHn) must not be set to 01_B.
5. When peak-hold processing and comparison processing are enable, peak-hold processing and comparison processing are executed for 32-bit data.
6. No floating-point conversion result can be a cascade input.
7. When enabling the cascade function of two or more channels and selecting a single cascade destination channel, pay attention to the priority of channels. Specify cascade channels so that higher priority is given to cascade destination channels than cascade enabled channels. To enable the cascade function of channel 1 and channel 2 and select the same cascade destination for channel 1 and channel 2, for example, specify channel 0 as a cascade destination channel.
8. When enabling the cascade function of channel n and the cascade function of subtraction channel 0-2 selecting a single cascade destination channel, it is possible to be output the filter result and subtraction result at the same time. At this time, these are processed in the following priority. Low priority cascade input is not executed to cascade processing. And the cascade input is discarded.
 Cascade input of channel n > Cascade input of subtraction channel 0 > Cascade input of subtraction channel 1 > Cascade input of subtraction channel 2
9. When the subtraction circuit is enabled, the cascade rounding error is always judged with the subtraction results even if its cascade function is not used. Depending on the result of this judgement, the subtraction circuit cascade rounding error may be occurred. In this case, set the error mask bit (DFEjSUBERMCHn.MSKSCER) to 1.

37.6.7 Operation when the Given Channel is Disabled

37.6.7.1 Channel Enable Bits and Invalidation

While the channel enable bit (DFEjCTRLACHn.EN) is set to 0, filter processing is not activated even if the AD tag value that is input from the AD is equal to the channel tag value.

In addition, timer trigger input of the trigger flag function specified for the channel and software trigger input are not accepted.

When the channel enable bit (DFEjCTRLACHn.EN) is set to 0 from 1, the data to be processed that was input before this bit was set to 0 is processed.

At this time, if a trigger for the trigger flag function is input before the channel enable bit is set to 0, the trigger flag function is enabled for the data to be processed.

When the channel enable bit (DFEjCTRLACHn.EN) is 1, input trigger for the trigger flag function is held until AD data is input and the trigger flag function is enabled for the data to be processed (trigger held).

After the channel enable bit (DFEjCTRLACHn.EN) is set to 0 from 1, the held trigger is cleared if there is no data to be processed.

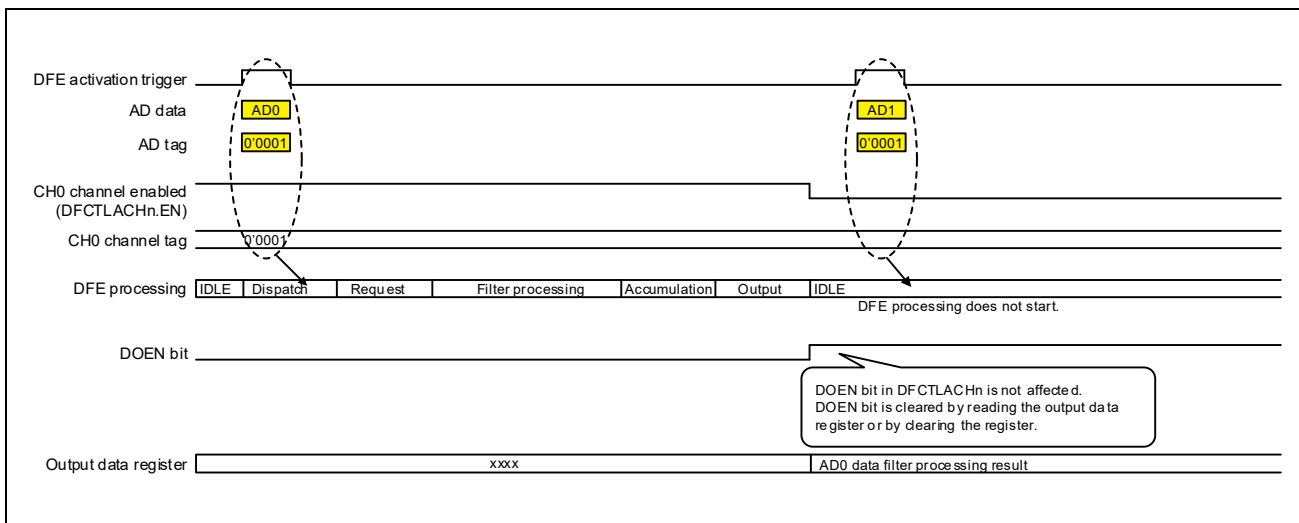


Figure 37.107 EN Bit and DOEN Bit

Even if the channel enable bit (EN in DFEjCTRLACHn) is cleared to 0, the status register is not cleared. Clear the status register with the software as necessary.

If the channel enable bit is set to 1 from 0 to restart the DFE processing:

- Accumulation/decimation counter starts at 0 in the same way as the first DFE processing after a reset.
- Comparison with the initial value register value is performed during the first peak-hold processing, and comparison with the peak-hold result register value is performed during the subsequent processing.

37.6.8 Restrictions on FIR 64TAP

If FIR64TAP is selected on an even-numbered channel (n) (DFEjCTRLACHn.CMD = 0111_B), no triggers or requests are accepted and are nullified on the next odd-numbered channel (n + 1).

37.6.9 Restrictions on Memory Access

8-bit or 16-bit write access to the coefficient memory is prohibited. Write data to the coefficient memory in 32 bits.

8 bit write access to data memory 0 and 1 is prohibited.

Upper/lower 16-bit write access to data memory 0 and 1 is enabled.

32-bit write access in batch to data memory 0 and 1 is enabled.

If a prohibited access is made, data may become corrupt. In the case of the coefficient memory, word data (32 bits) including a byte-write address may become corrupt. In the case of data memory 0 and data memory 1, half-word data (16 bits) including a byte-write address may become corrupt.

37.6.10 Restrictions on Trigger Setting Registers

A software trigger must not be specified for the accumulation/decimation initialization flag and the accumulation/decimation disable flag setting. Operations with prohibited settings are not guaranteed.

A software trigger must not be specified for the peak-hold initialization flag and the peak-hold end flag setting. Operations with prohibited settings are not guaranteed.

A software trigger must not be specified for the peak-hold mask start flag and the peak-hold mask end flag setting. Operation with prohibited setting is not guaranteed.

A software trigger must not be specified for the subtraction start flag and the subtraction end flag setting. Operation with prohibited setting is not guaranteed.

37.6.11 Setting of Peak-Hold Processing Disable Bit and Accumulation/Decimation Processing Disable Bit

When the EN bit in DFEjCTLACHn and VALID bit in DFEjSTCHn are both 0, the DISB bit (peak-hold processing disable bit) in DFEjCTLBCHn and DISA bit (accumulation/decimation processing disable bit) in DFEjCTLBCHn can be set by software.

When the PRCSB bit in DFEjCTLBCHn is set to 01_B (for peak-hold processing) or 11_B (for peak-hold processing and comparison processing simultaneously), peak-hold processing becomes invalid by writing 1 to the DISB bit in DFEjCTLBCHn. When the PRCSA bit is set to 01_B (for accumulation processing) or 10_B (for decimation processing), accumulation/decimation processing becomes invalid by writing 1 to the DISA bit in DFEjCTLBCHn.

37.6.12 Controlling Control Registers A and B

Set control registers A (DFEjCTLACHn) and B (DFEjCTLBCHn) in advance of trigger input. Setting control registers A and B is prohibited during digital filter operations while the value of the DFEjSTCHn.VALID or DFEjCTLACHn.EN1 bit is 1. If this is done, operations for digital filtering are not guaranteed.

Furthermore, as well as the above restrictions, writing to the intermediate value output register floating point conversion bit (DFEjCTLBCHn.HOFS) is prohibited while the value of the output data register enable bit (DFEjSTCHn.DOEN) is 1. Again, operations for digital filtering are not guaranteed if this is done.

37.6.13 Controlling the Peak-Hold Index Registers

A peak-hold index register retains its value even if the given channel is disabled (the DFEjCTRLACHn.EN bit is 0).

When Peak-hold circuit is set to peak-hold processing mode (DFEjCTLBCHn.PRC SB = 2b01), read a peak-hold index register, when the value of the condition matching bit (DFEjSTCHn.CND) is set to 1 after the value of the input data valid bit (DFEjSTCHn.VALID) has been 0. When Peak-hold circuit is set to peak-hold and comparison simultaneously mode (DFEjCTLBCHn.PRC SB = 2b11), read a peak-hold index register, when the value of the peak-hold end bit (DFEjSTCHn.PHE) has been 1 after the value of the input data valid bit (DFEjSTCHn.VALID) is set to 0.

37.6.14 Restrictions on the Intermediate Value Output Registers

Reading the intermediate value output registers (including the intermediate value output mirror register L) is prohibited for channels in use as IIR filters.

37.6.15 Restrictions on Subtraction Circuit

It is prohibited to set a minuend channel and a subtrahend channel to the same channel.

Use the same format for the filter results of the minuend and subtrahend channels.

It is prohibited to perform subtraction between the filter results with the different formats. In this case, data integrity is not guaranteed.

37.6.16 Restrictions on Buffer A Circuit

37.6.16.1 Restrictions on FIFO Channels In Use

Use it for ascending order from FIFO channel 0 consecutively by all means, when valid FIFO channels are less than the max FIFO channels which buffer A circuit has.

For example, set FIFO channel 0, 1, 2, and 3 to valid when 4 FIFO channels are used. It is prohibited to set the following settings.

- Setting FIFO channel 2, 3, 4, and 5 to valid (When not using FIFO channel 0)
- Setting FIFO channel 0, 2, 3, and 5 to valid (When not using FIFO channel in ascending order consecutively)

37.6.16.2 Notes about Read Registers

The output condition of buffer A capture interrupt request is the reading of the Buffer A output registers of all FIFO channels which are set to valid (DFBFACTLCHn.CHEN = 1).

Reading invalid FIFO channels is not included in the output condition for the interrupt request. The read value varies when according to the setting of DFBFACCTL.AUNE when reading invalid FIFO channels.

- The read value is as follows when output data of invalid FIFO channels is read.

When DFBFACCTL.AUNE is set to 0, read value is 32h00000000.

When DFBFACCTL.AUNE is set to 1, read value is 32h00000001.

- When valid FIFO channels are set to invalid, Set buffer A circuit is cleared once (Refer to **Section 37.4.2.1**). So the read value is as follows.

When DFBFACCTL.AUNE is set to 0, read value is 32h00000000.

When DFBFACCTL.AUNE is set to 1, read value is 32h00000001.

37.6.16.3 Notes about Buffer A Capture Interrupt Request

After an overflow error (DFBFACST.OVFAn (n = 0, 1, ..., 5)) has occurred on any of the FIFO channels, the buffer A capture interrupt request is not output even if the data of valid FIFO channels (DFBFACTLCHn.CHEN = 1) is read.

When an overflow error has occurred, clear buffer A circuit by using Buffer A clear bit (DFBFACLR.CLRA).

37.6.17 Restrictions on Buffer B Circuit

37.6.17.1 Notes Concerning the Buffer B Capture Interrupt Request

After an overflow error (DFBFBST.OVFB) has occurred, the buffer B capture interrupt request is not output even if FIFO data is read 4 times. When an overflow error has occurred, clear buffer B circuit by using Buffer B clear bit (DFBFBCLR.CLRB).

37.6.17.2 Restrictions on Buffer B Circuit when both Buffer A and Buffer B are in Use

When both Buffer A and Buffer B are used, it is prohibited to set same DFE and same DFE channel for Buffer A and Buffer B. For example, when 10CH of DFE0 is selected for FIFO channel 0 of Buffer A, it is prohibited to select 10CH of DFE0 for buffer B.

37.6.18 Note on Using Automatic Channel Initialization

Observe the following when using automatic channel initialization when restarting a channel:

- (1) Set the registers according to the channel start procedure described in **Section 37.4.2.1**.
- (2) If the channel is subject to capture by the buffer A circuit or buffer B circuit, set the registers according to the channel start procedure described in **Section 37.4.2.1**.

If either of the above are not observed, the following problems occur:

- Buffer A circuit:

If the DFBACLR.CLRA bit is not cleared after the buffer A circuit is stopped, the FIFO includes a mixture of the following data because a capture trigger is issued from the timer at the optionally set timing:

1. The filter results from before automatic channel initialization was enabled
2. Data that was cleared by automatic channel initialization
3. The filter results from after automatic channel initialization was enabled

This makes it impossible to judge the point from which data is valid.

- Buffer B circuit:

If the DFBACLR.CLRB bit is not cleared after the buffer B circuit is stopped, the FIFO includes a mixture of the following data because the data in the buffer B circuit is captured after the filter results of the target channel are output:

1. The filter results from before automatic channel initialization was enabled
2. The filter results from after automatic channel initialization was enabled

This makes it impossible to judge the point from which data is valid.

37.6.19 Restrictions on Using Peak-Hold Mask Function

When using the peak-hold mask function, be sure to disable the peak-hold mask function while peak-hold processing is enabled. Specifically, be sure to disable the peak-hold mask function by enabling the peak-hold mask end trigger flag before enabling the peak-hold end trigger flag.

37.6.20 Restrictions on Using Peak-Hold Function

When processing A and B below in the peak-hold circuit in parallel, allocate processing A to DFE_i and processing B to DFE_j. (where $i \neq j$)

When processing A and B in parallel in one DFE unit, use two DFE channels for processing A instead of one channel. Execute PH processing on one channel and comparison processing on the other channel.

- A) Both peak-hold processing and comparison processing simultaneously
- B) Peak-Hold-23 function

Section 38 Functional Safety

38.1 Overview

This section describes the intended safety mechanisms provided to detect the MCU failures with a short detection time. Here, the failures include both the recoverable transient failures such as soft errors of a memory and the unrecoverable permanent failures.

This MCU was developed to meet the requirements for a Safety Element out of Context (SEooC) as described in the ISO26262. In a SEooC development, safety requirements of target device (component) are assumed from market requirements by component manufacturers, and then the device is developed according to these assumed requirements. As this application is targeted for powertrain applications, some special considerations are already included to minimize gaps with powertrain requirements, however this should not restrict use with other applications. If Renesas assumptions are not enough to achieve system safety requirements derived from safety goals, the system safety requirements need to be modified. Contact our sale office for the details regarding the development process and safety organization.

The following lists the intended safety mechanisms provided by this product.

Note 1. The suitability of the safety mechanisms referenced in this chapter will finally be judged during the safety analysis and safety assessment of the MCU. Therefore, the safety documentation shall always be considered as additional requirements for the system or software derived from the results of the safety analysis.

ECC and EDC

Detect failures of memory and data transfer paths; correct failures.

Lockstep

Detects failures of the CPU and DTS with a short detection time.

Memory Protection

Detects erroneous access to the memory and peripheral circuits to protect the data in these elements against erroneous access.

BIST

Detects failures of the failure detection function itself.

MISG

Monitors write access to a specific address by the CPU, generates the signatures based on the written data, and automatically compares the generated signatures with each other.

Error Control Module (ECM)

Monitors various failure detection states in the MCU and defines the operation to be carried out when a failure occurs reaction.

Voltage Monitor

Monitors power supply voltage to detect over and under voltage.

Clock Monitor

Monitors the clock operation to detect abnormal operation.

Data CRC

Generates CRC to verify the data streams protected by CRC.

38.2 Reset Sources

Register reset conditions are shown in **Table 38.1**.

Table 38.1 Reset sources

Register Name	Reset Source					
	Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
Functional Safety related registers* ¹ (ECC and EDC, Hardware Redundancy, Memory Protection, and MISG)	✓	✓	✓	✓	—	—

Note 1. Except for BIST registers. For details about BIST register reset conditions, see **Section 38.6.2.3, Reset of Registers**. For details about register reset conditions of Voltage Monitor, Clock Monitor, ECM, and Data CRC Function, see **Section 12, Power Supply Voltage Monitor, Section 14, Clock Monitor, Section 39, Error Control Module (ECM), and Section 40, Data CRC Function B (DCRB)**.

38.3 ECC and EDC

38.3.1 Overview

38.3.1.1 ECC

This product incorporates ECC for the following memories. The ECC enables detection and correction of errors of the data retained in memory. The ECC also enables detection and correction of errors produced between the ECC encoder and memory; and memory and ECC decoder.

Table 38.2 ECC Overview

Applicable Memory	Applicable Data Width [bits]	Operation upon Error Detection				Failure Insertion
		Detection/Correction	Notice to ECM	Error Status	Address Capture	
Code flash	128	SEC-DED	Possible	Possible	Possible	Possible
Data flash	32	SEC-DED	Possible	Possible	Possible	Possible
Local RAM (all CPU cores)	32	SEC-DED	Possible	Possible	Possible	Possible
Cluster RAM	32	SEC-DED	Possible	Possible	Possible	Possible
Instruction cache (data)	64	SED-DED	Possible	Possible	Possible	Possible
Instruction cache (tag)	32	SED-DED	Possible	Possible	Possible	Possible
RAM for DTS	32	SEC-DED	Possible	Possible	Possible	Possible
RAM for sDMAC (Descriptor)	32	SEC-DED	Possible	Possible	Possible	Possible
RAM for sDMAC (Data Transfer)	64	SEC-DED	Possible	Possible	Possible	Possible
Peripheral RAM (32 bits)	32	SEC-DED	Possible	Possible* ¹	Possible* ¹	Possible* ¹
Peripheral RAM (16 bits)	16	SEC-DED	Possible	Possible	Possible	Possible
Data transfer path (data)	32	SEC-DED	Possible	Possible	N/A	Possible
Data transfer path (address)	32	SED-DED	Possible	Possible	N/A	Possible

Note 1. Except for Ethernet RAM.

Detection/Correction

SEC-DED: 1-bit errors can be detected and corrected, and 2-bit errors can only be detected.

SED-DED: 1-bit errors and 2-bit errors can only be detected.

Notice to ECM

A detected error can be reported to the ECM.

Error Status

The status of a detected error is retained.

Address Capture

The address of a detected error is retained.

Failure Insertion

An ECC error can be intentionally generated to enable software test of the ECC decoder operation.

38.3.1.2 Address Parity

This product incorporates address parity for the following types of memory. The address parity enables detection of errors during address decoding. It also enables detection of errors produced at addresses between the parity encoder and memory.

Table 38.3 Address Parity Overview

Applicable Memory	Parity Bit	Notice to ECM	Error Status	Address Capture	Failure Insertion
Code flash	1 bit	Possible	Possible	Possible	Possible

38.3.1.3 Address Feedback

This product incorporates an address feedback function for the following types of memory. The address feedback function enables detection of address errors inside memory module itself.

Table 38.4 Address Feedback Overview

Applicable Memory	Notice to ECM	Error Status	Address Capture	Failure Insertion
Local RAM (all CPU cores)	Possible	Possible	Possible	Possible
Cluster RAM	Possible	Possible	Possible	Possible
Instruction cache (data/tag)	Possible	Possible	Possible	Possible
RAM for DTS	Possible	Possible	Possible	Possible

38.3.2 Key Code Protection for ECC Control Registers

38.3.2.1 Overview

The product incorporates key code register to protect ECC control registers from unintended access.

38.3.2.2 List of Registers

Table 38.5 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
ECCKC (Peripheral Group0)	ECC Control Key Code Protection Register	KCPROT	FFFB 2800 _H	32	

38.3.2.3 KCPROT — ECC Control Key Code Protection Register

This register is used for protection against writing to the ECC control registers due to program malfunction and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.6 KCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit*2 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

Note 2. Unlocking this protection is one of necessary conditions to enable writing the protected registers. It is also necessary to unlock security protection driven by the ICUM when ICUM function is active. For details, see RH850/E2x-FCC1 ICUMD User's Manual: Hardware Section 9.

38.3.3 Code Flash ECC and Address Parity

38.3.3.1 Overview

The code flash ECC is summarized in the table below.

Table 38.7 Code Flash ECC

Item	Description
ECC error detection and correction	<p>ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected.</p> <ul style="list-style-type: none"> • ECC error detection and correction are carried out (2-bit error detection, and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>When disabled, neither error detection nor correction is carried out.</p> <p>In the initial state, this function is enabled; 1-bit error detection and correction and 2-bit error detection are carried out.</p>
Address parity	<p>Address parity check can be either enabled or disabled. When enabled, Address parity is checked during address decoding. When disabled, no address parity check is carried out. In the initial state, this function is enabled.</p>
Error notification	<p>The occurrence of an ECC error or a parity error is reported to the Error Control Module.</p> <p>ECC Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error.</p> <p>Address Parity Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an address parity error. <p>In the initial state, error notification is enabled upon detection of an address parity error.</p> <p>Overflow Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an address buffer overflow error for ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an address buffer overflow error.</p> <p>An ECC 2-bit error, an ECC 1-bit error, an address parity error, and an overflow error are handled as individual sources in ECM.</p> <p>An ECC 1-bit error signal is only issued to the ECM if the ECC 1-bit error address is not yet stored in the error address buffer.</p>
Error status	<p>A status register is provided that indicates the statuses of ECC 2-bit error detection, ECC 1-bit error detection, and address parity error detection. If an error occurs while no error status is set, the corresponding status is set.</p> <p>The error status can be cleared using the clear register.</p>
Address capture	<p>The address is captured when an ECC 2-bit error, an ECC 1-bit error, or an address parity error is detected. The error status serves as the enable bit of the capture address. Address buffers are updated if the error status is cleared.</p> <p>Multi-stage address buffers are provided for an ECC 1-bit error. ECC 1-bit error: Four stages ECC 2-bit error, address parity error shared: One stage</p>
Self-diagnosis	<p>The ROM data and the ECC and address parity bits in Code Flash can be read directly. Self-diagnosis of ECC and address parity is possible by reading data from ECC test area.</p>

38.3.3.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as an offset from the base addresses.

Table 38.8 Register Base Addresses

Base Address Name	Base Address	Bus Group
<ECCCNT_CFP_PE0CLO_base>	FFC4 8000 _H	Peripheral Group 0
<ECCCNT_CFP_PE1CLO_base>	FFC4 8080 _H	Peripheral Group 0
<ECCCNT_CFCCL0_base>	FFC4 8800 _H	Peripheral Group 0
<ECCCNT_CFS_base>	FFC4 8A00 _H	Peripheral Group 0
<MECCCAP_CFL_base>	FFC5 2400 _H	Peripheral Group 0

38.3.3.3 List of Registers

Table 38.9 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
ECCCNT_CFP_PEnCL0 (n = 0, 1)	ECC Control Register	CFPECCCTL	<ECCCNT_CFP_PEnCL0_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	CFPECCTSTCTL	<ECCCNT_CFP_PEnCL0_base> + 10 _H	8, 16, 32	KCPROT
ECCCNT_CFCCL0	Address Parity Control Register	CFCAPCTL	<ECCCNT_CFCCL0_base> + 00 _H	8, 16, 32	KCPROT
ECCCNT_CFS	ECC Control Register	CFSECCCTL	<ECCCNT_CFS_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	CFSECCTSTCTL	<ECCCNT_CFS_base> + 10 _H	8, 16, 32	KCPROT
MECCCAP_CFL	Error Notification Control Register	CF_ERRINT	<MECCCAP_CFL_base> + 00 _H	8, 16, 32	KCPROT
	1-bit Error Status Clear Register	CF_SSTCLR	<MECCCAP_CFL_base> + 10 _H	8, 16, 32	
	Fatal Error Status Clear Register	CF_DSTCLR	<MECCCAP_CFL_base> + 14 _H	8, 16, 32	
	1-bit Error Overflow Status Clear Register	CF_OVFCLR	<MECCCAP_CFL_base> + 18 _H	8, 16, 32	
	1-bit Error Status Register	CF_SERSTR	<MECCCAP_CFL_base> + 20 _H	8, 16, 32	
	Fatal Error Status Register	CF_DERSTR	<MECCCAP_CFL_base> + 24 _H	8, 16, 32	
	1-bit Error Overflow Status Register	CF_OVFSTR	<MECCCAP_CFL_base> + 28 _H	8, 16, 32	
	1-bit Error Location Information Register	CF_SERINF	<MECCCAP_CFL_base> + 30 _H	32	
	1st 1-bit Error Address Register	CF_00SEADR	<MECCCAP_CFL_base> + 70 _H	32	
	2nd 1-bit Error Address Register	CF_01SEADR	<MECCCAP_CFL_base> + 74 _H	32	
	3rd 1-bit Error Address Register	CF_02SEADR	<MECCCAP_CFL_base> + 78 _H	32	
	4th 1-bit Error Address Register	CF_03SEADR	<MECCCAP_CFL_base> + 7C _H	32	
	1st Fatal Error Address Register	CF_00DEADR	<MECCCAP_CFL_base> + F0 _H	32	

38.3.3.4 CFPECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for read data from Code Flash to PE.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.10 CFPECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	ECC 1-bit error correction disable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correction is enabled when 1-bit error is detected 1: Correction is disabled when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled.

38.3.3.5 CFPECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), address parity and ECC field written in Code Flash can be read from PE instead of data field.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 38.11 CFPECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects read data from Code Flash to PE. 0: Data field written in Code Flash normally 1: Address parity and ECC field written in Code Flash for self-diagnosis
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Instruction fetch from the Code Flash access path is not possible while ECC test mode is selected (ECCTST = 1). In ECC test mode including switching, the CPU must run a program from Local RAM or Cluster RAM and must not fetch instructions from Code Flash.

CPU has a data buffer for the Code Flash area. When switching the ECCTST bit, be sure to clear the data buffer. For how to clear the data buffer, see **Section 3, CPU System**.

When the Code Flash is read in ECC test mode, the ECC and address parity bits of the corresponding address are read. The bit assignment is as follows:

Table 38.12 Bit assignment of read data when ECC test mode is selected

Bit Position	Function
127 to 10	Always 0
9	Address parity bit
8 to 0	ECC bits

38.3.3.6 CFCAPCTL — Address Parity Control Register

This register controls address parity error detection for read data from Code Flash.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	APEDIS	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R

Table 38.13 CFCAPCTL Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	APEDIS	Address parity error disable bit Sets address parity error detection to enable/disable. 0: Address parity error detection is enabled 1: Address parity error detection is disabled
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.3.7 CFSECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for read data to SAXI2FAXI bridge.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.14 CFSECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	ECC 1-bit error correction enable bit Enables/disables 1-bit error correction when using ECC error detection/correction (ECCDIS = 0). 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Enables/disables ECC error detection/correction. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

38.3.3.8 CFSECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), address parity and ECC field written in Code Flash can be read from SAXI2FAXI bridge instead of data field.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 38.15 CFSECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects read data from Code Flash to SAXI2FAXI bridge. 0: Data field written in Code Flash normally 1: Address parity and ECC field written in Code Flash for self-diagnostics
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Instruction fetch from the Code Flash access path is not possible while ECC test mode is selected (ECCTST = 1). In ECC test mode including switching, ICU-M must run a program from Local RAM or Cluster RAM and must not fetch instructions from Code Flash.

When the Code Flash is read in ECC test mode, the ECC and address parity bits of the corresponding address are read. The bit assignment is as follows:

Table 38.16 Bit assignment of read data when ECC test mode is selected

Bit Position	Function
127 to 10	Always 0
9	Address parity bit
8 to 0	ECC bits

38.3.3.9 CF_ERRINT — Error Notification Control Register

This register controls whether error information is reported to the ECM when an address parity error, ECC error and/or overflow error has occurred.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0087_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEOVF IE	—	—	—	—	APEIE	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R/W	R/W	R/W

Table 38.17 CF_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	SEOVFIE	Controls error reports when overflow error occurs. 0: Overflow error report disabled 1: Overflow error report enabled
6 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	APEIE	Controls error reports when address parity error is detected. 0: Address parity error report disabled 1: Address parity error report enabled
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

38.3.3.10 CF_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in CF_SERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SST CLR03	SST CLR02	SST CLR01	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 38.18 CF_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	SSTCLR03	Writing 1 to this bit clears SEDF03 in CF_SERSTR.
2	SSTCLR02	Writing 1 to this bit clears SEDF02 in CF_SERSTR.
1	SSTCLR01	Writing 1 to this bit clears SEDF01 in CF_SERSTR.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in CF_SERSTR.

38.3.3.11 CF_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in CF_DERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.19 CF_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears APEF00 and DEDF00 in CF_DERSTR.

38.3.3.12 CF_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in CF_OVFSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.20 CF_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in CF_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in CF_OVFSTR.

38.3.3.13 CF_SERSTR — 1-bit Error Status Register

This register indicates whether an ECC 1-bit error has occurred. The location and address of the error that is detected is stored in CF_nSEADR register when SEDFn flag is set. The SEDFn flag is set only when a unique ECC 1-bit error, which is different from previous errors stored in 1-bit error address registers, is detected while the SEDFn is “0”. If the newly detected error has the same address with errors already stored, this flag is not set.

This register can be cleared by SSTCLRn in CF_SSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SEDF03	SEDF02	SEDF01	SEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.21 CF_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	SEDF03	Indicates that an ECC 1-bit error was detected and error address is stored in CF_03SEADR register.
2	SEDF02	Indicates that an ECC 1-bit error was detected and error address is stored in CF_02SEADR register.
1	SEDF01	Indicates that an ECC 1-bit error was detected and error address is stored in CF_01SEADR register.
0	SEDF00	Indicates that an ECC 1-bit error was detected and error address is stored in CF_00SEADR register.

38.3.3.14 CF_DERSTR — Fatal Error Status Register

This register indicates whether an address parity error and/or an ECC 2-bit error has occurred. The location and address of the error that is detected are stored in CF_00DEADR register when APEF00 and/or DEDF00 flags are set. The APEF00 and/or DEDF00 flag is set only when an address parity error and/or an ECC 2-bit error is detected while all the flags are “0”. Multiple flags are set only when multiple errors occur due to one error cause.

This register can be cleared by DSTCLR00 in CF_DSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	APEF00	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.22 CF_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	APEF00	Indicates that an address parity error was detected.
1	DEDF00	Indicates that an ECC 2-bit error was detected.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.3.15 CF_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether an overflow of the address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address is detected when the address buffer is fully used. If the newly detected error has the same address with errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flags can be cleared by SERROVFCLR0 and SERROVFCLR1 in CF_OVFCLR register respectively.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.23 CF_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVF1	Indicates that ECC 1-bit error s over working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error was detected when all ECC 1bit error flags in CF_SERSTR are set.

38.3.3.16 CF_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in CF_SERSTR register were detected.

This register is updated whenever CF_SERSTR register is updated.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDLIN F16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SEDLIN F03	SEDLIN F02	SEDLIN F01	SEDLIN F00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.24 CF_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	SEDLINF16	Indicates that an ECC 1-bit error was detected in read data from Code Flash to SAXI2FAXI.
15 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	SEDLINF03	Indicates that an ECC 1-bit error was detected in read data loaded from Code Flash to PE1.
2	SEDLINF02	Indicates that an ECC 1-bit error was detected in read data fetched from Code Flash to PE1.
1	SEDLINF01	Indicates that an ECC 1-bit error was detected in read data loaded from Code Flash to PE0.
0	SEDLINF00	Indicates that an ECC 1-bit error was detected in read data fetched from Code Flash to PE0.

38.3.3.17 CF_nSEADR — n-th 1-bit Error Address Register (n = 00 to 03)

This register is used to hold the address and the location of the error when the SEDFn flag is not set and an ECC 1-bit error is detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]					—	SEADR0[25:16]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SEADR0[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.25 CF_nSEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25 to 2	SEADR0[25:2]	Indicates at which address this error was detected.
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

NOTE

SEADR0 indicates the address right-shifted by 2 bits rather than the actual address, so it is necessary for users to multiply it by 4 to get address at which the error was detected.

Please refer to the following table to confirm the detailed information which each SEDL indicates.

Table 38.26 SEDL information

SEDL[4:0]	Indicated information
31 to 17	Reserved
16	Indicates an ECC 1-bit error is detected in loading data from Code Flash by any master except PEs.
15 to 4	Reserved
3	Indicates an ECC 1-bit error is detected in loading data from Code Flash by PE1 master.
2	Indicates an ECC 1-bit error is detected in fetching instructions from Code Flash by PE1 master.
1	Indicates an ECC 1-bit error is detected in loading data from Code Flash by PE0 master.
0	Indicates an ECC 1-bit error is detected in fetching instructions from Code Flash by PE0 master.

38.3.3.18 CF_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when neither APEF00 nor DEDF00 flag is set and an address parity error and/or ECC 2-bit error was detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]					—	DEADR0[25:16]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEADR0[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.27 CF_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25 to 2	DEADR0[25:2]	Indicates at which address this error was detected.
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

NOTE

DEADR0 indicates the address right-shifted by 2 bits rather than the actual address, so it is necessary for users to multiply it by 4 to get address at which the error was detected.

Please refer to the following table to confirm the detailed information which each DEDL indicates.

Table 38.28 DEDL Information

DEDL[4:0]	Indicated information
31 to 17	Reserved
16	Indicates a fatal error is detected in loading data from Code Flash by any master except PEs.
15 to 4	Reserved
3	Indicates a fatal error is detected in loading data from Code Flash by PE1 master.
2	Indicates a fatal error is detected in fetching instructions from Code Flash by PE1 master.
1	Indicates a fatal error is detected in loading data from Code Flash by PE0 master.
0	Indicates a fatal error is detected in fetching instructions from Code Flash by PE0 master.

38.3.3.19 Test Function

Through appropriate register settings, the code flash data, ECC bits, and address parity bit can be read out.

(1) Reading Code Flash

- (a) Set the ECCDIS bit in the CFPECCCTL register to 1 to disable ECC error detection and correction.
- (b) When ECCDIS=1, neither error detection nor correction proceeds when the code flash is read; the data output from the code flash is read unchanged.

How to exit this test mode:

- (a) Set the ECCDIS bit in the CFPECCCTL register to 0 to enable ECC error detection and correction.

(2) Reading the ECC and address parity bits

- (a) Set the ECCDIS bit in the CFPECCCTL register to 1 to disable ECC error detection and correction.
- (b) Set the ECCTST bit in the CFPECCTSTCTL register to 1 to set test mode.
- (c) When the code flash is read, the ECC and address parity bits are read instead of the code flash data.

How to exit this test mode:

- (a) Set the ECCDIS bit in the CFPECCCTL register to 0 to enable ECC error detection and correction.
- (b) Set the ECCTST bit in the CFPECCTSTCTL register to 0 to set normal mode.

(3) Self-diagnosis of ECC and address parity check function

Self-diagnosis of the ECC decoder and address parity checker for the access ports is possible by reading data from ECC test area.

For details about the ECC test area, refer to *RH850/E2x Safety Application Note*.

38.3.4 Data Flash ECC

38.3.4.1 Overview

The data flash ECC is summarized in the table below.

Table 38.29 Data Flash ECC

Item	Description
ECC error detection and correction	<p>ECC error detection and correction can be either enabled or disabled.</p> <p>When enabled, either of the following settings can be selected.</p> <ul style="list-style-type: none"> • ECC error detection and correction are carried out (2-bit error detection, and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>When disabled, neither error detection nor correction is carried out.</p> <p>In the initial state, this function is enabled; 1-bit error detection and correction, and 2-bit error detection are carried out.</p>
Error notification	<p>Upon occurrence of an ECC error, it is notified to the Error Control Module.</p> <p>ECC Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of a 2-bit ECC error. • Error notification can be either enabled or disabled upon detection of a 1-bit ECC error. <p>In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error.</p> <p>An ECC 2-bit error, an ECC 1-bit error and an overflow error are handled as individual sources in ECM.</p>
Error status	<p>A status register is provided that indicates the statuses of ECC 2-bit error detection and ECC 1-bit error detection. If an error occurs while no error status is set, the corresponding status is set.</p> <p>The error status can be cleared using the clear register.</p>
Address capture	<p>The address is captured when an ECC 2-bit error or an ECC 1-bit error is detected. The error status serves as the enable bit of the capture address.</p> <p>If an ECC error occurs while no error status is set, the address at which the associated error has occurred is captured.</p>
Self-diagnosis	<p>The ROM data and the ECC bits in Data Flash can be read directly.</p> <p>Desired values can be written to the data and the ECC bits in Data Flash.</p>

38.3.4.2 List of Registers

Table 38.30 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
ECCDF (Peripheral Group1)	Data Flash ECC Control Register	DFECCCTL	FFC6 2C00 _H	16, 32	DFKCPROT
	Data Flash Error Status Register	DFERSTR	FFC6 2C04 _H	32	
	Data Flash Error Status Clear Register	DFERSTC	FFC6 2C08 _H	8, 16, 32	
	Data Flash Error Overflow Status Register	DFOVFSTR	FFC6 2C0C _H	32	
	Data Flash Error Overflow Status Clear Register	DFOVFSTC	FFC6 2C10 _H	8, 16, 32	
	Data Flash Error Notification Control Register	DFERRINT	FFC6 2C14 _H	8, 16, 32	DFKCPROT
	Data Flash 1st Error Address Register	DFEADR	FFC6 2C18 _H	32	
	Data Flash Test Control Register	DFTSTCTL	FFC6 2C1C _H	16, 32	DFKCPROT
	Data Flash ECC Key Code Protection Register	DFKCPROT	FFC6 2C20 _H	32	

38.3.4.3 DFECCTL — Data Flash ECC Control Register

DFECCTL enables or disables ECC error detection and 1-bit error correction for read access.

DFECCTL register is protected by DFKCPROT register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.31 DFECCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	ECC 1-bit error correction disable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correction is enabled when 1-bit error is detected. 1: Correction is disabled when 1-bit error is detected.
0	ECCDIS	ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled. 1: ECC error detection/correction is disabled.

38.3.4.4 DFERSTR — Data Flash Error Status Register

DFERSTR monitors occurrence of errors.

The SEDF bit is set if an ECC 1-bit error is detected while ECC error detection/correction is enabled (ECCDIS = 0), and the DEDF bit is set if an ECC 2-bit error is detected.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF	SEDF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.32 DFERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDF	ECC 2-Bit Error Monitor Flag 0: An ECC 2-bit error is not generated. 1: An ECC 2-bit error is generated. Clearing condition: ERRCLR bit is set in Data Flash error status clear register. Setting condition: ECC 2-bit error is generated.
0	SEDF	ECC 1-bit error Monitor Flag 0: An ECC 1-bit error is not generated. 1: An ECC 1-bit error is generated. Clearing condition: ERRCLR bit is set in Data Flash error status clear register. Setting condition: ECC 1-bit error is generated with both SEDF and DEDF being 0.

38.3.4.5 DFERSTC — Data Flash Error Status Clear Register

DFERSTC clears the error flags in the Data Flash error status register. DFERSTC is a write-only register and is always read as 0.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERR CLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.33 DFERSTC Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ERRCLR	SEDF/DEDF Flag Clear 0: No effect (Setting the ERRCLR bit to 0 does not affect the DEDF and SEDF flags in DFERSTR.) 1: The SEDF/DEDF flag in DFERSTR is cleared.

38.3.4.6 DFOVFSTR — Data Flash Error Overflow Status Register

DFOVFSTR monitors occurrence of Data Flash error overflow.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERR OVF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.34 DFOVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ERROVF	<p>Error Overflow Flag</p> <p>ERROVF is set if the following occur:</p> <ul style="list-style-type: none"> • An ECC 1-bit error occurs when DFERSTR.SEDF = 1 and access address is different from DFEADR. • An ECC 1-bit error occurs when DFERSTR.DEDF = 1 • An ECC 2-bit error occurs when DFERSTR.SEDF = 1 • An ECC 2-bit error occurs when DFERSTR.DEDF = 1 and access address is different from DFEADR. <p>0: Did not occur 1: Occurred</p> <p>Clearing condition: Set the ERROVFCLR bit in DFOVFSTC to 1.</p>

38.3.4.7 DFOVFSTC — Data Flash Error Overflow Status Clear Register

DFOVFSTC clears the Data Flash error overflow flag.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERR OVF CLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.35 DFOVFSTC Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ERROVFCLR	Error Overflow Flag Clear 0: No effect (Setting the ERROVFCLR bit to 0 does not affect the flag in DFOVFSTR.) 1: The ERROVF flag in the DFOVFSTR register is cleared.

38.3.4.8 DFERRINT — Data Flash Error Notification Control Register

DFERRINT enables or disables generation of the error notification signal upon detection of an ECC 2-bit error or an ECC 1-bit error.

DFERRINT register is protected by DFKCPROT register.

Value after reset: 0000 0007_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	EOVFIE	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 38.36 DFERRINT Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	EOVFIE	ECC Error Address Overflow Notification Control Enables or disables generation of the error notification signal upon detection of an address buffer overflow error. 0: Error address overflow report disabled 1: Error address overflow report enabled
1	DEDIE	ECC 2-bit error Notification Control Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection/correction is enabled (ECCDIS = 0). 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-bit error Notification Control Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection/correction is enabled (ECCDIS = 0). 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

38.3.4.9 DFEADR — Data Flash 1st Error Address Register

DFEADR holds the address if the following occur:

- An ECC 1-bit error occurs when DFERSTR.SEDF = 0 and DFERSTR.DEDF = 0
- An ECC 2-bit error occurs when DFERSTR.DEDF = 0

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	DFEADR[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DFEADR[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.37 DFEADR Register Contents

Bit Position	Bit Name	Function
31 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20 to 2	DFEADR[20:2]	ECC Error Address DFEADR is a read-only field to monitor the address at which an ECC error has occurred. This register holds an internal address. Convert it to the actual address by adding the data flash base address described in Section 4 Address MAP.
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.4.10 DFTSTCTL — Data Flash Test Control Register

DFTSTCTL is used for ECC testing.

The data of the ECC bit can be read after setting the ECC test mode (ECCTST = 1).

DFTSTCTL register is protected by DFKCPROT register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 38.38 DFTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ECCTST	ECC Test By setting ECC test mode bit to “1” (ECCTST = 1), CPU can read ECC bit.

38.3.4.11 DFKCPROT — Data Flash ECC Key Code Protection Register

DFKCPROT is used for access protection of other Data Flash ECC registers.

Protected registers:

DFECCCTL

DFERRINT

DFTSTCTL

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KCPCROT[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KCPCROT[15:1]																KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.39 DFKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPCROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.* ¹
0	KCE	Key Code Enable bit* ² 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.

Write A5A5A501_H to this register to enable writing protected registers.

Note 2. Unlocking this protection is one of necessary conditions to enable writing the protected registers. It is also necessary to unlock security protection driven by the ICUM when ICUM function is active. For details, see RH850/E2x-FCC1 ICUMD User's Manual: Hardware Section 9.

38.3.4.12 Test Function

Data in the ROM and the ECC bits can be read by setting the data flash test control register (DFTSTCTL).

(1) Reading the ROM data

- (a) Set the ECCDIS bit in the data flash ECC control register to 1 to disable ECC error detection and correction.
- (b) When ECCDIS = 1, neither error detection nor correction proceeds when the data flash is read; the data output from the data flash is read unchanged.

How to exit this test mode:

- (a) Set the ECCDIS bit in the data flash ECC control register to 0 to enable ECC error detection and correction.

(2) Reading the ECC bits

- (a) Set the ECCDIS bit in the data flash ECC control register to 1 to disable ECC error detection and correction.
- (b) Set the ECCTST bit in the data flash test control register to 1 to set test mode.
- (c) When the data flash is read, the 7 lower-order bits of read data are read as ECC data.

How to exit this test mode:

- (a) Set the ECCDIS bit in the data flash ECC control register to 0 to enable ECC error detection and correction.
- (b) Set the ECCTST bit in the data flash test control register to 0 to set normal mode.

(3) Self-diagnosis of ECC check function

Self-diagnosis of the ECC decoder is possible by writing incorrect data to the data flash memory beforehand (fault injection) and then reading this data. A 1- or 2-bit ECC error fault can be injected by generating correct ECC bits once and inverting only the appropriate bits. For details on programming of the data flash, refer to *the RH850/E2x-FCC1 Flash Memory User's Manual: Hardware Interface*.

38.3.5 Local RAM ECC and Address Feedback

38.3.5.1 Overview

Local RAM ECC is summarized in the table below.

Table 38.40 Local RAM ECC

Item	Description
ECC error detection and correction	<p>ECC error detection and correction can be either enabled or disabled.</p> <p>When enabled, either of the following settings can be selected:</p> <ul style="list-style-type: none"> • ECC error detection and correction are carried out (2-bit error detection, and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>When disabled, neither error detection nor correction is carried out.</p> <p>In the initial state, this function is enabled; 1-bit error detection and correction, and 2-bit error detection are carried out.</p>
Address feedback	<p>Address feedback check can be either enabled or disabled.</p> <p>When enabled, Address feedback error detection is carried out.</p> <p>When disabled, no error detection is carried out.</p> <p>In the initial state, this function is enabled.</p>
Error notification	<p>The occurrence of an ECC error or an address feedback error is reported to the Error Control Module.</p> <p>ECC Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error.</p> <p>Address Feedback Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an address feedback error. <p>In the initial state, error notification is enabled upon detection of an address feedback error.</p> <p>Overflow Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an address buffer overflow error for ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an address buffer overflow error. The error notification signal is output with an address feedback error handled as one source. An ECC 2-bit error, an ECC 1-bit error and an overflow error are handled as individual sources in ECM. An ECC 1-bit error signal is only issued to the ECM, if the ECC 1-bit error address is not yet stored in the error address buffer.</p>
Error status	<p>A status register is provided that indicates the statuses of ECC 2-bit error detection, ECC 1-bit error detection, and address feedback error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.</p>
Address capture	<p>The address is captured when an ECC 2-bit error, an ECC 1-bit error, or an address feedback error is detected. The error status serves as the enable bit of the address that is captured.</p> <p>Address buffers are updated if the error status is cleared.</p> <p>Multi-stage address buffers are provided for an ECC 1-bit error.</p> <p>ECC 1-bit error: Eight stages (for each CPU)</p> <p>ECC 2-bit error and address feedback error: One stage (for each CPU)</p>
Self-diagnosis	<p>Desired values can be written to the RAM data and the ECC bits.</p> <p>The RAM data can be read directly and the ECC bits can be read via Read Buffer Register in ECC Test Mode.</p> <p>Moreover, the error injection to an address feedback checker is possible by setting AFINV.</p>

ECC of Local RAM is provided for each 32 bits of data and each 32-bit data is called bank 0 to 3. The relationship between addresses and bank numbers are as follows.

Table 38.41 Relationship between addresses and bank numbers

4 lower-order bits of address (Hexadecimal Notation)	F _H to C _H	B _H to 8 _H	7 _H to 4 _H	3 _H to 0 _H
Bank number	Bank 3	Bank 2	Bank 1	Bank 0

38.3.5.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offsets from the base addresses.

Table 38.42 Register Base Addresses

Base Address Name	Base Address	Bus Group
<ECCCNT_LR_PE0CL0_base>	FFC4 9800 _H	Peripheral Group 0
<ECCCNT_LR_PE1CL0_base>	FFC4 9880 _H	Peripheral Group 0
<MECCCAP_LR_PE0CL0_base>	FFC5 0000 _H	Peripheral Group 0
<MECCCAP_LR_PE1CL0_base>	FFC5 0100 _H	Peripheral Group 0
<MECCCAP_LRA_base>	FFC5 0800 _H	Peripheral Group 0

38.3.5.3 List of Registers

Table 38.43 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection
ECCCNT_LR_PEnCL0 (n = 0, 1)	ECC Control Register	LRECCCTL	<ECCCNT_LR_PEnCL0_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	LRECCSTCTL	<ECCCNT_LR_PEnCL0_base> + 10 _H	8, 16, 32	KCPROT
	Address Feedback Test Control Register	LRAFINV	<ECCCNT_LR_PEnCL0_base> + 20 _H	8, 16, 32	KCPROT
	ECC Read Buffer Register	LRTDATABFECCF	<ECCCNT_LR_PEnCL0_base> + 60 _H	8, 16, 32	
MECCCAP_LR_PEnCL0 (n = 0, 1)	Error Notification Control Register	LR0_ERRINT	<MECCCAP_LR_PEnCL0_base> + 00 _H	8, 16, 32	KCPROT
	1-bit Error Status Clear Register	LR0_SSTCLR	<MECCCAP_LR_PEnCL0_base> + 10 _H	8, 16, 32	
	Fatal Error Status Clear Register	LR0_DSTCLR	<MECCCAP_LR_PEnCL0_base> + 14 _H	8, 16, 32	
	1-bit Error Overflow Status Clear Register	LR0_OVFCLR	<MECCCAP_LR_PEnCL0_base> + 18 _H	8, 16, 32	
	1-bit Error Status Register	LR0_SERSTR	<MECCCAP_LR_PEnCL0_base> + 20 _H	8, 16, 32	
	Fatal Error Status Register	LR0_DERSTR	<MECCCAP_LR_PEnCL0_base> + 24 _H	8, 16, 32	
	1-bit Error Overflow Status Register	LR0_OVFSTR	<MECCCAP_LR_PEnCL0_base> + 28 _H	8, 16, 32	
	1-bit Error Location Information Register	LR0_SERINF	<MECCCAP_LR_PEnCL0_base> + 30 _H	32	
	1st 1-bit Error Address Register	LR0_00SEADR	<MECCCAP_LR_PEnCL0_base> + 70 _H	32	
	2nd 1-bit Error Address Register	LR0_01SEADR	<MECCCAP_LR_PEnCL0_base> + 74 _H	32	
	3rd 1-bit Error Address Register	LR0_02SEADR	<MECCCAP_LR_PEnCL0_base> + 78 _H	32	
	4th 1-bit Error Address Register	LR0_03SEADR	<MECCCAP_LR_PEnCL0_base> + 7C _H	32	
	5th 1-bit Error Address Register	LR0_04SEADR	<MECCCAP_LR_PEnCL0_base> + 80 _H	32	
	6th 1-bit Error Address Register	LR0_05SEADR	<MECCCAP_LR_PEnCL0_base> + 84 _H	32	

Table 38.43 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection
MECCCAP_LR_PEnCL0 (n = 0, 1)	7th 1-bit Error Address Register	LR0_06SEADR	<MECCCAP_LR_PEnCL0_base> + 88 _H	32	
	8th 1-bit Error Address Register	LR0_07SEADR	<MECCCAP_LR_PEnCL0_base> + 8C _H	32	
	1st Fatal Error Address Register	LR0_00DEADR	<MECCCAP_LR_PEnCL0_base> + F0 _H	32	
MECCCAP_LRA	Error Notification Control Register	LR1_ERRINT	<MECCCAP_LRA_base> + 00 _H	8, 16, 32	KCPROT
	1-bit Error Status Clear Register	LR1_SSTCLR	<MECCCAP_LRA_base> + 10 _H	8, 16, 32	
	Fatal Error Status Clear Register	LR1_DSTCLR	<MECCCAP_LRA_base> + 14 _H	8, 16, 32	
	1-bit Error Overflow Status Clear Register	LR1_OVFCLR	<MECCCAP_LRA_base> + 18 _H	8, 16, 32	
	1-bit Error Status Register	LR1_SERSTR	<MECCCAP_LRA_base> + 20 _H	8, 16, 32	
	Fatal Error Status Register	LR1_DERSTR	<MECCCAP_LRA_base> + 24 _H	8, 16, 32	
	1-bit Error Overflow Status Register	LR1_OVFSTR	<MECCCAP_LRA_base> + 28 _H	8, 16, 32	
	1-bit Error Location Information Register	LR1_SERINF	<MECCCAP_LRA_base> + 30 _H	32	
	1st 1-bit Error Address Register	LR1_00SEADR	<MECCCAP_LRA_base> + 70 _H	32	
	2nd 1-bit Error Address Register	LR1_01SEADR	<MECCCAP_LRA_base> + 74 _H	32	
	3rd 1-bit Error Address Register	LR1_02SEADR	<MECCCAP_LRA_base> + 78 _H	32	
	4th 1-bit Error Address Register	LR1_03SEADR	<MECCCAP_LRA_base> + 7C _H	32	
	5th 1-bit Error Address Register	LR1_04SEADR	<MECCCAP_LRA_base> + 80 _H	32	
	6th 1-bit Error Address Register	LR1_05SEADR	<MECCCAP_LRA_base> + 84 _H	32	
	7th 1-bit Error Address Register	LR1_06SEADR	<MECCCAP_LRA_base> + 88 _H	32	
8th 1-bit Error Address Register	LR1_07SEADR	<MECCCAP_LRA_base> + 8C _H	32		
1st Fatal Error Address Register	LR1_00DEADR	<MECCCAP_LRA_base> + F0 _H	32		

38.3.5.4 LRECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for read data from Local RAM and address feedback error detection for request address to Local RAM.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0004_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFE DIS	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W

Table 38.44 LRECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AFEDIS	Address feedback error disable bit Sets address feedback error detection to enable/disable. 0: Address feedback error detection is enabled 1: Address feedback error detection is disabled
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

38.3.5.5 LRECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), either data field or ECC field of Local RAM can be written separately.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	DAT SEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.45 LRECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Sets the write mode to Local RAM. 0: Normal mode 1: Test mode Either data field or ECC field can be written according to the DATSEL setting in test mode.
0	DATSEL	Sets the field to write data when ECCTST = 1. 0: Data field only 1: ECC field only

38.3.5.6 LRAFINV — Address Feedback Test Control Register

This register is used to inject errors into the feedback address from Local RAM for self-diagnosis. Feedback address XOR-ed with AFINV can be input to address feedback checker.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	AFINV[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AFINV[15:4]											—	—	—	—	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 38.46 LRAFINV Register Contents

Bit Position	Bit Name	Function
31 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20 to 4	AFINV[20:4]	Specifies bit pattern to inject errors into feedback address.
3 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.5.7 LRTDATBFECF — ECC Read Buffer Register

This register is used to hold ECC field read from Local RAM for self-diagnosis. ECC field in requested address can be stored this register while reading from Local RAM where ECCTST = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	BFEC3[6:0]						—	BFEC2[6:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BFEC1[6:0]						—	BFEC0[6:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.47 LRTDATBFECF Register Contents

Bit Position	Bit Name	Function
31	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
30 to 24	BFEC3[6:0]	ECC field read from the address that is requested in Local RAM bank 3
23	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	BFEC2[6:0]	ECC field read from the address that is requested in Local RAM bank 2
15	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	BFEC1[6:0]	ECC field read from the address that is requested in Local RAM bank 1
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	BFEC0[6:0]	ECC field read from the address that is requested in Local RAM bank 0

38.3.5.8 LR0_ERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address feedback error, ECC error and/or overflow error has occurred.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 008F_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEOVF IE	—	—	—	AFEIE	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W

Table 38.48 LR0_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	SEOVFIE	Controls error reports when overflow error occurs. 0: Overflow error report disabled 1: Overflow error report enabled
6 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AFEIE	Controls error reports when address feedback error is detected. 0: Address feedback error report disabled 1: Address feedback error report enabled
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

38.3.5.9 LR0_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in LR0_SERSTR. This is a write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SST CLR07	SST CLR06	SST CLR05	SST CLR04	SST CLR03	SST CLR02	SST CLR01	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 38.49 LR0_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	SSTCLR07	Writing 1 to this bit clears SEDF07 in LR0_SERSTR.
6	SSTCLR06	Writing 1 to this bit clears SEDF06 in LR0_SERSTR.
5	SSTCLR05	Writing 1 to this bit clears SEDF05 in LR0_SERSTR.
4	SSTCLR04	Writing 1 to this bit clears SEDF04 in LR0_SERSTR.
3	SSTCLR03	Writing 1 to this bit clears SEDF03 in LR0_SERSTR.
2	SSTCLR02	Writing 1 to this bit clears SEDF02 in LR0_SERSTR.
1	SSTCLR01	Writing 1 to this bit clears SEDF01 in LR0_SERSTR.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in LR0_SERSTR.

38.3.5.10 LR0_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in LR0_DERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.50 LR0_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears AFEF00 and DEDF00 in LR0_DERSTR.

38.3.5.11 LR0_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in LR0_OVFSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.51 LR0_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in LR0_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in LR0_OVFSTR.

38.3.5.12 LR0_SERSTR — 1-bit Error Status Register

This register indicates whether an ECC 1-bit error has occurred. The location and address of the error that is detected are stored in LR0_nSEADR register when SEDFn flag is set. The SEDFn flag is set only when a unique ECC 1-bit error, which is different from previous errors stored in 1-bit error address registers, is detected while the SEDFn is “0”. If the newly detected error has the same address with errors already stored, this flag is not set.

This register can be cleared by SSTCLRn in LR0_SSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEDF07	SEDF06	SEDF05	SEDF04	SEDF03	SEDF02	SEDF01	SEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.52 LR0_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	SEDF07	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_07SEADR register.
6	SEDF06	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_06SEADR register.
5	SEDF05	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_05SEADR register.
4	SEDF04	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_04SEADR register.
3	SEDF03	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_03SEADR register.
2	SEDF02	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_02SEADR register.
1	SEDF01	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_01SEADR register.
0	SEDF00	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_00SEADR register.

38.3.5.13 LR0_DERSTR — Fatal Error Status Register

This register indicates whether an address feedback error and/or an ECC 2-bit error has occurred. The location and address of the error that is detected are stored in LR0_00DEADR register when AFEF00 and/or DEDF00 flag are set. The AFEF00 and/or DEDF00 flag is set only when an address feedback error and/or an ECC 2-bit error is detected while all the flags are “0”. Multiple flags are set only when multiple errors occur due to one error cause.

This register can be cleared by DSTCLR00 in LR0_DSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEF00	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.53 LR0_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AFEF00	Indicates that an address feedback error was detected.
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDF00	Indicates that an ECC 2-bit error was detected.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.5.14 LR0_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address, is detected when the address buffer is fully used. If the newly detected error has the same address with errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flags can be cleared by SERROVFCLR0 and SERROVFCLR1 in LR0_OVFCLR register respectively.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.54 LR0_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVF1	Indicates that more ECC 1-bit errors than the working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error was detected when all ECC 1bit error flags in LR0_SERSTR are set.

38.3.5.15 LR0_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in LR0_SERSTR register were detected.

This register is updated whenever LR0_SERSTR register is updated.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SEDLIN F04	SEDLIN F03	SEDLIN F02	SEDLIN F01	SEDLIN F00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.55 LR0_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	SEDLINF04	Indicates that an ECC 1-bit error was detected in bank 3 of Local RAM.
3	SEDLINF03	Indicates that an ECC 1-bit error was detected in bank 2 of Local RAM.
2	SEDLINF02	Indicates that an ECC 1-bit error was detected in bank 1 of Local RAM.
1	SEDLINF01	Indicates that an ECC 1-bit error was detected in bank 0 of Local RAM.
0	SEDLINF00	Indicates that an ECC 1-bit error was detected in read data loaded from its own Local RAM speculatively.

38.3.5.16 LR0_nSEADR — n-th 1-bit Error Address Register (n = 00 to 07)

This register is used to hold the address and the location of the error when the SEDFn flag is not set and an ECC 1-bit error is detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	SEADR0[14:2]												—	—	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.56 LR0_nSEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 15	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
14 to 2	SEADR0[14:2]	Indicates at which address this error was detected.
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Please refer to the following table to confirm the detailed information which each SEDL indicates.

Table 38.57 SEDL information

SEDL[4:0]	Indicated information
31 to 5	Reserved
4	Indicates an ECC 1-bit error is detected in storing data to or fetching instructions from bank 3 of Local RAM.
3	Indicates an ECC 1-bit error is detected in storing data to or fetching instructions from bank 2 of Local RAM.
2	Indicates an ECC 1-bit error is detected in storing data to or fetching instructions from bank 1 of Local RAM.
1	Indicates an ECC 1-bit error is detected in storing data to or fetching instructions from bank 0 of Local RAM.
0	Indicates an ECC 1-bit error is detected in loading data from its own Local RAM.

38.3.5.17 LR0_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when neither AFEF00 nor DEDF00 flag is set and an address feedback error and/or ECC 2-bit error was detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	DEADR0[14:2]												—	—	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.58 LR0_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 15	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
14 to 2	DEADR0[14:2]	Indicates at which address this error was detected.
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Please refer to the following table to confirm the detailed information which each DEDL indicates.

Table 38.59 DEDL Information

DEDL[4:0]	Indicated information
31 to 5	Reserved
4	Indicates a fatal error is detected in storing data to or fetching instructions from bank 3 of Local RAM.
3	Indicates a fatal error is detected in storing data to or fetching instructions from bank 2 of Local RAM.
2	Indicates a fatal error is detected in storing data to or fetching instructions from bank 1 of Local RAM.
1	Indicates a fatal error is detected in storing data to or fetching instructions from bank 0 of Local RAM.
0	Indicates a fatal error is detected in loading data from its own Local RAM.

38.3.5.18 LR1_ERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM when an ECC error and/or overflow error occurs.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0087_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEOVF IE	—	—	—	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W

Table 38.60 LR1_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	SEOVFIE	Controls error reports when overflow error occurs. 0: Overflow error report disabled 1: Overflow error report enabled
6 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

38.3.5.19 LR1_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in LR1_SERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SST CLR07	SST CLR06	SST CLR05	SST CLR04	SST CLR03	SST CLR02	SST CLR01	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 38.61 LR1_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	SSTCLR07	Writing 1 to this bit clears SEDF07 in LR1_SERSTR.
6	SSTCLR06	Writing 1 to this bit clears SEDF06 in LR1_SERSTR.
5	SSTCLR05	Writing 1 to this bit clears SEDF05 in LR1_SERSTR.
4	SSTCLR04	Writing 1 to this bit clears SEDF04 in LR1_SERSTR.
3	SSTCLR03	Writing 1 to this bit clears SEDF03 in LR1_SERSTR.
2	SSTCLR02	Writing 1 to this bit clears SEDF02 in LR1_SERSTR.
1	SSTCLR01	Writing 1 to this bit clears SEDF01 in LR1_SERSTR.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in LR1_SERSTR.

38.3.5.20 LR1_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in LR1_DERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.62 LR1_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears DEDF00 in LR1_DERSTR.

38.3.5.21 LR1_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in LR1_OVFSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.63 LR1_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in LR1_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in LR1_OVFSTR.

38.3.5.22 LR1_SERSTR — 1-bit Error Status Register

This register indicates whether ECC 1-bit error has occurred. The location and address of the error that is detected are stored in LR1_nSEADR register when SEDFn flag is set. The SEDFn flag is set only when a unique ECC 1-bit error, which is different from previous errors stored in 1-bit error address registers, is detected while the SEDFn is “0”. If the newly detected error has the same address with errors already stored, this flag is not set.

This register can be cleared by SSTCLRn in LR1_SSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEDF07	SEDF06	SEDF05	SEDF04	SEDF03	SEDF02	SEDF01	SEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.64 LR1_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	SEDF07	Indicates that an ECC 1-bit error was detected and error address is stored in LR1_07SEADR register.
6	SEDF06	Indicates that an ECC 1-bit error was detected and error address is stored in LR1_06SEADR register.
5	SEDF05	Indicates that an ECC 1-bit error was detected and error address is stored in LR1_05SEADR register.
4	SEDF04	Indicates that an ECC 1-bit error was detected and error address is stored in LR1_04SEADR register.
3	SEDF03	Indicates that an ECC 1-bit error was detected and error address is stored in LR1_03SEADR register.
2	SEDF02	Indicates that an ECC 1-bit error was detected and error address is stored in LR1_02SEADR register.
1	SEDF01	Indicates that an ECC 1-bit error was detected and error address is stored in LR1_01SEADR register.
0	SEDF00	Indicates that an ECC 1-bit error r was detected and error address is stored in LR1_00SEADR register.

38.3.5.23 LR1_DERSTR — Fatal Error Status Register

This register indicates whether an ECC 2-bit error has occurred. The location and address of the error that is detected are stored in LR1_00DEADR register when DEDF00 flag is set. The DEDF00 flag is set only when an ECC 2-bit error is detected while the DEDF00 is “0”.

This register can be cleared by DSTCLR00 in LR1_DSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.65 LR1_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDF00	Indicates that an ECC 2-bit error was detected.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.5.24 LR1_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address, is detected when the address buffer is fully used. If the newly detected error has the same address with errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flags can be cleared by SERROVFCLR0 and SERROVFCLR1 in LR1_OVFCLR register respectively.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.66 LR1_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVF1	Indicates that more ECC 1-bit errors than the working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error was detected when all ECC 1bit error flags in LR1_SERSTR are set.

38.3.5.25 LR1_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in LR1_SERSTR register were detected.

This register is updated whenever LR1_SERSTR register is updated.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDLIN F01	SEDLIN F00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.67 LR1_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SEDLINF01	Indicates that an ECC 1-bit error was detected in read data to PE1 loaded from the other PE's Local RAM speculatively.
0	SEDLINF00	Indicates that an ECC 1-bit error was detected in read data to PE0 loaded from the other PE's Local RAM speculatively.

38.3.5.26 LR1_nSEADR — n-th 1-bit Error Address Register (n = 00 to 07)

This register is used to hold the address and the location of the error when SEDFn flag is not set and an ECC 1-bit error is detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]					—	—	—	—	—	SEADR0[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SEADR0[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.68 LR1_nSEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 22	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
21 to 2	SEADR0[21:2]	Indicates at which address this error was detected.
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Please refer to the following table to confirm the detailed information which each SEDL indicates.

Table 38.69 SEDL information

SEDL[4:0]	Indicated information
31 to 2	Reserved
1	Indicates an ECC 1-bit error is detected in loading data from the other PE's Local RAM by PE1 master.
0	Indicates an ECC 1-bit error is detected in loading data from the other PE's Local RAM by PE0 master.

38.3.5.27 LR1_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when DEDF00 flag is not set and an ECC 2-bit error is detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]					—	—	—	—	—	DEADR0[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEADR0[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.70 LR1_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 22	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
21 to 2	DEADR0[21:2]	Indicates at which address this error was detected.
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Please refer to the following table to confirm the detailed information which each DEDL indicates.

Table 38.71 DEDL Information

DEDL[4:0]	Indicated information
31 to 2	Reserved
1	Indicates a fatal error is detected in loading data from the other PE's Local RAM by PE1 master.
0	Indicates a fatal error is detected in loading data from the other PE's Local RAM by PE0 master.

38.3.5.28 Test Function

Through appropriate register settings, desired values can be written as RAM data and to the ECC bits. Also, data in the RAM and the ECC bits can all be read.

(1) Writing RAM data

- (a) Set the ECCTST bit in the ECC Test Control Register to 1 to set test mode.
- (b) Set the corresponding DATSEL bit in the ECC Test Control Register to 0 to select RAM data for access when writing.
- (c) When data is written to the Local RAM, only the RAM data can be modified without updating the ECC bits.

How to exit this test mode:

- (a) Set the ECCTST bit in the ECC Test Control Register to 0 to disable test mode (normal mode).

(2) Reading RAM data

- (a) Set the ECCDIS bit in the ECC Control Register to 1 to disable ECC error detection and correction.
- (b) Read the Local RAM. Since neither error detection nor correction proceeds when the Local RAM is read, the RAM data is read unchanged.

How to exit this test mode:

- (a) Set the ECCDIS bit in the ECC Control Register to 0 to enable ECC error detection and correction.

(3) Writing to the ECC bits

- (a) Set the ECCTST bit in the ECC Test Control Register to 1 to set test mode.
- (b) Set the corresponding DATSEL bit in the ECC Test Control Register to 1 to select the ECC bits for access when writing.
- (c) When data is written to the Local RAM, only the ECC bits can be modified without updating the RAM data. At that time, bits[6:0] are written to the ECC bits.

How to exit this test mode:

- (a) Set the ECCTST bit in the ECC Test Control Register to 0 to disable test mode (normal mode).

(4) Reading the ECC bits

- (a) Set the ECCTST bit in the ECC Test Control Register to 1 to set test mode.
- (b) When the Local RAM is read, the ECC bits are stored in the word corresponding to local RAM ECC read buffer register.

How to exit this test mode:

- (a) Set the ECCTST bit in the ECC Test Control Register to 0 to disable test mode (normal mode).

(5) Self-diagnosis of the ECC check function

Desired values can be written to the RAM data, ECC bits by the procedure described in (1) or (3) above. Therefore, a fault can be injected by, for example, inverting appropriate bits of the RAM data and/or ECC bits. After that, self-diagnosis of the ECC decoder is possible by reading the Local RAM in normal mode and checking the result of error correction or detection.

(6) Self-diagnosis of the address feedback check function

Self-diagnosis is enabled by following procedure below.

- (a) Set the desired bit(s) of AFINV[20:4] in the Address Feedback Test Control Register to 1 to set test mode.
- (b) When the Local RAM is read or written, address feedback error occurs because feedback address XOR-ed with AFINV can be input to address feedback checker.

How to exit this test mode:

- (a) Set the AFINV[20:4] bits in the Address Feedback Test Control Register all to 0 to disable test mode (normal mode).

38.3.6 Cluster RAM ECC and Address Feedback

38.3.6.1 Overview

The Cluster RAM ECC is summarized in the table below.

The ECC protection described below takes into consideration all possible Cluster RAM accesses from PE, H-Bus, System Bus and read access for read-modify-write processing unit of 8-bit and 16-bit data.

Table 38.72 Cluster RAM ECC

Item	Description
ECC error detection and correction	<p>ECC error detection and correction can be either enabled or disabled.</p> <p>When enabled, either of the following settings can be selected.</p> <ul style="list-style-type: none"> • ECC error detection and correction are carried out (2-bit error detection, and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>When disabled, neither error detection nor correction is carried out.</p> <p>In the initial state, the ECC function is enabled; 1-bit error detection and correction, and 2-bit error detection are carried out.</p>
Address feedback	<p>Address feedback check can be either enabled or disabled.</p> <p>When enabled, Address feedback error detection is carried out.</p> <p>When disabled, no error detection is carried out.</p> <p>In the initial state, this function is enabled.</p>
Error notification	<p>The occurrence of an ECC error or an address feedback error is reported to the Error Control Module.</p> <p>ECC Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error.</p> <p>Address Feedback Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an address feedback error. <p>In the initial state, error notification is enabled upon detection of an address feedback error.</p> <p>Overflow Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an address buffer overflow error for ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an address buffer overflow error. The error notification signal is output with an address feedback error handled as one source. An ECC 2-bit error, an ECC 1-bit error and an overflow error are handled as individual sources in ECM. An ECC 1-bit error signal is only issued to the ECM if the ECC 1-bit error address is not yet stored in the error address buffer.</p>
Error status	<p>A status register is provided that indicates the statuses of ECC 2-bit error detection, ECC 1-bit error detection, and address feedback error detection. If an error occurs while no error status is set, the corresponding status is set.</p> <p>The error status can be cleared using the clear register.</p>
Address capture	<p>The address is captured when an ECC 2-bit error, an ECC 1-bit error, or an address feedback error is detected. The error status serves as the enable bit of the address that is captured. Address buffers are updated if the error status is cleared.</p> <p>Multi-stage address buffers are provided for a 1-bit data error.</p> <ul style="list-style-type: none"> • ECC 1-bit error: Eight stages • ECC 2-bit error and address feedback error shared: One stage
Self-diagnosis	<p>Desired values can be written to the RAM data and the ECC bits.</p> <p>The RAM data can be read directly and the ECC bits can be read via Read Buffer Register in ECC Test Mode.</p> <p>Moreover, the error injection to an address feedback checker is possible by setting AFINV.</p>

ECC of Cluster RAM is provided for each 32 bits of data and each 32-bit data is called bank 0 to 3. The relationship between addresses and bank numbers are as follows.

Table 38.73 Relationship between addresses and bank numbers

4 lower-order bits of address (Hexadecimal Notation)	F _H to C _H	B _H to 8 _H	7 _H to 4 _H	3 _H to 0 _H
Bank number	Bank 3	Bank 2	Bank 1	Bank 0

38.3.6.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offsets from the base addresses.

Table 38.74 Register Base Addresses

Base Address Name	Base Address	Bus Group
<ECCCNT_CRCCL0_base>	FFC4 9C00 _H	Peripheral Group 0
<ECCCNT_CRCCL1_base>	FFC4 9C80 _H	Peripheral Group 0
<ECCCNT_CRA_base>	FFC4 9E00 _H	Peripheral Group 0
<MECCCAP_CRAM_base>	FFC5 2000 _H	Peripheral Group 0

38.3.6.3 List of Registers

Table 38.75 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
ECCCNT_CRCCLn (n = 0, 1)	Address Feedback Control Register	CRCAFCTL	<ECCCNT_CRCCLn_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	CRCECCTSTCTL	<ECCCNT_CRCCLn_base> + 10 _H	8, 16, 32	KCPROT
	Address Feedback Test Control Register	CRCAFINV	<ECCCNT_CRCCLn_base> + 20 _H	8, 16, 32	KCPROT
	ECC Read Buffer Register	CRCTDATBFECFF	<ECCCNT_CRCCLn_base> + 60 _H	8, 16, 32	
ECCCNT_CRA	ECC Control Register	CRAECCCTL	<ECCCNT_CRA_base> + 00 _H	8, 16, 32	KCPROT
MECCCAP_CRAM	Error Notification Control Register	CR_ERRINT	<MECCCAP_CRAM_base> + 00 _H	8, 16, 32	KCPROT
	1-bit Error Status Clear Register	CR_SSTCLR	<MECCCAP_CRAM_base> + 10 _H	8, 16, 32	
	Fatal Error Status Clear Register	CR_DSTCLR	<MECCCAP_CRAM_base> + 14 _H	8, 16, 32	
	1-bit Error Overflow Status Clear Register	CR_OVFCLR	<MECCCAP_CRAM_base> + 18 _H	8, 16, 32	
	1-bit Error Status Register	CR_SERSTR	<MECCCAP_CRAM_base> + 20 _H	8, 16, 32	
	Fatal Error Status Register	CR_DERSTR	<MECCCAP_CRAM_base> + 24 _H	8, 16, 32	
	1-bit Error Overflow Status Register	CR_OVFSTR	<MECCCAP_CRAM_base> + 28 _H	8, 16, 32	
	1-bit Error Location Information Register	CR_SERINF	<MECCCAP_CRAM_base> + 30 _H	32	
	1st 1-bit Error Address Register	CR_00SEADR	<MECCCAP_CRAM_base> + 70 _H	32	
	2nd 1-bit Error Address Register	CR_01SEADR	<MECCCAP_CRAM_base> + 74 _H	32	
	3rd 1-bit Error Address Register	CR_02SEADR	<MECCCAP_CRAM_base> + 78 _H	32	
	4th 1-bit Error Address Register	CR_03SEADR	<MECCCAP_CRAM_base> + 7C _H	32	
	5th 1-bit Error Address Register	CR_04SEADR	<MECCCAP_CRAM_base> + 80 _H	32	
	6th 1-bit Error Address Register	CR_05SEADR	<MECCCAP_CRAM_base> + 84 _H	32	
	7th 1-bit Error Address Register	CR_06SEADR	<MECCCAP_CRAM_base> + 88 _H	32	
	8th 1-bit Error Address Register	CR_07SEADR	<MECCCAP_CRAM_base> + 8C _H	32	
	1st Fatal Error Address Register	CR_00DEADR	<MECCCAP_CRAM_base> + F0 _H	32	

38.3.6.4 CRCAFCTL — Address Feedback Control Register

This register controls address feedback error detection for the address that is requested to Cluster RAM.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0004_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEDIS	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R

Table 38.76 CRCAFCTL Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AFEDIS	Address feedback error disable bit Sets address feedback error detection to enable/disable. 0: Address feedback error detection is enabled 1: Address feedback error detection is disabled
2 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.6.5 CRCECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), either data field or ECC field of Cluster RAM can be written separately.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	DAT SEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.77 CRCECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects the write mode to Cluster RAM. 0: Normal mode 1: Test mode Either data field or ECC field can be written according to DATSEL setting in test mode.
0	DATSEL	Selects field to write data when ECCTST = 1. 0: Data field only 1: ECC field only

38.3.6.6 CRCAFINV — Address Feedback Test Control Register

This register is used to inject errors into the feedback address from Cluster RAM for self-diagnosis. Feedback address XOR-ed with AFINV can be input to address feedback checker.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	AFINV[18:16]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AFINV[15:4]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Table 38.78 CRCAFINV Register Contents

Bit Position	Bit Name	Function
31 to 19	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
18 to 4	AFINV[18:4]	Specifies bit pattern to inject errors into feedback address.
3 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.6.7 CRCTDATBFECF — ECC Read Buffer Register

This register is used to hold ECC field read from Cluster RAM for self-diagnosis. ECC field in the address that is requested can be stored in this register while reading from Cluster RAM where ECCTST = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	BFEC3[6:0]						—	BFEC2[6:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BFEC1[6:0]						—	BFEC0[6:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.79 CRCTDATBFECF Register Contents

Bit Position	Bit Name	Function
31	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
30 to 24	BFEC3[6:0]	ECC field read from requested address in Cluster RAM bank 3
23	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	BFEC2[6:0]	ECC field read from requested address in Cluster RAM bank 2
15	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	BFEC1[6:0]	ECC field read from requested address in Cluster RAM bank 1
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	BFEC0[6:0]	ECC field read from requested address in Cluster RAM bank 0

38.3.6.8 CRAECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for data read from Cluster RAM.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.80 CRAECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

38.3.6.9 CR_ERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address feedback error, ECC error and/or overflow error occurs.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 008F_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEOVF IE	—	—	—	AFEIE	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W

Table 38.81 CR_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	SEOVFIE	Controls error reports when overflow error has occurred. 0: Overflow error report disabled 1: Overflow error report enabled
6 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AFEIE	Controls error reports when address feedback error is detected. 0: Address feedback error report disabled 1: Address feedback error report enabled
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

38.3.6.10 CR_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in CR_SERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SST CLR07	SST CLR06	SST CLR05	SST CLR04	SST CLR03	SST CLR02	SST CLR01	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 38.82 CR_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	SSTCLR07	Writing 1 to this bit clears SEDF07 in CR_SERSTR.
6	SSTCLR06	Writing 1 to this bit clears SEDF06 in CR_SERSTR.
5	SSTCLR05	Writing 1 to this bit clears SEDF05 in CR_SERSTR.
4	SSTCLR04	Writing 1 to this bit clears SEDF04 in CR_SERSTR.
3	SSTCLR03	Writing 1 to this bit clears SEDF03 in CR_SERSTR.
2	SSTCLR02	Writing 1 to this bit clears SEDF02 in CR_SERSTR.
1	SSTCLR01	Writing 1 to this bit clears SEDF01 in CR_SERSTR.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in CR_SERSTR.

38.3.6.11 CR_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in CR_DERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.83 CR_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears AFEF00 and DEDF00 in CR_DERSTR.

38.3.6.12 CR_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in CR_OVFSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.84 CR_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in CR_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in CR_OVFSTR.

38.3.6.13 CR_SERSTR — 1-bit Error Status Register

This register indicates whether ECC 1-bit error has occurred. The location and address of the error that is detected are stored in CR_nSEADR register when SEDFn flag is set. The SEDFn flag is set only when a unique ECC 1-bit error, which is different from previous errors stored in 1-bit error address registers, is detected while the SEDFn is “0”. If the newly detected error has the same address with errors already stored, this flag is not set.

This register can be cleared by SSTCLRn in CR_SSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEDF07	SEDF06	SEDF05	SEDF04	SEDF03	SEDF02	SEDF01	SEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.85 CR_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	SEDF07	Indicates that an ECC 1-bit error was detected and error address is stored in CR_07SEADR register.
6	SEDF06	Indicates that an ECC 1-bit error was detected and error address is stored in CR_06SEADR register.
5	SEDF05	Indicates that an ECC 1-bit error was detected and error address is stored in CR_05SEADR register.
4	SEDF04	Indicates that an ECC 1-bit error was detected and error address is stored in CR_04SEADR register.
3	SEDF03	Indicates that an ECC 1-bit error was detected and error address is stored in CR_03SEADR register.
2	SEDF02	Indicates that an ECC 1-bit error was detected and error address is stored in CR_02SEADR register.
1	SEDF01	Indicates that an ECC 1-bit error was detected and error address is stored in CR_01SEADR register.
0	SEDF00	Indicates that an ECC 1-bit error was detected and error address is stored in CR_00SEADR register.

38.3.6.14 CR_DERSTR — Fatal Error Status Register

This register indicates whether an address feedback error and/or an ECC 2-bit error has occurred. The location and address of the error that is detected are stored in CR_00DEADR register when AFEF00 and/or DEDF00 flags are set. The AFEF00 and/or DEDF00 flag is set only when an address feedback error and/or an ECC 2-bit error is detected while all the flags are “0”. Multiple flags are set only when multiple errors occur due to one error cause.

This register can be cleared by DSTCLR00 in CR_DSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEF00	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.86 CR_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AFEF00	Indicates that an address feedback error was detected.
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDF00	Indicates that an ECC 2-bit error was detected.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.6.15 CR_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address is detected when the address buffer is fully used. If the newly detected error has the same address as errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flags can be cleared by SERROVFCLR0 and SERROVFCLR1 in CR_OVFCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.87 CR_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVF1	Indicates that more ECC 1-bit errors than working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error was detected when all ECC 1bit error flags in CR_SERSTR are set.

38.3.6.16 CR_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in CR_SERSTR register were detected.

This register is updated whenever CR_SERSTR register is updated.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	SEDLIN F19	SEDLIN F18	SEDLIN F17	SEDLIN F16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SEDLIN F11	SEDLIN F10	SEDLIN F09	SEDLIN F08	—	—	—	—	—	—	SEDLIN F01	SEDLIN F00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.88 CR_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 20	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
19	SEDLINF19	Indicates that an ECC 1-bit error was detected in upper address of Cluster RAM for cluster 1 during the RMW operation.
18	SEDLINF18	Indicates that an ECC 1-bit error was detected in lower address of Cluster RAM for cluster 1 during the RMW operation.
17	SEDLINF17	Indicates that an ECC 1-bit error was detected in upper address of Cluster RAM for cluster 0 during the RMW operation.
16	SEDLINF16	Indicates that an ECC 1-bit error was detected in lower address of Cluster RAM for cluster 0 during the RMW operation.
15 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11	SEDLINF11	Indicates that an ECC 1-bit error was detected in the data read from upper address of Cluster RAM to SAXI2MBI for cluster 1.
10	SEDLINF10	Indicates that an ECC 1-bit error was detected in the data read from lower address of Cluster RAM to SAXI2MBI for cluster 1.
9	SEDLINF09	Indicates that an ECC 1-bit error was detected in the data read from upper address of Cluster RAM to SAXI2MBI for cluster 0.
8	SEDLINF08	Indicates that an ECC 1-bit error was detected in the data read from lower address of Cluster RAM to SAXI2MBI for cluster 0.
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SEDLINF01	Indicates that an ECC 1-bit error was detected in the data read from Cluster RAM to PE1.
0	SEDLINF00	Indicates that an ECC 1-bit error was detected in the data read from Cluster RAM to PE0.

38.3.6.17 CR_nSEADR — n-th 1-bit Error Address Register (n = 00 to 07)

This register is used to hold the address and the location of the error when SEDFn flag is not set and an ECC 1-bit error is detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]					—	—	—	—	—	—	—	SEADR0[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SEADR0[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.89 CR_nSEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 20	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
19 to 2	SEADR0[19:2]	Indicates at which address this error was detected.
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Please refer to the following table to confirm the detailed information which each SEDL indicates.

Table 38.90 SEDL information

SEDL[4:0]	Indicated information
31 to 20	Reserved
19	Indicates an ECC 1-bit error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 1: Storing sub-word data by any master. CAXI operation by any PE master.
18	Indicates an ECC 1-bit error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 1: Storing sub-word data by any master. CAXI operation by any PE master.
17	Indicates an ECC 1-bit error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 0: Storing sub-word data by any master. CAXI operation by any PE master.
16	Indicates an ECC 1-bit error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 0: Storing sub-word data by any master. CAXI operation by any PE master.
15 to 12	Reserved
11	Indicates an ECC 1-bit error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 1: Loading data by any master except PEs. Fetching instructions by any PE master.
10	Indicates an ECC 1-bit error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 1: Loading data by any master except PEs. Fetching instructions by any PE master.
9	Indicates an ECC 1-bit error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 0: Loading data by any master except PEs. Fetching instructions by any PE master.
8	Indicates an ECC 1-bit error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 0: Loading data by any master except PEs. Fetching instructions by any PE master.
7 to 2	Reserved
1	Indicates an ECC 1-bit error is detected in loading data from Cluster RAMs by PE1 master.
0	Indicates an ECC 1-bit error is detected in loading data from Cluster RAMs by PE0 master.

38.3.6.18 CR_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when neither AFEF00 nor DEDF00 flag is set and an address feedback error and/or ECC 2-bit error was detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]				—	—	—	—	—	—	—	DEADR0[19:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEADR0[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.91 CR_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 20	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
19 to 2	DEADR0[19:2]	Indicates at which address this error was detected.
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Please refer to the following table to confirm the detailed information which each DEDL indicates.

Table 38.92 DEDL Information

DEDL[4:0]	Indicated information
31 to 20	Reserved
19	Indicates a fatal error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 1: Storing sub-word data by any master. CAXI operation by any PE master.
18	Indicates a fatal error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 1: Storing sub-word data by any master. CAXI operation by any PE master.
17	Indicates a fatal error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 0: Storing sub-word data by any master. CAXI operation by any PE master.
16	Indicates a fatal error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 0: Storing sub-word data by any master. CAXI operation by any PE master.
15 to 12	Reserved
11	Indicates a fatal error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 1: Loading data by any master except PEs. Fetching instructions by any PE master.
10	Indicates a fatal error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 1: Loading data by any master except PEs. Fetching instructions by any PE master.
9	Indicates a fatal error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 0: Loading data by any master except PEs. Fetching instructions by any PE master.
8	Indicates a fatal error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 0: Loading data by any master except PEs. Fetching instructions by any PE master.
7 to 2	Reserved
1	Indicates a fatal error is detected in loading data from Cluster RAMs by PE1 master.
0	Indicates a fatal error is detected in loading data from Cluster RAMs by PE0 master.

38.3.6.19 Test Function

(1) Writing RAM data

- (a) Set the ECCTST bit in the ECC Test Control Register to 1 to set test mode.
- (b) Set the corresponding DATSEL bit in the ECC Test Control Register to 0 to select RAM data for access when writing.
- (c) When data is written to the cluster RAM, the RAM data is modified without updating the ECC bits.

How to exit this test mode:

- (a) Set the ECCTST bit in the ECC Test Control Register to 0 to disable test mode (normal mode).

(2) Reading RAM data

- (a) Set the ECCDIS bit in the ECC Control Register to 1 to disable ECC error detection and correction.
- (b) Read the Cluster RAM. Since neither error detection nor correction proceeds when the Cluster RAM is read, the RAM data is read unchanged.

How to exit this test mode:

- (a) Set the ECCDIS bit in the ECC Control Register to 0 to enable ECC error detection and correction.

(3) Writing to the ECC bits

- (a) Set the ECCTST bit in the ECC Test Control Register to 1 to set test mode.
- (b) Set the corresponding DATSEL bit in the ECC Test Control Register to 1 to select the ECC bits for access when writing.
- (c) When data is written to the Cluster RAM, the ECC bits are modified without updating the RAM data. At that time, bits[6:0] are respectively written to the ECC bits.

How to exit this test mode:

- (a) Set the ECCTST bit in the ECC Test Control Register to 0 to disable test mode (normal mode).

(4) Reading the ECC bits

- (a) Set the ECCTST bit in the ECC Test Control Register to 1 to set test mode.
- (b) When the Cluster RAM is read, the ECC bits are stored in the word corresponding to cluster RAM ECC read buffer register.

How to exit this test mode:

- (a) Set the ECCTST bit in the ECC Test Control Register to 0 to disable test mode (normal mode).

(5) Self-diagnosis of the ECC check function

Desired values can be written to the RAM data, ECC bits by using the procedure described in (1) or (3) above. Therefore, a fault can be injected by, for example, inverting appropriate bits of the RAM data and/or ECC bits. After that, self-diagnosis of the ECC decoder is possible by reading the Cluster RAM in normal mode and checking the result of error correction or detection.

(6) Self-diagnosis of the address feedback check function

Self-diagnosis is enabled by following procedure below.

- (a) Set the desired bit(s) of AFINV in the Address Feedback Test Control Register to 1 to set test mode.
- (b) When the Cluster RAM is read or written, address feedback error occurs because feedback address XOR-ed with AFINV can be input to address feedback checker.

How to exit this test mode:

- (a) Set the AFINV bits in the Address Feedback Test Control Register all to 0 to disable test mode (normal mode).

38.3.7 Instruction Cache EDC and Address Feedback

38.3.7.1 Overview

The instruction cache EDC is summarized in the table below.

Table 38.93 Instruction Cache EDC

Item	Description
ECC error detection	<p>ECC error detection can be either enabled or disabled.</p> <p>When enabled, the following settings can be selected:</p> <ul style="list-style-type: none"> • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>When disabled, error detection is not carried out.</p> <p>In the initial state, the ECC function is enabled; 1-bit error detection, and 2-bit error detection are carried out.</p>
Address feedback	<p>Address feedback check can be either enabled or disabled.</p> <p>When enabled, Address feedback error detection is carried out.</p> <p>When disabled, no error detection is carried out.</p> <p>In the initial state, this function is enabled.</p>
Error notification	<p>The occurrence of an ECC error or an address feedback error is reported to the Error Control Module.</p> <p>ECC Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. <p>In the initial state, notification of both an ECC 2-bit error and an ECC 1-bit error is disabled.</p> <p>Address Feedback Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an address feedback error. <p>In the initial state, error notification is disabled upon detection of an address feedback error.</p> <p>The error notification signal is output with an ECC 2-bit error and an ECC 1-bit error handled as one source. An address feedback error is handled as individual source.</p> <p>An ECC error signal is only issued to the ECM if the ECC error address is not yet stored in the error address buffer.</p>
Error status	<p>A status register is provided that indicates the statuses of ECC 2-bit error detection, ECC 1-bit error detection and address feedback error detection. If an error occurs while no error status is set, the corresponding status is set.</p> <p>The error status can be cleared using the clear register.</p>
Address capture	<p>The address is captured when an ECC 2-bit error, an ECC 1-bit error or an address feedback error is detected. The error status serves as the enable bit of the captured address.</p> <p>If an ECC error occurs while no error status is set, the address at which the associated error occurred is captured.</p>
Self-diagnosis	<p>A cache instruction is used to write the desired values as RAM data and to the ECC bits.</p> <p>Similarly, data in the RAM and the ECC bits can be read directly.</p> <p>Since cache instructions go through the same encoding or decoding path as a normal cache fill or instruction fetch, errors can be both inserted and confirmed simply by using a cache instruction.</p> <p>Errors can also be injected into an address feedback checker by setting AFINV.</p>

38.3.7.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as an offset from the base addresses.

Table 38.94 Register Base Addresses

Base Address Name	Base Address	Bus Group
<ECCCNT_IT_PE0CL0_base>	FFC4 9000 _H	Peripheral Group 0
<ECCCNT_IT_PE1CL0_base>	FFC4 9080 _H	Peripheral Group 0
<ECCCNT_ID_PE0CL0_base>	FFC4 9400 _H	Peripheral Group 0
<ECCCNT_ID_PE1CL0_base>	FFC4 9480 _H	Peripheral Group 0
<MECCCAP_IT_PE0CL0_base>	FFC5 1000 _H	Peripheral Group 0
<MECCCAP_IT_PE1CL0_base>	FFC5 1100 _H	Peripheral Group 0
<MECCCAP_ID_PE0CL0_base>	FFC5 1800 _H	Peripheral Group 0
<MECCCAP_ID_PE1CL0_base>	FFC5 1900 _H	Peripheral Group 0

38.3.7.3 List of Registers

Table 38.95 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
ECCCNT_IT_PEnCL0 (n = 0, 1)	ECC Control Register	ITECCCTL	<ECCCNT_IT_PEnCL0_base> + 00 _H	8, 16, 32	KCPROT
	Address Feedback Test Control Register	ITAFINV	<ECCCNT_IT_PEnCL0_base> + 20 _H	8, 16, 32	KCPROT
ECCCNT_ID_PEnCL0 (n = 0, 1)	ECC Control Register	IDECCCTL	<ECCCNT_ID_PEnCL0_base> + 00 _H	8, 16, 32	KCPROT
	Address Feedback Test Control Register	IDAFINV	<ECCCNT_ID_PEnCL0_base> + 20 _H	8, 16, 32	KCPROT
MECCCAP_IT_PEnCL0 (n = 0, 1)	Error Notification Control Register	IT_ERRINT	<MECCCAP_IT_PEnCL0_base> + 00 _H	8, 16, 32	KCPROT
	1-bit Error Status Clear Register	IT_SSTCLR	<MECCCAP_IT_PEnCL0_base> + 10 _H	8, 16, 32	
	Fatal Error Status Clear Register	IT_DSTCLR	<MECCCAP_IT_PEnCL0_base> + 14 _H	8, 16, 32	
	1-bit Error Overflow Status Clear Register	IT_OVFCLR	<MECCCAP_IT_PEnCL0_base> + 18 _H	8, 16, 32	
	1-bit Error Status Register	IT_SERSTR	<MECCCAP_IT_PEnCL0_base> + 20 _H	8, 16, 32	
	Fatal Error Status Register	IT_DERSTR	<MECCCAP_IT_PEnCL0_base> + 24 _H	8, 16, 32	
	1-bit Error Overflow Status Register	IT_OVFSTR	<MECCCAP_IT_PEnCL0_base> + 28 _H	8, 16, 32	
	1-bit Error Location Information Register	IT_SERINF	<MECCCAP_IT_PEnCL0_base> + 30 _H	32	
	1st 1-bit Error Address Register	IT_00SEADR	<MECCCAP_IT_PEnCL0_base> + 70 _H	32	
	1st fatal Error Address Register	IT_00DEADR	<MECCCAP_IT_PEnCL0_base> + F0 _H	32	
MECCCAP_ID_PEnCL0 (n = 0, 1)	Error Notification Control Register	ID_ERRINT	<MECCCAP_ID_PEnCL0_base> + 00 _H	8, 16, 32	KCPROT
	1-bit Error Status Clear Register	ID_SSTCLR	<MECCCAP_ID_PEnCL0_base> + 10 _H	8, 16, 32	
	Fatal Error Status Clear Register	ID_DSTCLR	<MECCCAP_ID_PEnCL0_base> + 14 _H	8, 16, 32	
	1-bit Error Overflow Status Clear Register	ID_OVFCLR	<MECCCAP_ID_PEnCL0_base> + 18 _H	8, 16, 32	
	1-bit Error Status Register	ID_SERSTR	<MECCCAP_ID_PEnCL0_base> + 20 _H	8, 16, 32	
	Fatal Error Status Register	ID_DERSTR	<MECCCAP_ID_PEnCL0_base> + 24 _H	8, 16, 32	
	1-bit Error Overflow Status Register	ID_OVFSTR	<MECCCAP_ID_PEnCL0_base> + 28 _H	8, 16, 32	
	1-bit Error Location Information Register	ID_SERINF	<MECCCAP_ID_PEnCL0_base> + 30 _H	32	
	1st 1-bit Error Address Register	ID_00SEADR	<MECCCAP_ID_PEnCL0_base> + 70 _H	32	
	1st Fatal Error Address Register	ID_00DEADR	<MECCCAP_ID_PEnCL0_base> + F0 _H	32	

38.3.7.4 ITECCCTL — ECC Control Register

This register controls ECC error detection for data read from cache tag RAM and address feedback error detection for the address that is requested to cache tag RAM.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFE DIS	—	—	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W

Table 38.96 ITECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AFEDIS	Address feedback error disable bit Sets the address feedback error detection to enable/disable. 0: Address feedback error detection is enabled 1: Address feedback error detection is disabled
2, 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ECCDIS	ECC disable bit Sets the ECC error detection to enable/disable. 0: ECC error detection is enabled 1: ECC error detection is disabled

38.3.7.5 ITAFINV — Address Feedback Test Control Register

This register is used to inject errors into the feedback address from cache tag RAM for self-diagnosis. Feedback address XOR-ed with AFINV can be input to address feedback checker.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	AFINV[11:5]							—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R

Table 38.97 ITAFINV Register Contents

Bit Position	Bit Name	Function
31 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 5	AFINV[11:5]	Specifies bit pattern to inject errors into feedback address.
4 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.7.6 IDECCCTL — ECC Control Register

This register controls ECC error detection for data read from cache data RAM and address feedback error detection for the address that is requested to the cache data RAM.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFE DIS	—	—	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W

Table 38.98 IDECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AFEDIS	Address feedback error disable bit Sets the address feedback error detection to enable/disable. 0: Address feedback error detection is enabled 1: Address feedback error detection is disabled
2, 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ECCDIS	ECC disable bit Sets the ECC error detection to enable/disable. 0: ECC error detection is enabled 1: ECC error detection is disabled

38.3.7.7 IDAFINV — Address Feedback Test Control Register

This register is used to inject errors into the feedback address from the cache data RAM for self-diagnosis. Feedback address XOR-ed with AFINV can be input to address feedback checker.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	AFINV[11:3]									—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Table 38.99 IDAFINV Register Contents

Bit Position	Bit Name	Function
31 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 3	AFINV[11:3]	Specifies bit pattern to inject errors into feedback address.
2 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.7.8 IT_ERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM when an address feedback error, ECC error and/or overflow error occurs.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEIE	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W

Table 38.100 IT_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AFEIE	Controls error reports when address feedback error is detected. 0: Address feedback error report disabled 1: Address feedback error report enabled
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

38.3.7.9 IT_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in IT_SERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.101 IT_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in IT_SERSTR.

38.3.7.10 IT_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in IT_DERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.102 IT_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears AFEF00 and DEDF00 in IT_DERSTR.

38.3.7.11 IT_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in IT_OVFSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.103 IT_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in IT_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in IT_OVFSTR.

38.3.7.12 IT_SERSTR — 1-bit Error Status Register

This register indicates whether ECC 1-bit error has occurred. The location and address of the error that is detected are stored in IT_00SEADR register when SEDF00 flag is set. The SEDF00 flag is set only when an ECC 1-bit error is detected while the SEDF00 is “0”.

This register can be cleared by SSTCLR00 in IT_SSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.104 IT_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SEDF00	Indicates that an ECC 1-bit error was detected.

38.3.7.13 IT_DERSTR — Fatal Error Status Register

This register indicates whether an address feedback error and/or an ECC 2-bit error has occurred. The location and address of the error that is detected are stored in IT_00DEADR register when AFEF00 and/or DEDF00 flags are set. The AFEF00 and/or DEDF00 flag is set only when an address feedback error and/or an ECC 2-bit error is detected while all the flags are “0”. Multiple flags are set only when multiple errors occur due to one error cause.

This register can be cleared by DSTCLR00 in IT_DSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEF00	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.105 IT_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AFEF00	Indicates that an address feedback error was detected.
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDF00	Indicates that an ECC 2-bit error was detected.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.7.14 IT_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address is detected when the address buffer is fully used. If the newly detected error has the same address as errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flags can be cleared by SERROVFCLR0 and SERROVFCLR1 in IT_OVFCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.106 IT_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVF1	Indicates that more ECC 1-bit errors than working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error was detected when SEDF00 was already set.

38.3.7.15 IT_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in IT_SERSTR register were detected.

This is up-to-date whenever IT_SERSTR register is updated.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SEDLIN F03	SEDLIN F02	SEDLIN F01	SEDLIN F00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.107 IT_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	SEDLINF03	Indicates that an ECC 1-bit error was detected in way 3 of cache tag RAM.
2	SEDLINF02	Indicates that an ECC 1-bit error was detected in way 2 of cache tag RAM.
1	SEDLINF01	Indicates that an ECC 1-bit error was detected in way 1 of cache tag RAM.
0	SEDLINF00	Indicates that an ECC 1-bit error was detected in way 0 of cache tag RAM.

38.3.7.16 IT_00SEADR — 1st 1-bit Error Address Register

This register is used to hold the address and the location of the error when SEDF00 flag is not set and an ECC 1-bit error is detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SEADR0[11:5]							—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.108 IT_00SEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 5	SEADR0[11:5]	Indicates at which address this error was detected.
4 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Please refer to the following table to confirm the detailed information which each SEDL indicates.

Table 38.109 SEDL information

SEDL[4:0]	Indicated information
31 to 4	Reserved
3	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 3 of cache tag RAM.
2	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 2 of cache tag RAM.
1	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 1 of cache tag RAM.
0	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 0 of cache tag RAM.

38.3.7.17 IT_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when neither AFEF00 nor DEDF00 flag is set and an address feedback error and/or ECC 2-bit error is detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DEADR0[11:5]							—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.110 IT_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 5	DEADR0[11:5]	Indicates at which address this error was detected.
4 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Please refer to the following table to confirm the detailed information which each DEDL indicates.

Table 38.111 DEDL Information

DEDL[4:0]	Indicated information
31 to 4	Reserved
3	Indicates a fatal error is detected in storing data to or loading data from way 3 of cache tag RAM.
2	Indicates a fatal error is detected in storing data to or loading data from way 2 of cache tag RAM.
1	Indicates a fatal error is detected in storing data to or loading data from way 1 of cache tag RAM.
0	Indicates a fatal error is detected in storing data to or loading data from way 0 of cache tag RAM.

38.3.7.18 ID_ERRINT — Error Notification Control Register

This register controls whether error information is reported to the ECM when address feedback error, ECC error and/or overflow error occurs.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEIE	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W

Table 38.112 ID_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AFEIE	Controls error reports when address feedback error is detected. 0: Address feedback error report disabled 1: Address feedback error report enabled
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

38.3.7.19 ID_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in ID_SERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.113 ID_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in ID_SERSTR.

38.3.7.20 ID_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in ID_DERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.114 ID_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears AFEF00 and DEDF00 in ID_DERSTR.

38.3.7.21 ID_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in ID_OVFSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.115 ID_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in ID_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in ID_OVFSTR.

38.3.7.22 ID_SERSTR — 1-bit Error Status Register

This register indicates whether ECC 1-bit error has occurred. The location and address of the error that is detected are stored in ID_00SEADR register when SEDF00 flag is set. The SEDF00 flag is set only when an ECC 1-bit error is detected while the SEDF00 is “0”.

This register can be cleared by SSTCLR00 in ID_SSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.116 ID_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SEDF00	Indicates an ECC 1-bit error is detected.

38.3.7.23 ID_DERSTR — Fatal Error Status Register

This register indicates whether an address feedback error and/or an ECC 2-bit error has occurred. The location and address of the error that is detected are stored in ID_00DEADR register when AFEF00 and/or DEDF00 flags are set. The AFEF00 and/or DEDF00 flag is set only when an address feedback error and/or an ECC 2-bit error is detected while all the flags are “0”. Multiple flags are set only when multiple errors occur due to one error cause.

This register can be cleared by DSTCLR00 in ID_DSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEF00	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.117 ID_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AFEF00	Indicates that address feedback error was detected.
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDF00	Indicates that an ECC 2-bit error was detected.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.7.24 ID_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address is detected when the address buffer is fully used. If the newly detected error has the same address as errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flags can be cleared by SERROVFCLR0 and SERROVFCLR1 in ID_OVFCLR register respectively.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.118 ID_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVF1	Indicates that more ECC 1-bit errors than working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error was detected when SEDF00 was already set.

38.3.7.25 ID_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in ID_SERSTR register were detected.

This is up-to-date whenever ID_SERSTR register is updated.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SEDLIN F03	SEDLIN F02	SEDLIN F01	SEDLIN F00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.119 ID_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	SEDLINF03	Indicates that an ECC 1-bit error was detected in way 3 of the cache data RAM.
2	SEDLINF02	Indicates that an ECC 1-bit error was detected in way 2 of the cache data RAM.
1	SEDLINF01	Indicates that an ECC 1-bit error was detected in way 1 of the cache data RAM.
0	SEDLINF00	Indicates that an ECC 1-bit error was detected in way 0 of the cache data RAM.

38.3.7.26 ID_00SEADR — 1st 1-bit Error Address Register

This register is used to hold the address and the location of the error when SEDF00 flag is not set and an ECC 1-bit error is detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SEADR0[11:3]							—	—	—		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.120 ID_00SEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 3	SEADR0[11:3]	Indicates at which address this error was detected.
2 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Please refer to the following table to confirm the detailed information which each SEDL indicates.

Table 38.121 SEDL information

SEDL[4:0]	Indicated information
31 to 4	Reserved
3	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 3 of cache data RAM.
2	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 2 of cache data RAM.
1	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 1 of cache data RAM.
0	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 0 of cache data RAM.

38.3.7.27 ID_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when neither AFEF00 nor DEDF00 flag is set and an address feedback error and/or ECC 2-bit error is detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DEADR0[11:3]							—	—	—		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.122 ID_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 3	DEADR0[11:3]	Indicates at which address this error was detected.
2 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Please refer to the following table to confirm the detailed information which each DEDL indicates.

Table 38.123 DEDL Information

DEDL[4:0]	Indicated information
31 to 4	Reserved
3	Indicates a fatal error is detected in storing data to or loading data from way 3 of cache data RAM.
2	Indicates a fatal error is detected in storing data to or loading data from way 2 of cache data RAM.
1	Indicates a fatal error is detected in storing data to or loading data from way 1 of cache data RAM.
0	Indicates a fatal error is detected in storing data to or loading data from way 0 of cache data RAM.

38.3.7.28 Test Function

A cache instruction is used to write the desired values as RAM data and to the ECC bits, and read data in the RAM and the ECC bits directly.

Since cache instructions go through the same encoding or decoding path as a normal cache fill or instruction fetch, errors can be both inserted and confirmed simply by using a cache instruction.

For details, refer to *the RH850G4MH User's Manual: Software*.

38.3.8 sDMAC/DTS RAM ECC and Address Feedback

38.3.8.1 Overview

sDMAC RAM ECC is summarized in the table below.

Table 38.124 sDMAC RAM ECC

Item	Description
ECC error detection and correction	<p>ECC error detection and correction can be either enabled or disabled.</p> <p>When enabled, either of the following settings can be selected:</p> <ul style="list-style-type: none"> • ECC error detection and correction are carried out (2-bit error detection, and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>When disabled, neither error detection nor correction is carried out.</p> <p>In the initial state, the ECC function is enabled; 1-bit error detection and correction, and 2-bit error detection are carried out.</p>
Error notification	<p>The occurrence of an ECC error is reported to the Error Control Module.</p> <p>ECC Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. <p>In the initial state, notification of both a 2-bit error and a 1-bit error is enabled.</p> <p>Overflow Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an address buffer overflow error for ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an address buffer overflow error.</p> <p>An ECC 2-bit error, an ECC 1-bit error and an overflow error are handled as individual sources. An ECC 1-bit error signal is only issued to the ECM if the ECC 1-bit error address is not yet stored in the error address buffer.</p>
Error status	<p>A status register is provided that indicates the statuses of ECC 2-bit error detection and ECC 1-bit error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.</p>
Address capture	<p>The address is captured when an ECC 2-bit error or an ECC 1-bit error is detected. The error status serves as the enable bit of the captured address.</p> <p>ECC 1-bit error: One stage ECC 2-bit error: One stage</p>
Self-diagnosis	<p>Desired values can be written to the RAM data and the ECC bits in sDMAC RAM in ECC Test Mode.</p> <p>The RAM data and the ECC bits in sDMAC RAM can be also read via Read Buffer Registers in this mode.</p>

DTS RAM ECC is summarized in the table below.

Table 38.125 DTS RAM ECC

Item	Description
ECC error detection and correction	<p>ECC error detection and correction can be either enabled or disabled.</p> <p>When enabled, either of the following settings can be selected.</p> <ul style="list-style-type: none"> • ECC error detection and correction are carried out (2-bit error detection, 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>When disabled, neither error detection nor correction is carried out.</p> <p>In the initial state, the ECC function is enabled; 1-bit error detection and correction, and 2-bit error detection are carried out.</p>
Address feedback	<p>Address feedback check can be either enabled or disabled.</p> <p>When enabled, Address feedback error detection is carried out.</p> <p>When disabled, no error detection is carried out.</p> <p>In the initial state, this function is enabled.</p>
Error notification	<p>The occurrence of an ECC error or an address feedback error is reported to the Error Control Module.</p> <p>ECC Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. <p>In the initial state, notification of both a 2-bit error and a 1-bit error is enabled.</p> <p>Address Feedback Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an address feedback error. <p>In the initial state, error notification is enabled upon detection of an address feedback error.</p> <p>Overflow Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an address buffer overflow error for ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an address buffer overflow error.</p> <p>The error notification signal is output with an ECC 2-bit error and an address feedback error handled as one source, and an ECC 1-bit error and an overflow error are handled as individual sources. An ECC 1-bit error signal is only issued to the ECM, if the ECC 1-bit error address is not yet stored in the error address buffer.</p>
Error status	<p>A status register is provided that indicates the statuses of ECC 2-bit error detection, ECC 1-bit error detection, and address feedback error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.</p>
Address capture	<p>The address is captured when an ECC 2-bit error, an ECC 1-bit error, or an address feedback error is detected. The error status serves as the enable bit of the capture address.</p> <p>ECC 1-bit error: One stage</p> <p>ECC 2-bit error and address feedback error: One stage</p>
Self-diagnosis	<p>Desired values can be written to the RAM data and the ECC bits in DTS RAM in ECC Test Mode.</p> <p>The RAM data and the ECC bits in DTS RAM can be also read via Read Buffer Registers in this mode.</p> <p>Moreover, the error injection to an address feedback checker is possible by setting AFINV.</p>

38.3.8.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offset from the base addresses.

Table 38.126 Register Base Addresses

Base Address Name	Base Address	Bus Group
<ECCCNT_DTS_base>	FFC4 A000 _H	Peripheral Group 0
<ECCCNT_DMDE0_base>	FFC4 A400 _H	Peripheral Group 0
<ECCCNT_DMDA0_base>	FFC4 A480 _H	Peripheral Group 0
<ECCCNT_DMDE1_base>	FFC4 A500 _H	Peripheral Group 0
<ECCCNT_DMDA1_base>	FFC4 A580 _H	Peripheral Group 0
<MECCCAP_DTS_base>	FFC5 2600 _H	Peripheral Group 0
<MECCCAP_DMDE0_base>	FFC5 2800 _H	Peripheral Group 0
<MECCCAP_DMDA0_base>	FFC5 2900 _H	Peripheral Group 0
<MECCCAP_DMDE1_base>	FFC5 2A00 _H	Peripheral Group 0
<MECCCAP_DMDA1_base>	FFC5 2B00 _H	Peripheral Group 0

38.3.8.3 List of Registers

Table 38.127 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection
ECCCNT_DTS	ECC Control Register	DRECCCTL	<ECCCNT_DTS_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	DRECCSTCTL	<ECCCNT_DTS_base> + 10 _H	8, 16, 32	KCPROT
	Address Feedback Test Control Register	DRAFINV	<ECCCNT_DTS_base> + 20 _H	8, 16, 32	KCPROT
	Test Data Input Register	DRECCSTDIN0	<ECCCNT_DTS_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	DRECCSTEIN	<ECCCNT_DTS_base> + 40 _H	8, 16, 32	KCPROT
	Data Read Buffer Register	DRTDATABFDATAF0	<ECCCNT_DTS_base> + 50 _H	32	
	ECC Read Buffer Register	DRTDATABFECCF	<ECCCNT_DTS_base> + 60 _H	8, 16, 32	
ECCCNT_DMDEn (n = 0, 1)	ECC Control Register	DEECCCTL	<ECCCNT_DMDEn_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	DEECCSTCTL	<ECCCNT_DMDEn_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	DEECCSTDIN0	<ECCCNT_DMDEn_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	DEECCSTEIN	<ECCCNT_DMDEn_base> + 40 _H	8, 16, 32	KCPROT
	Data Read Buffer Register	DETATABFDATAF0	<ECCCNT_DMDEn_base> + 50 _H	32	
	ECC Read Buffer Register	DETATABFECCF	<ECCCNT_DMDEn_base> + 60 _H	8, 16, 32	
ECCCNT_DMDAn (n = 0, 1)	ECC Control Register	DAECCCTL	<ECCCNT_DMDAn_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	DAECCSTCTL	<ECCCNT_DMDAn_base> + 10 _H	8, 16, 32	KCPROT
	Test Lower Data Input Register	DAECCSTDIN0	<ECCCNT_DMDAn_base> + 30 _H	8, 16, 32	KCPROT
	Test Upper Data Input Register	DAECCSTDIN1	<ECCCNT_DMDAn_base> + 34 _H	8, 16, 32	KCPROT
	Test ECC Input Register	DAECCSTEIN	<ECCCNT_DMDAn_base> + 40 _H	8, 16, 32	KCPROT
	Lower Data Read Buffer Register	DATDATABFDATAF0	<ECCCNT_DMDAn_base> + 50 _H	32	

Table 38.127 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection
ECCCNT_DMDAn (n = 0, 1)	Upper Data Read Buffer Register	DATDATBFDATAF1	<ECCCNT_DMDAn_base> + 54 _H	32	
	ECC Read Buffer Register	DATDATBFECCEF	<ECCCNT_DMDAn_base> + 60 _H	8, 16, 32	
MECCCAP_DTS	Error Notification Control Register	DR_ERRINT	<MECCCAP_DTS_base> + 00 _H	8, 16, 32	KCPROT
	1-bit Error Status Clear Register	DR_SSTCLR	<MECCCAP_DTS_base> + 10 _H	8, 16, 32	
	Fatal Error Status Clear Register	DR_DSTCLR	<MECCCAP_DTS_base> + 14 _H	8, 16, 32	
	1-bit Error Overflow Status Clear Register	DR_OVFCLR	<MECCCAP_DTS_base> + 18 _H	8, 16, 32	
	1-bit Error Status Register	DR_SERSTR	<MECCCAP_DTS_base> + 20 _H	8, 16, 32	
	Fatal Error Status Register	DR_DERSTR	<MECCCAP_DTS_base> + 24 _H	8, 16, 32	
	1-bit Error Overflow Status Register	DR_OVFSTR	<MECCCAP_DTS_base> + 28 _H	8, 16, 32	
	1-bit Error Location Information Register	DR_SERINF	<MECCCAP_DTS_base> + 30 _H	32	
	1st 1-bit Error Address Register	DR_00SEADR	<MECCCAP_DTS_base> + 70 _H	32	
	1st Fatal Error Address Register	DR_00DEADR	<MECCCAP_DTS_base> + F0 _H	32	
MECCCAP_DMDEn (n = 0, 1)	Error Notification Control Register	DE_ERRINT	<MECCCAP_DMDEn_base> + 00 _H	8, 16, 32	KCPROT
	1-bit Error Status Clear Register	DE_SSTCLR	<MECCCAP_DMDEn_base> + 10 _H	8, 16, 32	
	Fatal Error Status Clear Register	DE_DSTCLR	<MECCCAP_DMDEn_base> + 14 _H	8, 16, 32	
	1-bit Error Overflow Status Clear Register	DE_OVFCLR	<MECCCAP_DMDEn_base> + 18 _H	8, 16, 32	
	1-bit Error Status Register	DE_SERSTR	<MECCCAP_DMDEn_base> + 20 _H	8, 16, 32	
	Fatal Error Status Register	DE_DERSTR	<MECCCAP_DMDEn_base> + 24 _H	8, 16, 32	
	1-bit Error Overflow Status Register	DE_OVFSTR	<MECCCAP_DMDEn_base> + 28 _H	8, 16, 32	
	1-bit Error Location Information Register	DE_SERINF	<MECCCAP_DMDEn_base> + 30 _H	32	
	1st 1-bit Error Address Register	DE_00SEADR	<MECCCAP_DMDEn_base> + 70 _H	32	
	1st fatal Error Address Register	DE_00DEADR	<MECCCAP_DMDEn_base> + F0 _H	32	
MECCCAP_DMDAn (n = 0, 1)	Error Notification Control Register	DA_ERRINT	<MECCCAP_DMDAn_base> + 00 _H	8, 16, 32	KCPROT
	1-bit Error Status Clear Register	DA_SSTCLR	<MECCCAP_DMDAn_base> + 10 _H	8, 16, 32	
	Fatal Error Status Clear Register	DA_DSTCLR	<MECCCAP_DMDAn_base> + 14 _H	8, 16, 32	
	1-bit Error Overflow Status Clear Register	DA_OVFCLR	<MECCCAP_DMDAn_base> + 18 _H	8, 16, 32	
	1-bit Error Status Register	DA_SERSTR	<MECCCAP_DMDAn_base> + 20 _H	8, 16, 32	
	Fatal Error Status Register	DA_DERSTR	<MECCCAP_DMDAn_base> + 24 _H	8, 16, 32	
	1-bit Error Overflow Status Register	DA_OVFSTR	<MECCCAP_DMDAn_base> + 28 _H	8, 16, 32	
	1-bit Error Location Information Register	DA_SERINF	<MECCCAP_DMDAn_base> + 30 _H	32	
	1st 1-bit Error Address Register	DA_00SEADR	<MECCCAP_DMDAn_base> + 70 _H	32	
	1st Fatal Error Address Register	DA_00DEADR	<MECCCAP_DMDAn_base> + F0 _H	32	

Note: The initial characters of DTS/sDMAC register symbols indicates the followings.

DR: DTS RAM

DE: Descriptor RAM of sDMAC

DA: Data RAM of sDMAC

38.3.8.4 DRECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for data read from DTS RAM and address feedback error detection for the address that is requested to DTS RAM.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFE DIS	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W

Table 38.128 DRECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AFEDIS	Address feedback error disable bit Sets the address feedback error detection to enable/disable. 0: Address feedback error detection is enabled 1: Address feedback error detection is disabled
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets the ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

38.3.8.5 DRECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), arbitrary data value can be written to data field and ECC field of DTS RAM.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 38.129 DRECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for DTS RAM. 0: Normal mode 1: Test mode The values of DRECCTSTDIN0 and DRECCTSTEIN registers are written to data and ECC fields of DTS RAM when writing in test mode.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.8.6 DRAFINV — Address Feedback Test Control Register

This register is used to inject errors into the feedback address from DTS RAM for self-diagnosis. Feedback address XOR-ed with AFINV can be input to address feedback checker.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	AFINV[11:2]										—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 38.130 DRAFINV Register Contents

Bit Position	Bit Name	Function
31 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 2	AFINV[11:2]	Specifies bit pattern to inject errors into feedback address.
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.8.7 DRECCTSTDIN0 — Test Data Input Register

This register is used to inject errors into data field of DTS RAM for self-diagnosis. The values of DRECCTSTDIN0 and DRECCTSTEIN registers are written to the requested address when write operation to DTS RAM occurs in test mode.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.131 DRECCTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for DTS RAM. The data is provided to data field of DTS RAM when ECCTST = 1.

38.3.8.8 DRECCTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of DTS RAM for self-diagnosis. The values of DRECCTSTDIN0 and DRECCTSTEIN registers are written to the requested address when write operation to DTS RAM occurs in test mode.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.132 DRECCTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECC[6:0]	Specify the test data for DTS RAM. The data is provided to ECC field of DTS RAM when ECCTST = 1.

38.3.8.9 DRTDATBFDATAF0 — Data Read Buffer Register

This register is used to hold data field read from DTS RAM for self-diagnosis. The data field in the requested address can be stored this register while reading from DTS RAM when ECCTST = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BFDATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BFDATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.133 DRTDATBFDATAF0 Register Contents

Bit Position	Bit Name	Function
31 to 0	BFDATA[31:0]	Data field read from requested address in DTS RAM

38.3.8.10 DRTDATBFECF — ECC Read Buffer Register

This register is used to hold ECC field read from DTS RAM for self-diagnosis. ECC field in the requested address can be stored in this register while reading from DTS RAM when ECCTST = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	BFEC0[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.134 DRTDATBFECF Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	BFEC0[6:0]	ECC field read from the requested address in DTS RAM

38.3.8.11 DEECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for data read from Descriptor RAM of sDMAC.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.135 DEECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets the 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets the ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled.

38.3.8.12 DEECTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), arbitrary data value can be written to data field and ECC field for Descriptor RAM of sDMAC.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 38.136 DEECTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for Descriptor RAM of sDMAC. 0: Normal mode 1: Test mode The values of DEECTSTDIN0 and DEECTSTEIN registers are written to data and ECC field for Descriptor RAM of sDMAC when writing is in test mode.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.8.13 DEECTSTDIN0 — Test Data Input Register

This register is used to inject errors into data field of Descriptor RAM of sDMAC for self-diagnosis. The values of DEECTSTDIN0 and DEECTSTEIN registers are written to the requested address when write operation to Descriptor RAM occurs in test mode.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.137 DEECTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify test data for Descriptor RAM of sDMAC. The data is provided to data field of Descriptor RAM when ECCTST = 1.

38.3.8.14 DEECCTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of Descriptor RAM of sDMAC for self-diagnosis. The values of DEECCTSTDIN0 and DEECCTSTEIN registers are written to the requested address when write operation to Descriptor RAM occurs in test mode.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.138 DEECCTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECC[6:0]	Specify the test data for Descriptor RAM of sDMAC. The data is provided to ECC field of Descriptor RAM when ECCTST = 1.

38.3.8.15 DETDATBFDATAF0 — Data Read Buffer Register

This register is used to hold data field read from Descriptor RAM of sDMAC for self-diagnosis. The data field in the requested address can be stored in this register while reading from Descriptor RAM when ECCTST = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BFDATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BFDATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.139 DETDATBFDATAF0 Register Contents

Bit Position	Bit Name	Function
31 to 0	BFDATA[31:0]	Data field read from the requested address in Descriptor RAM

38.3.8.16 DETDATBFECF — ECC Read Buffer Register

This register is used to hold ECC field read from Descriptor RAM of sDMAC for self-diagnosis. ECC field in the address that is requested can be stored in this register while reading from Descriptor RAM where ECCTST = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	BFEC0[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.140 DETDATBFECF Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	BFEC0[6:0]	ECC field read from the address that is requested in Descriptor RAM

38.3.8.17 DAECCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for read data from Data RAM of sDMAC.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.141 DAECCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), set the 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets the ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

38.3.8.18 DAECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), arbitrary data value can be written to data field and ECC field for Data RAM of sDMAC.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 38.142 DAECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for Data RAM of sDMAC. 0: Normal mode 1: Test mode The values of DAECCTSTDIN0, DAECCTSTDIN1 and DAECCTSTEIN registers are written to data and ECC field for Data RAM of sDMAC when buffering transfer data in test mode.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.8.19 DAECCTSTDIN0 — Test Lower Data Input Register

This register is used to inject errors into lower data field of Data RAM of sDMAC for self-diagnosis. The values of DAECCTSTDIN0, DAECCTSTDIN1 and DAECCTSTEIN registers are written to the requested address when write operation to Data RAM occurs in test mode.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.143 DAECCTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify test data for Data RAM of sDMAC. The data is provided to lower data field of Data RAM when ECCTST = 1.

38.3.8.20 DAECCTSTDIN1 — Test Upper Data Input Register

This register is used to inject errors into upper data field of Data RAM of sDMAC for self-diagnosis. The values of DAECCTSTDIN0, DAECCTSTDIN1 and DAECCTSTEIN registers are written to the requested address when write operation to Data RAM occurs in test mode.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.144 DAECCTSTDIN1 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for Data RAM of sDMAC. The data is provided to upper data field of Data RAM when ECCTST = 1.

38.3.8.21 DAECCTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of Data RAM of sDMAC for self-diagnosis. The values of DAECCTSTDIN0, DAECCTSTDIN1 and DAECCTSTEIN registers are written to the requested address when write operation to Data RAM occurs in test mode.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ECC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.145 DAECCTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	ECC[7:0]	Specify the test data for Data RAM of sDMAC. The data is provided to ECC field of Data RAM when ECCTST = 1.

38.3.8.22 DATDATBFDATAF0 — Lower Data Read Buffer Register

This register is used to hold data field read from Data RAM of sDMAC for self-diagnosis. The data field in the requested address can be stored in this register while reading from Data RAM when ECCTST = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BFDATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BFDATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.146 DATDATBFDATAF0 Register Contents

Bit Position	Bit Name	Function
31 to 0	BFDATA[31:0]	Lower data field read from the requested address in Data RAM

38.3.8.23 DATDATBFDATAF1 — Upper Data Read Buffer Register

This register is used to hold data field read from Data RAM of sDMAC for self-diagnosis. The data field in the requested address can be stored in this register while reading from Data RAM when ECCTST = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BFDATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BFDATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.147 DATDATBFDATAF1 Register Contents

Bit Position	Bit Name	Function
31 to 0	BFDATA[31:0]	Upper data field read from the requested address in Data RAM

38.3.8.24 DATDATBFECF — ECC Read Buffer Register

This register is used to hold ECC field read from Data RAM of sDMAC for self-diagnosis. ECC field in the requested address can be stored in this register while reading from Data RAM when ECCTST = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	BFEC0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.148 DATDATBFECF Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	BFEC0[7:0]	ECC field read from the requested address in Data RAM

38.3.8.25 DR_ERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address feedback error, ECC error and/or overflow error occurs.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 008B_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEOVF IE	—	—	—	AFEIE	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W

Table 38.149 DR_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	SEOVFIE	Controls error reports when overflow error occurs. 0: Overflow error report disabled 1: Overflow error report enabled
6 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AFEIE	Controls error reports when address feedback error is detected. 0: Address feedback error report disabled 1: Address feedback error report enabled
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

38.3.8.26 DR_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in DR_SERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.150 DR_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in DR_SERSTR.

38.3.8.27 DR_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in DR_DERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.151 DR_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears AFEF00 and DEDF00 in DR_DERSTR.

38.3.8.28 DR_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in DR_OVFSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.152 DR_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in DR_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in DR_OVFSTR.

38.3.8.29 DR_SERSTR — 1-bit Error Status Register

This register indicates whether ECC 1-bit error has occurred. The location and address of the error that is detected are stored in DR_00SEADR register when SEDF00 flag is set. The SEDF00 flag is set only when an ECC 1-bit error is detected while the SEDF00 is “0”.

This register can be cleared by SSTCLR00 in DR_SSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.153 DR_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SEDF00	Indicates that ECC 1-bit error was detected.

38.3.8.30 DR_DERSTR — Fatal Error Status Register

This register indicates whether an address feedback error and/or an ECC 2-bit error has occurred. The location and address of the error that is detected are stored in DR_00DEADR register when AFEF00 and/or DEDF00 flags are set. The AFEF00 and/or DEDF00 flag is set only when an address feedback error and/or an ECC 2-bit error is detected while all the flags are “0”. Multiple flags are set only when multiple errors occur due to one error cause.

This register can be cleared by DSTCLR00 in DR_DSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEF00	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.154 DR_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	AFEF00	Indicates that address feedback error was detected.
2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDF00	Indicates that ECC 2-bit error was detected.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.8.31 DR_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address is detected when the address buffer is fully used. If the newly detected error has the same address as errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flag can be cleared by SERROVFCLR0 and SERROVFCLR1 in DR_OVFCLR register respectively.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.155 DR_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVF1	Indicates that more ECC 1-bit errors than working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC error was detected when SEDF00 was already set.

38.3.8.32 DR_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in DR_SERSTR register were detected.

This register is updated whenever DR_SERSTR register is updated.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDLIN F00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.156 DR_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SEDLINF00	Indicates that ECC error was detected in DTS RAM.

38.3.8.33 DR_00SEADR — 1st 1-bit Error Address Register

This register is used to hold the address and the location of the error when SEDF00 flag is not set and an ECC 1-bit error is detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SEADR0[11:2]										—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.157 DR_00SEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 2	SEADR0[11:2]	Indicates at which address this error was detected.
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 38.158 SEDL Information

Values	Function
31 to 1	These values are reserved.
0	Indicates an ECC 1bit error is detected in storing data to or loading data from DTS RAM.

38.3.8.34 DR_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when neither AFEF00 nor DEDF00 flag is set and an address feedback error and/or ECC 2-bit error is detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DEADR0[11:2]										—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.159 DR_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 2	DEADR0[11:2]	Indicates at which address this error was detected.
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Please refer to the following table to confirm the detailed information which each DEDL indicates.

Table 38.160 DEDL Information

Values	Function
31 to 1	These values are reserved.
0	Indicates a fatal error is detected in storing data to or loading data from DTS RAM.

38.3.8.35 DE_ERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when ECC error and/or overflow error occurs.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0083_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEOVF IE	—	—	—	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W

Table 38.161 DE_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	SEOVFIE	Controls error reports when overflow error occurs. 0: Overflow error report disabled 1: Overflow error report enabled
6 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

38.3.8.36 DE_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in DE_SERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.162 DE_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in DE_SERSTR.

38.3.8.37 DE_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in DE_DERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.163 DE_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears DEDF00 in DE_DERSTR.

38.3.8.38 DE_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in DE_OVFSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.164 DE_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in DE_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in DE_OVFSTR.

38.3.8.39 DE_SERSTR — 1-bit Error Status Register

This register indicates whether ECC 1-bit error has occurred. The location and address of the error that is detected are stored in DE_00SEADR register when SEDF00 flag is set. The SEDF00 flag is set only when an ECC 1-bit error is detected while the SEDF00 is “0”.

This register can be cleared by SSTCLR00 in DE_SSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.165 DE_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SEDF00	Indicates that ECC 1-bit error was detected.

38.3.8.40 DE_DERSTR — Fatal Error Status Register

This register indicates whether ECC 2-bit error has occurred. The location and the address of the error that is detected are stored in DE_00DEADR register when DEDF00 flag is set. The DEDF00 flag is set only when an ECC 2-bit error is detected while the DEDF00 is “0”.

This register can be cleared by DSTCLR00 in DE_DSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.166 DE_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDF00	Indicates that ECC 2-bit error was detected.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.8.41 DE_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address is detected when the address buffer is fully used. If the newly detected error has the same address as errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flags can be cleared by SERROVFCLR0 and SERROVFCLR1 in DE_OVFCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.167 DE_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVF1	Indicates that more ECC 1-bit errors than working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error was detected when SEDF00 was already set.

38.3.8.42 DE_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in DE_SERSTR register were detected.

This register is updated whenever DE_SERSTR register is updated.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDLIN F00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.168 DE_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SEDLINF00	Indicates that ECC 1-bit error was detected in Descriptor RAM of sDMAC.

38.3.8.43 DE_00SEADR — 1st 1-bit Error Address Register

This register is used to hold the address and the location of the error when SEDF00 flag is not set and an ECC 1-bit error is detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SEADR0[11:2]										—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.169 DE_00SEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 2	SEADR0[11:2]	Indicates at which address this error was detected.
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 38.170 SEDL Information

Values	Function
31 to 1	These values are reserved.
0	Indicates an ECC 1bit error is detected in storing data to or loading data from Descriptor RAM of sDMAC.

38.3.8.44 DE_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when DEDF00 flag is not set and an ECC 2-bit error is detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DEADR0[11:2]										—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.171 DE_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11 to 2	DEADR0[11:2]	Indicates at which address this error was detected.
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Please refer to the following table to confirm the detailed information which each DEDL indicates.

Table 38.172 DEDL Information

Values	Function
31 to 1	These values are reserved.
0	Indicates a fatal error is detected in storing data to or loading data from Descriptor RAM of sDMAC.

38.3.8.45 DA_ERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when ECC error and/or overflow error occurs.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only

when KCPROT.KCE = 1.

Value after reset: 0000 0083_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEOVF IE	—	—	—	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W

Table 38.173 DA_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	SEOVFIE	Controls error reports when overflow error occurs. 0: Overflow error report disabled 1: Overflow error report enabled
6 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

38.3.8.46 DA_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in DA_SERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.174 DA_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in DA_SERSTR.

38.3.8.47 DA_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in DA_DERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.175 DA_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears DEDF00 in DA_DERSTR.

38.3.8.48 DA_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in DA_OVFSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.176 DA_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in DA_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in DA_OVFSTR.

38.3.8.49 DA_SERSTR — 1-bit Error Status Register

This register indicates whether ECC 1-bit error has occurred. The location and address of the error that is detected are stored in DA_00SEADR register when SEDF00 flag is set. The SEDF00 flag is set only when an ECC 1-bit error is detected while the SEDF00 is “0”.

This register can be cleared by SSTCLR00 in DA_SSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.177 DA_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SEDF00	Indicates that ECC 1-bit error was detected.

38.3.8.50 DA_DERSTR — Fatal Error Status Register

This register indicates whether ECC 2-bit error has occurred. The location and address of the error that is detected are stored in DA_00DEADR register when DEDF00 flag is set. The DEDF00 flag is set only when an ECC 2-bit error is detected while the DEDF00 is “0”.

This register can be cleared by DSTCLR00 in DA_DSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.178 DA_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDF00	Indicates that ECC 2-bit error was detected.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.8.51 DA_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address is detected when the address buffer is fully used. If the newly detected error has the same address as errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flag can be cleared by SERROVFCLR0 and SERROVFCLR1 in DA_OVFCLR register respectively.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.179 DA_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SERROVF1	Indicates that more ECC 1-bit errors than the working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error was detected when SEDF00 was already set.

38.3.8.52 DA_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in DA_SERSTR register were detected.

This register is updated whenever DA_SERSTR register is updated.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDLIN F00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.180 DA_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	SEDLINF00	Indicates that ECC 1-bit error was detected in Data RAM of sDMAC.

38.3.8.53 DA_00SEADR — 1st 1-bit Error Address Register

This register is used to hold the address and the location of the error when SEDF00 flag is not set and an ECC 1-bit error is detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SEADR0[10:3]							—	—	—	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.181 DA_00SEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 11	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
10 to 3	SEADR0[10:3]	Indicates at which address this error was detected.
2 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 38.182 SEDL Information

Values	Function
31 to 1	These values are reserved.
0	Indicates an ECC 1bit error is detected in storing data to or loading data from Data RAM of sDMAC.

38.3.8.54 DA_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when DEDF00 flag is not set and an ECC 2-bit error is detected.

This register can only be cleared by reset.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DEADR0[10:3]							—	—	—	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.183 DA_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 11	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
10 to 3	DEADR0[10:3]	Indicates at which address this error was detected.
2 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Please refer to the following table to confirm the detailed information which each DEDL indicates.

Table 38.184 DEDL Information

Values	Function
31 to 1	These values are reserved.
0	Indicates a fatal error is detected in storing data to or loading data from Data RAM of sDMAC.

38.3.8.55 Test Function

(1) DTS RAM ECC

The ECC of the DTS RAM can be tested by using the following procedure:

1. Make sure that the RAM address to be tested is not modified by any ongoing DTS transfer. It is recommended to disable DTS entirely.
2. Write data to any DTS RAM address.
3. Write 0x02 to the DTS RAM ECC Test Control Register (DRECCTSTCTL). Enable ECC Test Mode and encode ECC from write data.
4. Read data from the same DTS RAM address.
5. Check Data and ECC field by using the Data Read Buffer Register (DRTDATBFDATAF0) and ECC Read Buffer Register (DRTDATBFECCECF).
6. Restore DTS RAM data and ECC with 1-bit or 2-bit errors by using the Test Data Input Register (DRECCTSTDIN0) or the Test ECC Input Register (DRECCTSTEIN).
7. Write data to the same DTS RAM address.
8. Write 0x00 to the DTS RAM ECC Test Control Register (DRECCTSTCTL). Then, Read data from the same DTS RAM address and check if ECC error occurs.

(2) sDMAC Descriptor RAM ECC

The ECC of the sDMAC Descriptor RAM can be tested by using the following procedure:

1. Make sure that the RAM address to be tested is not modified by any ongoing sDMAC transfer. It is recommended to disable sDMAC entirely.
2. Write data to any sDMAC Descriptor RAM address.
3. Write 0x02 to the sDMAC Descriptor RAM ECC Test Control Register (DEECCTSTCTL). Enable ECC Test Mode and encode ECC from write data.
4. Read data from the same sDMAC Descriptor RAM address.
5. Check Data and ECC field by using the Data Read Buffer Register (DETDATBFDATAF0) and ECC Read Buffer Register (DETDATBFECCECF).
6. Restore sDMAC Descriptor RAM data and ECC with 1-bit or 2-bit errors by using the Test Data Input Register (DEECCTSTDIN0) or the Test ECC Input Register (DEECCTSTEIN).
7. Write data to the same sDMAC Descriptor RAM address.
8. Write 0x00 to the sDMAC Descriptor RAM ECC Test Control Register (DEECCTSTCTL). Then, Read data from the same sDMAC Descriptor RAM address and check if ECC error occurs.

(3) sDMAC Data RAM ECC

The ECC of the sDMAC Data RAM can be tested by using ECC calculation of ERRGEN. For details, see **Section 38.3.12.8, Usage of ERRGEN**.

38.3.9 ECC for Peripheral RAM (32 Bits)

38.3.9.1 Overview

This is an ECC module for the RAM of the following peripheral modules:

RS-CAN-FD, FlexRay, Ethernet, GTM and DFE

(1) RS-CAN-FD, FlexRay, GTM and DFE

Error Detection and Correction

Seven-bit ECC data is appended to the 32-bit RAM data.

This ECC module provides 2-bit ECC error detection and 1-bit ECC error detection and correction.

Table 38.185 RS-CANFD, FlexRay, GTM and DFE RAM ECC

Item	Description
ECC error detection and correction	<p>ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected.</p> <ul style="list-style-type: none"> • ECC error detection and correction are carried out (2-bit error detection, and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>When disabled, neither error detection nor correction is carried out.</p> <p>In the initial state, this function is enabled; 2-bit error detection, and 1-bit error detection and correction are carried out.</p>
Error notification	<p>Upon occurrence of an ECC error, it is notified to the Error Control Module.</p> <p>ECC Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is disabled upon detection of an ECC 1-bit error.</p> <p>An ECC 2-bit error and an ECC 1-bit error are handled as individual source in ECM. All overflow errors of each peripheral module (RS-CANFD, FlexRay, GTM and DFE) are notified to ECM as one factor.</p> <p>An ECC error signal is only issued to the ECM, if the ECC error address is not yet stored in the error address buffer.</p>
Error status	<p>A status register is provided, which indicates the statuses of ECC 2-bit error detection and ECC 1-bit error detection. A one-stage buffer is provided for 2-bit error and 1-bit error.</p> <p>The error status can be cleared using the clear register.</p>
Address capture	<p>The address is captured when an ECC 2-bit error or an ECC 1-bit error is detected.</p> <p>The error status serves as the enable bit of the capture address. Address buffer is updated if the error status is cleared.</p> <p>One-stage address buffer is provided for an ECC 2-bit error and an ECC 1-bit error.</p>
Self-diagnosis	<ul style="list-style-type: none"> • By setting the test mode, register values can be used as the data to be output to the RAM. When a peripheral module writes to the RAM, the E710TED.ECEDB[31:0] bit value can be written to the RAM data section, and the E710TRC.ECERDB[6:0] bit value can be written to the ECC redundant bit section. • By setting the test mode, the ECC redundant bit section can be latched when RAM data is read, and the value can be confirmed. • By setting the test mode, the ECC redundant bit (encoding circuit) and syndrome code (decoding circuit), which are generated from the input data, can be confirmed.

(2) Ethernet

Error Detection and Correction

Seven-bit ECC data is appended to the 32-bit RAM data.

This ECC module provides 2-bit ECC error detection and 1-bit ECC error detection and correction.

Table 38.186 Ethernet RAM ECC

Item	Description
ECC error detection and correction	<p>ECC error detection can not be disabled. ECC error correction can be either enabled or disabled. Either of the following settings can be selected.</p> <ul style="list-style-type: none"> • ECC error detection and correction are carried out (2-bit error detection, and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>In the initial state, this function is enabled; 2-bit error detection, and 1-bit error detection and correction are carried out.</p>
Error notification	<p>Upon occurrence of an ECC error, it is notified to the Error Control Module.</p> <p>ECC Error:</p> <ul style="list-style-type: none"> • Error notification can not be disabled upon detection of an ECC 2-bit error. • Error notification can not be disabled upon detection of an ECC 1-bit error.
Error Status	None
Address Capture	None
Self-diagnosis	None

38.3.9.2 List of Registers

(1) List of ECC Modules

The RAMs of the multiple peripheral functions are provided with the ECC modules. The following table shows the peripheral functions provided with the ECC modules, the corresponding ECC module names, and base addresses of the ECC modules.

Table 38.187 List of ECC Modules

Peripheral Functions		ECC Module Names and Register Base Address	
		Module Name	Base Address <base_addr>
RS-CAN-FD0 (Peripheral Group3)	Message buffer RAM (MB RAM)	E7RC01	FFC7 0000 _H
	Acceptance filter list RAM (AFL RAM)	E7RC02	FFC7 0020 _H
FlexRay0 (Peripheral Group9)	Message RAM (MRAM)	E7FR00	FF05 0000 _H
	Temporary buffer (TBF A)	E7FR01	FF05 0020 _H
	Temporary buffer (TBF B)	E7FR02	FF05 0040 _H
GTM (Peripheral Group2H)	MCS0_RAM0	E7GT00	FFDE 4000 _H
	MCS0_RAM1	E7GT01	FFDE 4100 _H
	MCS1_RAM0	E7GT10	FFDE 4200 _H
	MCS1_RAM1	E7GT11	FFDE 4300 _H
	MCS2_RAM0	E7GT20	FFDE 4400 _H
	MCS2_RAM1	E7GT21	FFDE 4500 _H
	MCS3_RAM0	E7GT30	FFDE 4600 _H
	MCS3_RAM1	E7GT31	FFDE 4700 _H
	MCS4_RAM0	E7GT40	FFDE 4800 _H
	MCS4_RAM1	E7GT41	FFDE 4900 _H
	MCS5_RAM0	E7GT50	FFDE 4A00 _H
	MCS5_RAM1	E7GT51	FFDE 4B00 _H
	MCS6_RAM0	E7GT60	FFDE 4C00 _H
	MCS6_RAM1	E7GT61	FFDE 4D00 _H
	FIFO0_RAM	E7GT70	FFDE 5800 _H
	FIFO1_RAM	E7GT71	FFDE 5900 _H
	DPLL_RAM1A	E7GT80	FFDE 5A00 _H
	DPLL_RAM1B (include PLL_RAM1C)	E7GT81	FFDE 5B00 _H
	DPLL_RAM2	E7GT82	FFDE 5C00 _H
DFE0 (Peripheral Group5)	Coefficient memory (CMEM)	E7DF00	FFC7 8000 _H
DFE1 (Peripheral Group5)	Coefficient memory (CMEM)	E7DF10	FFC7 8060 _H
Ethernet (Peripheral Group9)	Transmit/Receive buffer	EETC	FF06 0000 _H

(2) List of Registers

Each ECC module has the registers shown in the following table.

Table 38.188 List of Registers

Register Name	Symbol	Address	Access	Access Protection
ECC Control Register	E710CTL *1	<base_addr> + 00 _H	8, 16	
ECC Test Mode Control Register	E710TMC *1	<base_addr> + 04 _H	8, 16	
ECC Redundant Bit Data Control Test Register	E710TRC *1	<base_addr> + 08 _H	32	
ECC Encoder and Decoder Data Test Register	E710TED *1	<base_addr> + 0C _H	32	
ECC Error Address Register	E710EAD *1	<base_addr> + 10 _H	32	
Ethernet TX RAM ECC Control Register	ETCTXECCCTL	FF06 0000 _H	8, 16, 32	
Ethernet RX RAM ECC Control Register	ETCRXECCCTL	FF06 0004 _H	8, 16, 32	

Note 1. This function is not implemented in Ethernet.

(3) Register Map

Table 38.189 Register Map

Symbol	31	24	23	16	15	8	7	0	Address
E710CTL	— (00 _H)		— (00 _H)		ECCTL[15:8]		ECCTL[7:0]		00 _H
E710TMC	— (00 _H)		— (00 _H)		ECTMC[15:8]		ECTMC[7:0]		04 _H
E710TRC	ECSYND[7:0]		ECHORD[7:0]		ECECRD[7:0]		ECERDB[7:0]		08 _H
E710TED	ECEDB[31:24]		ECEDB[23:16]		ECEDB[15:8]		ECEDB[7:0]		0C _H
E710EAD	ECEAD[31:24]		ECEAD[23:16]		ECEAD[15:8]		ECEAD[7:0]		10 _H

38.3.9.3 E710CTL — ECC Control Register

E710CTL controls the status and modes of the ECC modules.

E710CTL can be read and written to using the 16-bit or 8-bit manipulation instruction.

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA[1:0]	—	—	—	ECOVFF	ECER2C	ECER1C	—	ECTHM	—	EC1ECP	EC2EDC	EC1EDC	ECER2F	ECER1F	ECEMF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	Undefined
	R/W	R/W*1	R/W*1	R	R	R	R/W*1	R/W*1	R	R/W	R	R/W	R/W	R/W	R	R

Note 1. This bit is always read as 0.

Table 38.190 E710CTL Register Contents (1/2)

Bit Position	Bit Name	Function
15, 14	EMCA[1:0]	Sets Access Control 1 and 0 to ECC Mode Select Bit These bits enable or disable writing to bit 7. The read value is always 0.
13, 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11	ECOVFF	Error Overflow Detection Flag This flag is set when an error of any source other than the same error address occurs while the error flag (ECER2F or ECER1F) is set. One of the following clears this flag: <ul style="list-style-type: none"> • An internal or external reset • When ECC function is disabled (ECTHM = 1) • Writing 1 to ECER2C and ECER1C
10	ECER2C	2-Bit ECC Error Detection Flag Clear Clears the bit 2 (ECER2F) status flag. The read value is always 0, and writing 0 to ECER2C does not change the state. If writing 1 to ECER2C conflicts with the condition for setting bit 2, the former takes priority. Writing 1 to ECER2C while ECER2F is set clears ECER2F.
9	ECER1C	1-Bit ECC Error Detection Flag Clear Clears the bit 1 (ECER1F) status flag. The read value is always 0, and writing 0 to ECER1C does not change the internal state. If writing 1 to ECER1C conflicts with the condition for setting bit 1, the former takes priority. Writing 1 to ECER1C while ECER1F is set clears ECER1F.
8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	ECTHM	ECC Function Disable Select Selects the ECC decoding operation. Write access to ECTHM is enabled when the value of bits 15 and 14 is 01 _B . Therefore, only the 16-bit manipulation instruction is valid. Setting ECTHM to 1 disables error detection and bit correction. Here, if the data to be output to the peripheral module contains an error, the data is output without bit correction. Setting ECTHM to 1 has no effect on the encoder side. <ul style="list-style-type: none"> 0: Enable ECC detection and correction. 1: Disable ECC detection and correction.
6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Table 38.190 E710CTL Register Contents (2/2)

Bit Position	Bit Name	Function
5	EC1ECP	<p>1-Bit ECC Error Correction Enable</p> <p>0: Enable 1-bit error correction upon error detection. 1: Disable 1-bit error correction upon error detection.</p>
4	EC2EDIC	<p>2-Bit ECC Error Detection Notification Enable</p> <p>0: When a 2-bit error is detected, the ECM is not notified of the error. 1: When a 2-bit error is detected, the ECM is notified of the error.</p>
3	EC1EDIC	<p>1-Bit ECC Error Detection Notification Enable</p> <p>0: When a 1-bit error is detected, the ECM is not notified of the error. 1: When a 1-bit error is detected, the ECM is notified of the error.</p>
2	ECER2F	<p>2-Bit ECC Error Detection Flag</p> <p>Indicates that errors have been detected at two bits in bits 0 to 38 of the data read from the RAM during RAM read access while error detection is enabled. This bit is a read only flag.</p> <p>0: A 2-bit error has not occurred since this bit was cleared. 1: A 2-bit error has occurred.</p> <p>Clearing conditions</p> <p>(1) Reset is applied. (2) 1 is written to ECER2C. (3) ECC function is disabled (ECTHM = 1)</p>
1	ECER1F	<p>1-Bit ECC Error Detection and Correction Flag</p> <p>Indicates that an error has been detected at one bit in bits 0 to 38 of the data read from the RAM during RAM read access while error detection is enabled. Setting this bit generates no interrupt signal. This bit is a read only flag.</p> <p>0: A 1-bit error has not occurred. 1: A 1-bit error has occurred.</p> <p>Clearing conditions</p> <p>(1) Reset is applied. (2) 1 is written to ECER1C. (3) ECC function is disabled (ECTHM = 1)</p>
0	ECEMF	<p>ECC Error Message Flag</p> <p>Indicates that the current read data contains an error. ECEMF is updated every time RAM data is read. Since the RAM value after reset is undefined, if this bit is read before the RAM is initialized, this bit might be set, indicating an error. Therefore, the ECEMF value after reset is undefined.</p> <p>0: The current RAM data contains no bit errors. 1: The current RAM data contains bit errors.</p> <p>Clearing conditions</p> <p>(1) ECC function is disabled (ECTHM = 1) (2) Decoding circuit input data contains no 1-bit errors.</p> <p>ECEMF remains set as long as RAM data containing a bit error is output with error detection being enabled.</p>

38.3.9.4 E710TMC — ECC Test Mode Control Register

E710TMC is a 16-bit register used to switch the mode to test mode and control the mode.

E710TMC can be read and written to using the 16-bit or 8-bit manipulation instruction.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA[1:0]		—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*	R/W*	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Note 1. This bit is always read as 0.

Table 38.191 E710TMC Register Contents (1/2)

Bit Position	Bit Name	Function
15, 14	ETMA[1:0]	Access Control 1 and 0 to ECC Test Mode Control Enable Bit These bits enable or disable writing to bit 7. The read value is always 0.
13 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	ECTMCE	ECC Test Mode Control Enable ECTMCE enables or disables access to the test registers and test control bits. Write access to ECTMCE is enabled when the value of bits 15 and 14 is 10 _B . 0: Disable access to the test registers and test control bits. 1: Enable access to the test registers and test control bits.
6, 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	ECTRRS	ECC RAM Read Test Mode Control Enable ECTRRS enables the read status of RAM to be generated by reading the E710TED register, and also allows the RAM output data to be read out when the E710TRC:ECERDB[7:0] bits and the E710TED register are read. Write access to ECTRRS is enabled only when ECTMCE is 1 (can be set simultaneously). ECTRRS is cleared by writing 0 to ECTMCE (can be cleared synchronously). 0: Disable generation of RAM read status for testing when E710TED is read. 1: Enable generation of RAM read status for testing when E710TED is read. When E710TRC:ECERDB[7:0] and E710TED are read, the values of the RAM output data pin are read out.
3	ECREOS	ECC Redundant Bit Output Data Select ECREOS selects either the ECC encoder output data or the ECERDB register value to be output as the ECC redundant bit output. Write access to ECREOS is enabled only when ECTMCE is 1 (can be set simultaneously). ECREOS is cleared by writing 0 to ECTMCE (can be cleared synchronously). 0: Allow encoding result to be output as the ECC redundant bit output. 1: Allow the E710TRC:ECERDB[6:0] value to be output as the ECC redundant bit output.
2	ECENS	ECC Encoder Input Select ECENS selects either the data value from the peripheral module or the internal test register value (E710TED:ECEDB[31:0]) as the input signal to be encoded. Write access to ECENS is enabled only when ECTMCE is 1 (can be set simultaneously). ECENS is cleared by writing 0 to ECTMCE (can be cleared synchronously). 0: Allow the RAM write data from the peripheral module to be input as the ECC encoder input data. 1: Allow the E710TED:ECEDB[31:0] value to be input as the ECC encoder input data.

Table 38.191 E710TMC Register Contents (2/2)

Bit Position	Bit Name	Function
1	ECDCS	<p>ECC Decoder Input Select</p> <p>ECDCS selects either the lower 32-bit data value from the RAM or the internal test register value (E710TED:ECEDB[31:0]) as the lower 32-bit data of the input signal to be decoded. Write access to ECDCS is enabled only when ECTMCE is 1 (can be set simultaneously). ECDCS is cleared by writing 0 to ECTMCE (can be cleared synchronously).</p> <p>0: Allow the lower 32-bit RAM output data to be input to the data area (32 lower-order bits) to the decoder.</p> <p>1: Allow the E710TED:ECEDB[31:0] value to be input to the data area to the decoder.</p>
0	ECREIS	<p>ECC Redundant Bit Input Data Select</p> <p>ECREIS selects either the upper 7-bit data value from the RAM (redundant bit area) or the internal test register value (E710TRC:ECERDB[6:0]) as the upper 7-bit data of the input signal to be decoded. Write access to ECREIS is enabled only when ECTMCE is 1 (can be set simultaneously). ECREIS is cleared by writing 0 to ECTMCE (can be cleared synchronously).</p> <p>0: Allow the upper 7-bit RAM output data to be input to the ECC redundant bit area to the decoder.</p> <p>1: Allow the E710TRC:ECERDB[6:0] value to be input to the ECC redundant bit area to the decoder.</p>

38.3.9.5 E710TED — ECC Encoder and Decoder Data Test Register

E710TED is a 32-bit data test register for ECC encoding and decoding.

When ECTMCE = 1, E710TED can be read and written to using the 32-bit manipulation instruction.

When ECTMCE = 0, E710TED is always read as 0.

In test mode, the E710TED value can be used as the data input to the encoding circuit and decoding circuit.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECEDB[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECEDB[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NOTE

Changing ECTMCE from 1 to 0 resets E710TED synchronously.

When E710TMC:ECENS = 1, the ECEDB value is input to the encoding circuit and supplied to the RAM.

When E710TMC:ECDCS = 1, the ECEDB value is input as bits 31 to 0 of the data input to the decoding circuit.

When E710TMC:ECTRRS = 1, reading ECEDB returns the RAM output data instead of the data written to ECEDB.

38.3.9.6 E710TRC — ECC Redundant Bit Data Control Test Register

E710TRC is a 32-bit test register consisting of four fields (ECSYND, ECHORD, ECECRD, and ECRODM) corresponding to the ECC redundant bit area. Each field can be accessed as the 8-bit register with the same name. For details of each field, refer to the descriptions of these four registers.

When ECTMCE = 0, E710TRC is always read as 0.

When ECTMCE = 1, E710TRC can be read using the 32-bit manipulation instruction.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECSYND								ECHORD							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECECRD								ECERDB							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NOTE

Changing ECTMCE from 1 to 0 resets E710TRC synchronously.

38.3.9.7 ECSYND — ECC Decoder Syndrome Data Register

ECSYND is a read-only register used to confirm the syndrome code generated by the decoding circuit in test mode (ECTMCE = 1).

Write access to ECSYND is ignored.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	SYND[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

When read, the ECSYND bits return the value of the syndrome code (SYND[6:0]) generated based on the data input to the decoding circuit.

The ECSYND bits are not holding circuits; the register value changes as the input signal changes.

ECSYND is valid only when ECTMCE = 1, and is always read as 00_H when ECTMCE = 0.

Bit 7 is reserved. When read, the value after reset is returned. When writing, write the value after reset.

38.3.9.8 ECHORD — ECC 7-Bit Redundant Data Holding Test Register

ECHORD holds the 7-bit ECC redundant area (upper 7-bit RAM data) that cannot be confirmed by the peripheral module when the peripheral module accesses the RAM for reading in test mode (ECTMCE = 1).

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	HORD[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

The ECHORD bits are loaded with the upper 7-bit RAM output data at the next rising edge of the operating clock signal after the peripheral module accesses the RAM for reading data in test mode (ECTMCE = 1).

The ECHORD bits are also loaded with the data on the input pins EC7TERI38 to EC7TERI32 at the next operating clock pulse when the ECEDB[15:0] register is read while E710TMC:ECTRRS = 1.

ECHORD is valid only when ECTMCE = 1, and is always read as 00H when ECTMCE = 0.

Bit 7 is reserved. When read, the value after reset is returned. When writing, write the value after reset.

38.3.9.9 ECECRD — ECC Encoder Test Register

ECECRD is a read-only register used to read the 7-bit redundant section generated by the encoding circuit in test mode (ECTMCE = 1).

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ECECRD[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

ECECRD is used to confirm the redundant bits generated by the input data from the peripheral module. Here, the data that is read is the result of encoding (ECC[6:0]), not the output value.

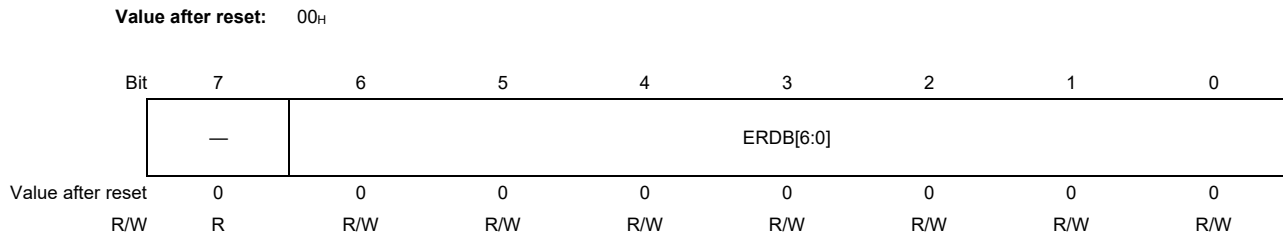
ECECRD is valid only when ECTMCE = 1, and is always read as 00_H when ECTMCE = 0.

Bit 7 is reserved. When read, the value after reset is returned. When writing, write the value after reset.

38.3.9.10 ECERDB — ECC Redundant Bit Input and Output Substitution Buffer Register

ECERDB is a buffer register for the data that substitutes for the input and output data for the 7-bit ECC redundant data area in test mode (ECTMCE = 1).

ECERDB can be read and written to in ECC test mode (ECTMCE = 1).



When ECREOS = 1, the ECERDB value, instead of the seven redundant bits generated by the encoding circuit, is output to the pin and supplied to the RAM.

When ECREIS = 1, the ECERDB value, instead of the upper seven data bits to be input to the decoding circuit, is handled by the decoding circuit.

When ECTRRS = 1, reading ECERDB returns the signal value supplied to RAM instead of the data written to ECERDB.

Bit 7 is reserved. When read, the value after reset is returned. When writing, write the value after reset.

38.3.9.11 E710EAD — ECC Error Address Register

E710EAD is a read-only register used to hold the address at which an ECC error has occurred.

If an ECC error is detected while ECC error detection is enabled, the RAM address is latched using the detection signal as a trigger, and the address is stored in E710EAD as the address at which the ECC error occurred.

The address is stored upon detection of the first ECC error while no error status is set. However, if a 1-bit error is followed by a 2-bit error, the address of the latter is stored. The address can be calculated by adding the base address.

Only one address can be held in E710EAD.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECEAD[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECEAD[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

38.3.9.12 ETCTXECCCTL— Ethernet TX RAM ECC Control Register

This register controls 1-bit ECC error correction for data read from Ethernet TX RAM.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	EC1ECP	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R

Table 38.192 ETCTXECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	EC1ECP	1-Bit ECC Error Correction Enable 0: Enable 1-bit error correction upon error detection. 1: Disable 1-bit error correction upon error detection.
4 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.9.13 ETCRXECCCTL— Ethernet RX RAM ECC Control Register

This register controls 1-bit ECC error correction for data read from Ethernet RX RAM.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	EC1ECP	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R

Table 38.193 ETCRXECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	EC1ECP	1-Bit ECC Error Correction Enable 0: Enable 1-bit error correction upon error detection. 1: Disable 1-bit error correction upon error detection.
4 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.9.14 Notification to ECM

Detection of errors in two bits can be set for this module and the ECM is notified of detected errors.

- 1-bit Error Notification

While EC1EDIC is set to 1_B, when an error is detected in one bit from among bits 0 to 38 of data read from RAM, the ECM is notified of the 1-bit error. When ECER1F or ECER2F is already set, however, the ECM will not be notified of the error.

- 2-bit Error Notification

While EC2EDIC is set to 1_B, when an error is detected in two bits from among bits 0 to 38 of data read from RAM, the ECM is notified of the 2-bit error. When ECER2F is already set, however, the ECM will not be notified of the error.

38.3.9.15 Test Function

(1) Writing RAM Data

Write data to the peripheral RAM. However, the ECC corresponding to the written data will be written to the ECC bits simultaneously. In order to write a specified value to the ECC bits, use the ECC test mode described in **(3)** below.

(2) Reading RAM Data

- Set the ECTHM bit in the E710CTL register to 1 to disable ECC error detection and correction.
- Read the peripheral RAM. Since neither error detection nor correction proceeds when the peripheral RAM is read, the RAM data is read unchanged.

How to exit this test mode:

- Set the ECTHM bit in the E710CTL register to 0 to enable ECC error detection and correction.

(3) Writing to the ECC Bits

- Set the ECTMCE bit in the E710TMC register to 1 to set ECC test mode.
- Write the value to be written to the ECC bits to E710TRC.ECERDB[6:0].
- Set the ECREOS bit in the E710TMC register to 1 to select writing of the value of the E710TRC.ECERDB[6:0] bits to the ECC bits.
- When data is written to the peripheral RAM, the value in the E710TRC.ECERDB[6:0] bits will be written to the ECC bits.

How to exit this test mode:

- Set the ECTMCE bit in the E710TMC register to 0 to set normal mode.

(4) Reading the ECC Bits

- Set the ECTMCE bit in the E710TMC register to 1 to set ECC test mode.
- When data in the peripheral RAM is read, the ECC bits are stored in the E710TRC.ECHORD[6:0] bits.

How to exit this test mode:

- Set the ECTMCE bit in the E710TMC register to 0 to set normal mode.

38.3.10 ECC for Peripheral RAM (16 Bits)

38.3.10.1 Overview

This is an ECC module for the RAM of the following peripheral module:

DFE

Error Detection and Correction

Six-bit ECC data is appended to the 16-bit RAM data.

This ECC module provides 2-bit ECC error detection and 1-bit ECC error detection and correction.

Table 38.194 DFE RAM ECC

Item	Description
ECC error detection and correction	<p>ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected.</p> <ul style="list-style-type: none"> • ECC error detection and correction are carried out (2-bit error detection, and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled; 2-bit error detection, and 1-bit error detection and correction are carried out.</p>
Error notification	<p>Upon occurrence of an ECC error, it is notified to the Error Control Module.</p> <p>ECC Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is disabled upon detection of an ECC 1-bit error.</p> <p>An ECC 2-bit error and an ECC 1-bit error are handled as individual source in ECM. All overflow errors of each peripheral module (RS-CANFD, FlexRay, GTM and DFE) are notified to ECM as one factor. An ECC error signal is only issued to the ECM, if the ECC error address is not yet stored in the error address buffer.</p>
Error status	<p>A status register is provided, which indicates the statuses of ECC 2-bit error detection and ECC 1-bit error detection. A one-stage buffer is provided for 2-bit error and 1-bit error. The error status can be cleared using the clear register.</p>
Address capture	<p>The address is captured when an ECC 2-bit error or an ECC 1-bit error is detected. The error status serves as the enable bit of the capture address. Address buffer is updated if the error status is cleared.</p> <p>One-stage address buffer is provided for an ECC 2-bit error and an ECC 1-bit error.</p>
Self-diagnosis	<ul style="list-style-type: none"> • By setting the test mode, register values can be used as the data to be output to the RAM. When a peripheral module writes to the RAM, the E610TED.ECEDB[15:0] bit value can be written to the RAM data section, and the E610TED.ERDB[5:0] bit value can be written to the ECC redundant bit section. • By setting the test mode, the ECC redundant bit section can be latched when RAM data is read, and the value can be confirmed. • By setting the test mode, the ECC redundant bit (encoding circuit) and syndrome code (decoding circuit), which are generated from the input data, can be confirmed.

38.3.10.2 List of Registers

(1) List of ECC Modules

The RAMs of the multiple peripheral functions are provided with the ECC modules. The following table shows the peripheral functions provided with the ECC modules, the corresponding ECC module names, and base addresses of the ECC modules.

Table 38.195 List of ECC modules

Peripheral Functions		ECC Module Names and Register Base Addresses	
		Module Name	Base Address <base_addr>
DFE0 (Peripheral Group5)	Data memory (DMEM0)	E6DF00	FFC7 8020 _H
	Data memory (DMEM1)	E6DF01	FFC7 8040 _H
DFE1 (Peripheral Group5)	Data memory (DMEM0)	E6DF10	FFC7 8080 _H
	Data memory (DMEM1)	E6DF11	FFC7 80A0 _H

(2) List of Registers

Each ECC module has the registers shown in the following table.

Table 38.196 List of Registers

Register Name	Symbol	Address	Access	Access Protection
ECC Control Register	E610CTL	<base_addr> + 00 _H	8, 16	
ECC Test Mode Control Register	E610TMC	<base_addr> + 04 _H	8, 16	
ECC Redundant Bit Data Control Test Register	E610TRC	<base_addr> + 08 _H	32	
ECC Encoder and Decoder Data Test Register	E610TED	<base_addr> + 0C _H	32	
ECC Error Address Register	E610EAD	<base_addr> + 10 _H	32	

(3) Register Map

Table 38.197 Register Map

Symbol	31	24	23	16	15	8	7	0	Address
E610CTL	— (00 _H)		— (00 _H)		ECCTL[15:8]		ECCTL[7:0]		00 _H
E610TMC	— (00 _H)		— (00 _H)		ECTMC[15:8]		ECTMC[7:0]		04 _H
E610TRC	ECSYND[7:0]		ECHORD[7:0]		ECECRD[7:0]		ECRODM[7:0]		08 _H
E610TED	ECEDB[31:24]		ECEDB[23:16]		ECRIDM[15:8]		ECERDB[7:0]		0C _H
E610EAD	ECEAD[31:24]		ECEAD[23:16]		ECEAD[15:8]		ECEAD[7:0]		10 _H

38.3.10.3 E610CTL — ECC Control Register

E610CTL controls the status and modes of the ECC modules.

E610CTL can be read and written to using the 16-bit or 8-bit manipulation instruction.

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA[1:0]	—	—	—	ECOVFF	ECER2C	ECER1C	—	ECTHM	—	EC1ECP	EC2EDC	EC1EDC	ECER2F	ECER1F	ECEMF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	Undefined
	R/W	R/W*	R/W*	R	R	R	R/W*	R/W*	R	R/W	R	R/W	R/W	R/W	R	R

Note 1. This bit is always read as 0.

Table 38.198 E610CTL Register Contents (1/2)

Bit Position	Bit Name	Function
15, 14	EMCA[1:0]	Access Control 1 and 0 to ECC Mode Select Bit These bits enable or disable writing to bit 7. The read value is always 0.
13, 12	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
11	ECOVFF	Error Overflow Detection Flag This flag is set when an error of any source other than the same error address occurs while the error flag (ECER2F or ECER1F) is set. This flag is cleared by an internal or external reset, by disabling ECC function (ECTHM = 1), or by writing 1 to ECER2C and ECER1C.
10	ECER2C	2-Bit ECC Error Detection Flag Clear Clears the bit 2 (ECER2F) status flag. The read value is always 0, and writing 0 to ECER2C does not change the state. If writing 1 to ECER2C conflicts with the condition for setting bit 2, the former takes priority. Writing 1 to ECER2C while ECER2F is set clears ECER2F.
9	ECER1C	1-Bit ECC Error Detection And Correction Accumulated Flag Clear Clears the bit 1 (ECER1F) status flag. The read value is always 0, and writing 0 to ECER1C does not change the state. If writing 1 to ECER1C conflicts with the condition for setting bit 1, the former takes priority. Writing 1 to ECER1C while ECER1F is set clears ECER1F.
8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	ECTHM	ECC Function Disable Selection Selects the ECC decoding operation. Write access to ECTHM is enabled when the value of bits 15 and 14 is 01 _B . Therefore, only the 16-bit manipulation instruction is valid. Setting ECTHM to 1 disables error detection and bit correction. Here, if the data to be output to the peripheral module contains an error, the data is output without bit correction. Setting ECTHM to 1 has no effect on the encoder side. 0: Enable ECC detection and correction 1: Disable ECC detection and correction. Has no effect on the encoder side.
6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	EC1ECP	1-Bit ECC Error Correction Enable 0: Enable 1-bit error correction upon error detection. 1: Disable 1-bit error correction upon error detection.

Table 38.198 E610CTL Register Contents (2/2)

Bit Position	Bit Name	Function
4	EC2EDIC	<p>2-Bit ECC Error Detection Notification Enable</p> <p>0: When a 2-bit error is detected, the ECM is not notified of the error. 1: When a 2-bit error is detected, the ECM is notified of the error.</p>
3	EC1EDIC	<p>1-Bit ECC Error Detection Notification Enable</p> <p>0: When a 1-bit error is detected, the ECM is not notified of the error. 1: When a 1-bit error is detected, the ECM is notified of the error.</p>
2	ECER2F	<p>2-Bit ECC Error Detection Flag</p> <p>Indicates that errors have been detected at two bits in bits 0 to 21 of the data read from the RAM during RAM read access while error detection is enabled.</p> <p>This bit is a read only flag.</p> <p>0: A 2-bit error has not occurred since this bit was cleared. 1: A 2-bit error has occurred.</p> <p>Clearing conditions</p> <p>(1) Reset is applied. (2) 1 is written to ECER2C. (3) ECC function is disabled (ECTHM = 1).</p>
1	ECER1F	<p>1-Bit ECC Error Detection And Correction Flag</p> <p>Indicates that an error has been detected at one bit in bits 0 to 21 of the data read from the RAM during RAM read access while error detection is enabled. Setting this bit generates no interrupt signal.</p> <p>This bit is a read only flag.</p> <p>0: A 1-bit error has not occurred since this bit was cleared. 1: A 1-bit error has occurred.</p> <p>Clearing conditions</p> <p>(1) Reset is applied. (2) 1 is written to ECER1C. (3) ECC function is disabled.</p>
0	ECEMF	<p>ECC Error Message Flag</p> <p>Indicates that the current read data contains an error.</p> <p>ECEMF is updated every time RAM data is read. Since the RAM value after reset is undefined, if this bit is read before the RAM is initialized, this bit might be set, indicating an error. Therefore, the ECEMF value after reset is undefined.</p> <p>0: The current RAM data contains no bit errors. 1: The current RAM data contains bit errors.</p> <p>Clearing conditions</p> <p>(1) ECC function is disabled. (2) Decoding circuit input data contains no 1-bit errors.</p> <p>ECEMF remains set as long as RAM data containing a bit error is output with error detection being enabled.</p>

38.3.10.4 E610TMC — ECC Test Mode Control Register

E610TMC is a 16-bit register used to switch the mode to test mode and control the mode.

E610TMC can be read and written to using the 16-bit or 8-bit manipulation instruction.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA[1:0]		—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*	R/W*	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Note 1. This bit is always read as 0.

Table 38.199 E610TMC Register Contents (1/2)

Bit Position	Bit Name	Function
15, 14	ETMA[1:0]	Sets Access Control 1 and 0 to ECC Test Mode Control Enable Bit These bits enable or disable writing to bit 7. The read value is always 0.
13 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	ECTMCE	ECC Test Mode Control Enable ECTMCE enables or disables access to the test registers and test control bits. Write access to ECTMCE is enabled when the value of bits 15 and 14 is 10 _B . 0: Disable access to the test registers and test control bits. 1: Enable access to the test registers and test control bits.
6, 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	ECTRRS	ECC RAM Read Test Mode Control Enable ECTRRS enables the read status of RAM to be generated by reading the E610TED register, and also allows the RAM output data to be read out when the E610TED:ECEDB[15:0] and E610TED:ECRIDM[7:0] are read. Write access to ECTRRS is enabled only when ECTMCE is 1 (can be set simultaneously). ECTRRS is cleared by writing 0 to ECTMCE (can be cleared synchronously). 0: Disable generation of RAM read status for testing when E610TED is read. 1: Enable generation of RAM read status for testing when E610TED is read. When E610TED:ECEDB[15:0] and E610TED:ECRIDM[7:0] are read, the values of the RAM output data pin are read out.
3	ECREOS	ECC Redundant Bit Output Data Select ECREOS selects either the ECC encoder output data or the ECERDB register value to be output as the ECC redundant bit output. Write access to ECREOS is enabled only when ECTMCE is 1 (can be set simultaneously). ECREOS is cleared by writing 0 to ECTMCE (can be cleared synchronously). 0: Allow encoding results to be output as the ECC redundant bit output. 1: Allow the E610TED:ECERDB[5:0] value to be output as the ECC redundant bit output.
2	ECENS	ECC Encoder Input Select ECENS selects either the data value from the peripheral module or the internal test register value (E610TED: ECEDB[15:0]) as the input signal to be encoded. Write access to ECENS is enabled only when ECTMCE is 1 (can be set simultaneously). ECENS is cleared by writing 0 to ECTMCE (can be cleared synchronously). 0: Allow the RAM write data from the peripheral module to be input as the ECC encoder input data. 1: Allow the E610TED:ECEDB[15:0] value to be input as the ECC encoder input data.

Table 38.199 E610TMC Register Contents (2/2)

Bit Position	Bit Name	Function
1	ECDCS	<p>ECC Decoder Input Select</p> <p>ECDCS selects either the lower 16-bit data value from the RAM or the internal test register value (E610TED:ECEDB[15:0]) as the lower 32-bit data of the input signal to be decoded. Write access to ECDCS is enabled only when ECTMCE is 1 (can be set simultaneously). ECDCS is cleared by writing 0 to ECTMCE (can be cleared synchronously).</p> <p>0: Allow the lower 16-bit RAM output data to be input to the data area (lower 16 bits) to the decoder.</p> <p>1: Allow the E610TED:ECEDB[15:0] value to be input to the data area to the decoder.</p>
0	ECREIS	<p>ECC Redundant Bit Input Data Select</p> <p>ECREIS selects either the upper 6-bit data value from the RAM (redundant bit area) or the internal test register value (E610TED:ECERDB[5:0]) as the upper 6-bit data of the input signal to be decoded. Write access to ECREIS is enabled only when ECTMCE is 1 (can be set simultaneously). ECREIS is cleared by writing 0 to ECTMCE (can be cleared synchronously).</p> <p>0: Allow the upper 6-bit RAM output data to be input to the ECC redundant bit area to the decoder.</p> <p>1: Allow the E610TED:ECERDB[5:0] value to be input to the ECC redundant bit area to the decoder.</p>

38.3.10.5 E610TED — ECC Encoder and Decoder Data Test Register

E610TED is a 32-bit test register consisting of three fields (ECEDB, ECRIDM, and ECERDB) corresponding to the ECC data area and redundant bit area. Each field can be accessed as the register with the same name. For details of each field, refer to the descriptions of these three registers.

When ECTMCE = 1, E610TED can be read and written to using the 32-bit manipulation instruction.

When ECTMCE = 0, E610TED is always read as 0.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECEDB																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECRIDM								ECERDB								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

NOTE

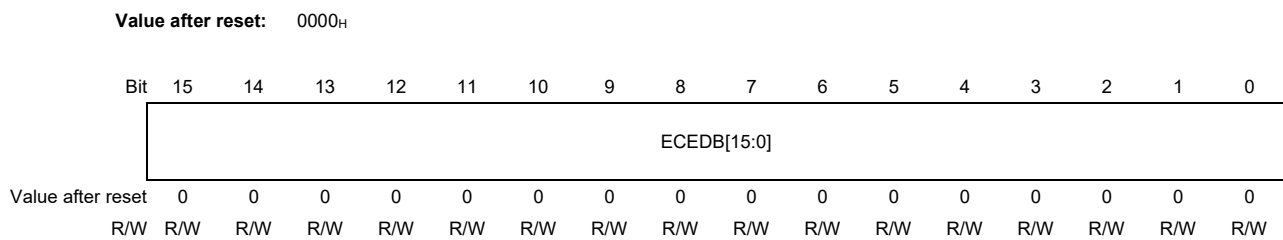
Changing ECTMCE from 1 to 0 resets E610TED synchronously.

38.3.10.6 ECEDB — ECC Encoder and Decoder Data Input and Output Substitution Buffer Register

ECEDB is a 16-bit data buffer register for ECC encoding and decoding.

ECEDB can be read and written to in test mode (ECTMCE = 1).

When ECTMCE = 1, the ECEDB value can be used as the data input to the encoding circuit and decoding circuit.



When E610TMC:ECENS = 1, the ECEDB value is input to the encoding circuit and supplied to the RAM.

When E610TMC:ECDCS = 1, the ECEDB value is input as the 15th to 0th bits of the input data to the decoding circuit.

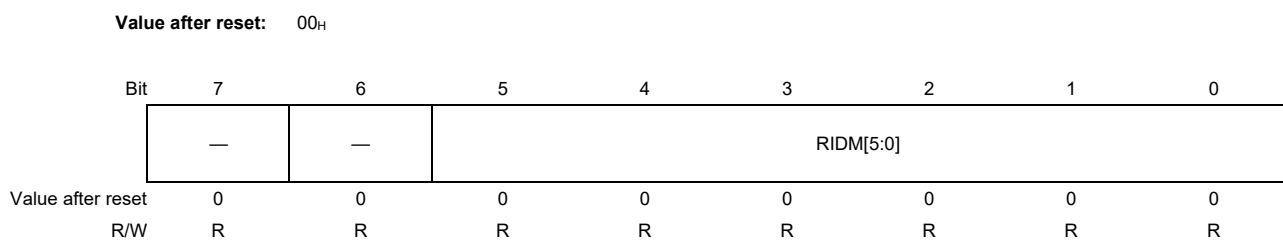
When E610TMC:ECTRRS = 1, reading ECEDB returns the RAM output data instead of the data written to ECEDB.

38.3.10.7 ECRIDM — ECC Redundant Bit Input Data Monitoring Register

ECRIDM is a read-only register used to monitor the upper 6-bit data (redundant bit area) value from the RAM in test mode (ECTMCE = 1) when E610TMC:ECTRRS = 1. With ECRIDM, the RAM output data can be confirmed easily.

When ECTRRS = 0, ECRIDM is always read as 0.

Bit 7 and 6 are reserved. When read, the value after reset is returned. When writing, write the value after reset.



38.3.10.8 ECERDB — ECC Redundant Bit Input and Output Substitution Buffer Register

ECERDB is a buffer register for the data that substitutes for the input and output data for the 6-bit ECC redundant data area in test mode (ECTMCE = 1).

ECERDB can be read and written to in test mode (ECTMCE = 1).

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	ERDB[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

When ECREOS = 1, the ECERDB value, instead of the six redundant bits generated by the encoding circuit, is output to the ECC redundant bit output pin and supplied to the RAM.

When ECREIS = 1, the ECERDB value, instead of the upper six data bits to be input to the decoding circuit, is handled by the decoding circuit.

Bit 7 and 6 are reserved. When read, the value after reset is returned. When writing, write the value after reset.

38.3.10.9 E610TRC — ECC Redundant Bit Data Control Test Register

E610TRC is a 32-bit test register consisting of four fields (ECSYND, ECHORD, ECECRD, and ECRODM) corresponding to the ECC redundant bit area. Each field can be accessed as the 8-bit register with the same name. For details of each field, refer to the descriptions of these four registers.

When ECTMCE = 0, E610TRC is always read as 0.

When ECTMCE = 1, E610TRC can be read using the 32-bit manipulation instruction.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECSYND								ECHORD							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECECRD								ECRODM							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

38.3.10.10 ECSYND — ECC Decoder Syndrome Data Register

ECSYND is a read-only register used to confirm the syndrome code generated by the decoding circuit in test mode (ECTMCE = 1).

Write access to ECSYND is ignored.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	SYND[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

When read, the ECSYND bits return the value of the syndrome code (SYND[5:0]) generated based on the data input to the decoding circuit.

The ECSYND bits are not holding circuits; the register value changes as the input signal changes.

ECSYND is valid only when ECTMCE = 1, and is always read as 00_H when ECTMCE = 0.

Bit 7 and 6 are reserved. When read, the value after reset is returned. When writing, write the value after reset.

38.3.10.11 ECHORD — ECC 6-Bit Redundant Data Holding Test Register

ECHORD holds the 6-bit ECC redundant area (upper 6-bit RAM data) that cannot be confirmed by the peripheral module when the peripheral module accesses the RAM for reading in test mode (ECTMCE = 1).

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	HORD[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

The ECHORD bits are loaded with the upper 6-bit RAM output data at the next rising edge of the operating clock signal after the peripheral module accesses the RAM for reading data in test mode (ECTMCE = 1).

The ECHORD bits are also loaded with the data on the input pins at the next operating clock pulse when the ECEDB[15:0] register is read while E610TMC:ECTRRS = 1.

ECHORD is valid only when ECTMCE = 1, and is always read as 00_H when ECTMCE = 0.

Bit 7 and 6 are reserved. When read, the value after reset is returned. When writing, write the value after reset.

38.3.10.12 ECECRD — ECC Encoder Test Register

ECECRD is a read-only register used to read the 6-bit redundant section generated by the encoding circuit in test mode (ECTMCE = 1).

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	ECRD[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

ECECRD is used to confirm the redundant bits generated by the input data from the peripheral module. Here, the data that is read is the result of encoding (ECC[5:0]), not the value output from the ECC redundant bits.

ECECRD is valid only when ECTMCE = 1, and is always read as 00_H when ECTMCE = 0.

Bit 7 and 6 are reserved. When read, the value after reset is returned. When writing, write the value after reset.

38.3.10.13 ECRODM — ECC Redundant Bit Output Data Monitoring Register

ECRODM is a read-only register used to monitor the value of the ECC redundant bit output pin in test mode (ECTMCE = 1).

With ECRODM, the upper 6-bit output data to the RAM can be confirmed easily.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	RODM[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

ECRODM is valid only when ECTMCE = 1, and is always read as 00_H when ECTMCE = 0.

Bit 7 and 6 are reserved. When read, the value after reset is returned. When writing, write the value after reset.

38.3.10.14 E610EAD — ECC Error Address Register

E610EAD is a read-only register used to hold the address at which an ECC error has occurred.

If an ECC error is detected while ECC error detection is enabled, the RAM address is latched using the detection signal as a trigger, and the address is stored in E610EAD as the address at which the ECC error occurred.

The address is stored upon detection of the first ECC error while no error status is set. However, if a 1-bit error is followed by a 2-bit error, the address of the latter is stored. The address can be calculated by adding the base address.

Only one address can be held in E610EAD.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECEAD[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECEAD[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

38.3.10.15 Notification to ECM

Detection of errors in two bits can be set for this module and the ECM is notified of detected errors.

- 1-bit Error Notification

While EC1EDIC is set to 1_B, when an error is detected in one bit from among bits 0 to 21 of data read from RAM, the ECM is notified of the 1-bit error. When ECER1F or ECER2F is already set, however, the ECM will not be notified of the error.

- 2-bit Error Notification

While EC2EDIC is set to 1_B, when an error is detected in two bits from among bits 0 to 21 of data read from RAM, the ECM is notified of the 2-bit error. When ECER2F is already set, however, the ECM will not be notified of the error.

38.3.10.16 Test Function

(1) Writing RAM Data

Write data to the peripheral RAM. However, the ECC corresponding to the written data will be written to the ECC bits simultaneously. In order to write a specified value to the ECC bits, use the ECC test mode described in **(3)** below.

(2) Reading RAM Data

- Set the ECTHM bit in the E610CTL register to 1 to disable ECC error detection and correction.
- Read the peripheral RAM. Since neither error detection nor correction proceed when the peripheral RAM is read, the RAM data is read unchanged.

How to exit this test mode:

- Set the ECTHM bit in the E610CTL register to 0 to enable ECC error detection and correction.

(3) Writing to the ECC Bits

- Set the ECTMCE bit in the E610TMC register to 1 to set ECC test mode.
- Write the value to be written to the ECC bits to E610TED.ECERDB[5:0].
- Set the ECREOS bit in the E610TMC register to 1 to select writing of the value of the E610TED.ECERDB[5:0] bits to the ECC bits.
- When data is written to the peripheral RAM, the value in the E610TED.ECERDB[5:0] bits will be written to the ECC bits.

How to exit this test mode:

- Set the ECTMCE bit in the E610TMC register to 0 to set normal mode.

(4) Reading the ECC Bits

- Set the ECTMCE bit in the E610TMC register to 1 to set ECC test mode.
- When data in the peripheral RAM is read, the ECC bits are stored in the E610TRC.ECHORD[5:0] bits.

How to exit this test mode:

- Set the ECTMCE bit in the E610TMC register to 0 to set normal mode.

38.3.11 Safety Mechanism on Data Transfer Path

38.3.11.1 Overview

This product provides a function that protects the transaction data and the addresses on each bus. This function applies to Cluster RAM Bus, Local FLASH Bus, Global FLASH Bus, Inter-processor element Bus, Inter-cluster Bus, System Bus, Peripheral Bus and High speed Bus.

ECC protection on bus is summarized in the table below.

Table 38.200 ECC protection on Bus

Item	Description
Data ECC error detection and correction	<p>ECC error detection and correction on data bus between master and slave can be either enabled or disabled.</p> <p>When enabled, either of the following settings can be selected:</p> <ul style="list-style-type: none"> • ECC error detection and correction are carried out (2-bit error detection, 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>When disabled, neither error detection nor correction is carried out.</p> <p>In the initial state, the ECC function is enabled; 1-bit error detection and correction, and 2-bit error detection are carried out.</p>
Address ECC error detection	<p>ECC error detection on address bus between master and slave can be either enabled or disabled.</p> <p>In the initial state, 1-bit error detection and 2-bit error detection are carried out.</p>
Error notification	<p>The occurrence of a Data ECC error or an Address EDC error is reported to the Error Control Module.</p> <p>Data ECC Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an ECC 2-bit and ECC 1-bit error.</p> <p>Address EDC Error:</p> <ul style="list-style-type: none"> • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an ECC 2-bit and ECC 1-bit error.</p> <p>The error notification signal is output with an Address ECC 2-bit error, an Address ECC 1-bit error handled as one source, and a Data ECC 2-bit error and a Data ECC 1-bit error are handled as individual sources.</p>
Error status	<p>A status register is provided that indicates the status of Data ECC 2-bit error detection, Data ECC 1-bit error detection, Address ECC 2-bit error detection, and Address ECC 1-bit error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.</p>
Self-diagnosis	<p>ERRGEN is available for self-diagnosis of ECC decoder of data transfer path.</p>

38.3.11.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as an offset from the base addresses.

Table 38.201 Register Base Addresses (1/3)

Base Address Name	Base Address	Bus Group
<ECCCNT_A_CCIB0CL0_base>	FFC4 B000 _H	Peripheral Group 0
<ECCCNT_A_CCIB1CL0_base>	FFC4 B080 _H	Peripheral Group 0
<ECCCNT_SA_PE0CL0_base>	FFC4 C000 _H	Peripheral Group 0
<ECCCNT_A_GCFU0ICL0_base>	FFC4 C080 _H	Peripheral Group 0
<ECCCNT_A_GCFU0DCL0_base>	FFC4 C100 _H	Peripheral Group 0
<ECCCNT_SA_PE1CL0_base>	FFC4 C200 _H	Peripheral Group 0
<ECCCNT_A_GCFU1ICL0_base>	FFC4 C280 _H	Peripheral Group 0
<ECCCNT_A_GCFU1DCL0_base>	FFC4 C300 _H	Peripheral Group 0
<ECCCNT_A_X2VCL0_base>	FFC4 D000 _H	Peripheral Group 0
<ECCCNT_A_BARR_base>	FFC4 D200 _H	Peripheral Group 0
<ECCCNT_A_IPIR_base>	FFC4 D280 _H	Peripheral Group 0
<ECCCNT_A_MEV_base>	FFC4 D300 _H	Peripheral Group 0
<ECCCNT_A_TPTM_base>	FFC4 D380 _H	Peripheral Group 0
<ECCCNT_A_CRAM_base>	FFC4 D400 _H	Peripheral Group 0
<ECCCNT_A_SG0_base>	FFC4 D480 _H	Peripheral Group 0
<ECCCNT_A_SX2PV_base>	FFC4 D500 _H	Peripheral Group 0
<ECCCNT_A_SX2FX_base>	FFC4 D580 _H	Peripheral Group 0
<ECCCNT_A_FX2SX_base>	FFC4 D600 _H	Peripheral Group 0
<ECCCNT_A_GCFUF_base>	FFC4 D680 _H	Peripheral Group 0
<ECCCNT_A_SX2MB_base>	FFC4 D700 _H	Peripheral Group 0
<ECCCNT_A_V2A1_base>	FFC6 3800 _H	Peripheral Group 1
<ECCCNT_A_V2A2_base>	FFDE 0000 _H	Peripheral Group 2H
<ECCCNT_A_V2A3_base>	FFC7 2000 _H	Peripheral Group 3
<ECCCNT_A_V2A4_base>	FFC7 4000 _H	Peripheral Group 4
<ECCCNT_A_V2A5_base>	FFC7 9000 _H	Peripheral Group 5
<ECCCNT_A_V2A6_base>	FFC8 0000 _H	Peripheral Group 6
<ECCCNT_A_V2A7_base>	FFF4 8000 _H	Peripheral Group 7
<ECCCNT_A_V2A8_base>	FFF5 8000 _H	Peripheral Group 8
<ECCCNT_A_V2A9_base>	FF0A 0000 _H	Peripheral Group 9
<ECCCNT_D_V2XWCL0_base>	FFC4 E000 _H	Peripheral Group 0
<ECCCNT_D_V2XRCL0_base>	FFC4 E080 _H	Peripheral Group 0
<ECCCNT_D_PV2APBW_base>	FFC4 E400 _H	Peripheral Group 0
<ECCCNT_D_PV2APBR_base>	FFC4 E480 _H	Peripheral Group 0
<ECCCNT_D_CRAM_base>	FFC4 E500 _H	Peripheral Group 0
<ECCCNT_D_EMU_base>	FFC4 E580 _H	Peripheral Group 0
<ECCCNT_D_DMDE0_base>	FFC4 E600 _H	Peripheral Group 0
<ECCCNT_D_DMDE1_base>	FFC4 E680 _H	Peripheral Group 0
<ECCCNT_SD_PE0CL0_base>	FFC4 F000 _H	Peripheral Group 0

Table 38.201 Register Base Addresses (2/3)

Base Address Name	Base Address	Bus Group
<ECCCNT_MD_PE0CL0_base>	FFC4 F080 _H	Peripheral Group 0
<ECCCNT_SD_PE1CL0_base>	FFC4 F100 _H	Peripheral Group 0
<ECCCNT_MD_PE1CL0_base>	FFC4 F180 _H	Peripheral Group 0
<ECCCNT_D_BARR_base>	FFC4 F800 _H	Peripheral Group 0
<ECCCNT_D_IPIR_base>	FFC4 F880 _H	Peripheral Group 0
<ECCCNT_D_MEV_base>	FFC4 F900 _H	Peripheral Group 0
<ECCCNT_D_TPTM_base>	FFC4 F980 _H	Peripheral Group 0
<ECCCNT_D_DTS_base>	FFC4 FA00 _H	Peripheral Group 0
<ECCCNT_D_DMA0_base>	FFC4 FB00 _H	Peripheral Group 0
<ECCCNT_D_DMA1_base>	FFC4 FB80 _H	Peripheral Group 0
<ECCCNT_D_V2A1W_base>	FFC6 3880 _H	Peripheral Group 1
<ECCCNT_D_V2A2W_base>	FFDE 0080 _H	Peripheral Group 2H
<ECCCNT_D_V2A3W_base>	FFC7 2080 _H	Peripheral Group 3
<ECCCNT_D_V2A4W_base>	FFC7 4080 _H	Peripheral Group 4
<ECCCNT_D_V2A5W_base>	FFC7 9080 _H	Peripheral Group 5
<ECCCNT_D_V2A6W_base>	FFC8 0080 _H	Peripheral Group 6
<ECCCNT_D_V2A7W_base>	FFF4 8080 _H	Peripheral Group 7
<ECCCNT_D_V2A8W_base>	FFF5 8080 _H	Peripheral Group 8
<ECCCNT_D_V2A9W_base>	FF0A 0080 _H	Peripheral Group 9
<ECCCNT_D_V2A1R_base>	FFC6 3900 _H	Peripheral Group 1
<ECCCNT_D_V2A2R_base>	FFDE 0100 _H	Peripheral Group 2H
<ECCCNT_D_V2A3R_base>	FFC7 2100 _H	Peripheral Group 3
<ECCCNT_D_V2A4R_base>	FFC7 4100 _H	Peripheral Group 4
<ECCCNT_D_V2A5R_base>	FFC7 9100 _H	Peripheral Group 5
<ECCCNT_D_V2A6R_base>	FFC8 0100 _H	Peripheral Group 6
<ECCCNT_D_V2A7R_base>	FFF4 8100 _H	Peripheral Group 7
<ECCCNT_D_V2A8R_base>	FFF5 8100 _H	Peripheral Group 8
<ECCCNT_D_V2A9R_base>	FF0A 0100 _H	Peripheral Group 9
<BECCCAP_LRAM_base>	FFC5 4000 _H	Peripheral Group 0
<BECCCAP_CRAM_base>	FFC5 4100 _H	Peripheral Group 0
<BECCCAP_CFL_base>	FFC5 4300 _H	Peripheral Group 0
<BECCCAP_PERI_base>	FFC5 4400 _H	Peripheral Group 0
<BECCCAP_DMDT_base>	FFC5 4500 _H	Peripheral Group 0
<BECCCAP_EMU_base>	FFC5 4800 _H	Peripheral Group 0
<PB0ECC_base>	FFC5 8000 _H	Peripheral Group 0
<PB1ECC_base>	FFC6 2E00 _H	Peripheral Group 1
<PB2LECC_base>	FFDA 0000 _H	Peripheral Group 2L
<PB2HECC_base>	FFDE 0800 _H	Peripheral Group 2H
<PB3ECC_base>	FFC7 2800 _H	Peripheral Group 3
<PB4ECC_base>	FFC7 5000 _H	Peripheral Group 4
<PB5ECC_base>	FFC7 A000 _H	Peripheral Group 5
<PB6ECC_base>	FFC8 0800 _H	Peripheral Group 6
<PB7ECC_base>	FFF4 8800 _H	Peripheral Group 7
<PB8ECC_base>	FFF5 8800 _H	Peripheral Group 8
<PB9ECC_base>	FF0A 1000 _H	Peripheral Group 9

Table 38.201 Register Base Addresses (3/3)

Base Address Name	Base Address	Bus Group
<HB91MECC_base>	FF0C 0C00 _H	Peripheral Group 9 (RHSIF0 Master)
<HB91SECC_base>	FF0C 0E00 _H	Peripheral Group 9 (RHSIF0 Slave)
<HB93MECC_base>	FF0C 0000 _H	Peripheral Group 9 (FlexRay0 Master)
<HB93SECC_base>	FF0C 0200 _H	Peripheral Group 9 (FlexRay0 Slave)
<HB95MECC_base>	FF0C 0800 _H	Peripheral Group 9 (Ethernet Master)
<HB95SECC_base>	FF0C 0A00 _H	Peripheral Group 9 (Ethernet Slave)
<HB96MECC_base>	FF0C 1400 _H	Peripheral Group 9 (HSSPI0 Master)
<BECCCAP_V2A1_base>	FFC6 3A00 _H	Peripheral Group 1
<BECCCAP_V2A2_base>	FFDE 0200 _H	Peripheral Group 2H
<BECCCAP_V2A3_base>	FFC7 2200 _H	Peripheral Group 3
<BECCCAP_V2A4_base>	FFC7 4200 _H	Peripheral Group 4
<BECCCAP_V2A5_base>	FFC7 9200 _H	Peripheral Group 5
<BECCCAP_V2A6_base>	FFC8 0200 _H	Peripheral Group 6
<BECCCAP_V2A7_base>	FFF4 8200 _H	Peripheral Group 7
<BECCCAP_V2A8_base>	FFF5 8200 _H	Peripheral Group 8
<BECCCAP_V2A9_base>	FF0A 0200 _H	Peripheral Group 9

38.3.11.3 List of Registers

Table 38.202 List of Registers (1/4)

Module Name	Register Name	Symbol	Address	Access	Access Protection
ECCCNT_A_CCIBnCL0 (n = 0-1)	ECC Control Register	BUSAECCCTL	<ECCCNT_A_CCIBnCL0_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_CCIBnCL0_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_CCIBnCL0_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_CCIBnCL0_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_SA_PEnCL0 (n = 0, 1)	ECC Control Register	BUSAECCCTL	<ECCCNT_SA_PEnCL0_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_SA_PEnCL0_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_SA_PEnCL0_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_SA_PEnCL0_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_A_GCFUnICL0 (n = 0, 1)	ECC Control Register	BUSAECCCTL	<ECCCNT_A_GCFUnICL0_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_GCFUnICL0_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_GCFUnICL0_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_GCFUnICL0_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_A_GCFUnDCL0 (n = 0, 1)	ECC Control Register	BUSAECCCTL	<ECCCNT_A_GCFUnDCL0_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_GCFUnDCL0_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_GCFUnDCL0_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_GCFUnDCL0_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_A_X2VCL0	ECC Control Register	BUSAECCCTL	<ECCCNT_A_X2VCL0_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_X2VCL0_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_X2VCL0_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_X2VCL0_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_A_BARR	ECC Control Register	BUSAECCCTL	<ECCCNT_A_BARR_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_BARR_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_BARR_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_BARR_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_A_IPIR	ECC Control Register	BUSAECCCTL	<ECCCNT_A_IPIR_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_IPIR_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_IPIR_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_IPIR_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_A_MEV	ECC Control Register	BUSAECCCTL	<ECCCNT_A_MEV_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_MEV_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_MEV_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_MEV_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_A_TPTM	ECC Control Register	BUSAECCCTL	<ECCCNT_A_TPTM_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_TPTM_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_TPTM_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_TPTM_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_A_CRAM	ECC Control Register	BUSAECCCTL	<ECCCNT_A_CRAM_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_CRAM_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_CRAM_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_CRAM_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_A_SG0	ECC Control Register	BUSAECCCTL	<ECCCNT_A_SG0_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_SG0_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_SG0_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_SG0_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_A_SX2PV	ECC Control Register	BUSAECCCTL	<ECCCNT_A_SX2PV_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_SX2PV_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_SX2PV_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_SX2PV_base> + 40 _H	8, 16, 32	KCPROT

Table 38.202 List of Registers (2/4)

Module Name	Register Name	Symbol	Address	Access	Access Protection
ECCCNT_A_SX2FX	ECC Control Register	BUSAECCCTL	<ECCCNT_A_SX2FX_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_SX2FX_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_SX2FX_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_SX2FX_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_A_FX2SX	ECC Control Register	BUSAECCCTL	<ECCCNT_A_FX2SX_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_FX2SX_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_FX2SX_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_FX2SX_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_A_GCFUF	ECC Control Register	BUSAECCCTL	<ECCCNT_A_GCFUF_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_GCFUF_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_GCFUF_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_GCFUF_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_A_SX2MB	ECC Control Register	BUSAECCCTL	<ECCCNT_A_SX2MB_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_SX2MB_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_SX2MB_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_SX2MB_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_A_V2An (n = 1-9)	ECC Control Register	BUSAECCCTL	<ECCCNT_A_V2An_base> + 00 _H	8, 16, 32	APECKCPROT*1
	ECC Test Control Register	BUSAECCSTSTCTL	<ECCCNT_A_V2An_base> + 10 _H	8, 16, 32	APECKCPROT*1
	Test Data Input Register	BUSAECCSTDIN0	<ECCCNT_A_V2An_base> + 30 _H	8, 16, 32	APECKCPROT*1
	Test ECC Input Register	BUSAECCSTSTEIN	<ECCCNT_A_V2An_base> + 40 _H	8, 16, 32	APECKCPROT*1
ECCCNT_D_V2XWCL0	ECC Control Register	BUSDVCECCCTL	<ECCCNT_D_V2XWCL0_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSDVCECCSTSTCTL	<ECCCNT_D_V2XWCL0_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSDVCECCSTDIN0	<ECCCNT_D_V2XWCL0_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSDVCECCSTSTEIN	<ECCCNT_D_V2XWCL0_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_D_V2XRCL0	ECC Control Register	BUSDVCECCCTL	<ECCCNT_D_V2XRCL0_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSDVCECCSTSTCTL	<ECCCNT_D_V2XRCL0_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSDVCECCSTDIN0	<ECCCNT_D_V2XRCL0_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSDVCECCSTSTEIN	<ECCCNT_D_V2XRCL0_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_D_PV2APBW	ECC Control Register	BUSDVCECCCTL	<ECCCNT_D_PV2APBW_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSDVCECCSTSTCTL	<ECCCNT_D_PV2APBW_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSDVCECCSTDIN0	<ECCCNT_D_PV2APBW_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSDVCECCSTSTEIN	<ECCCNT_D_PV2APBW_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_D_PV2APBR	ECC Control Register	BUSDV1ECCTL	<ECCCNT_D_PV2APBR_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSDV1ECSTSTCTL	<ECCCNT_D_PV2APBR_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSDV1ECSTDIN0	<ECCCNT_D_PV2APBR_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSDV1ECSTSTEIN	<ECCCNT_D_PV2APBR_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_D_CRAM	ECC Control Register	BUSDVCECCCTL	<ECCCNT_D_CRAM_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSDVCECCSTSTCTL	<ECCCNT_D_CRAM_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSDVCECCSTDIN0	<ECCCNT_D_CRAM_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSDVCECCSTSTEIN	<ECCCNT_D_CRAM_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_D_EMU	ECC Control Register	BUSDEMECCCTL	<ECCCNT_D_EMU_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSDEMECCSTSTCTL	<ECCCNT_D_EMU_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSDEMECCSTDIN0	<ECCCNT_D_EMU_base> + 30 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSDEMECCSTSTEIN	<ECCCNT_D_EMU_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_D_DMDEn (n = 0, 1)	ECC Control Register	BUSDDECCCTL	<ECCCNT_D_DMDEn_base> + 00 _H	8, 16, 32	KCPROT
	ECC Test Control Register	BUSDDECCSTSTCTL	<ECCCNT_D_DMDEn_base> + 10 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSDDECCSTDIN0	<ECCCNT_D_DMDEn_base> + 30 _H	8, 16, 32	KCPROT
	Test Data Input Register	BUSDDECCSTDIN1	<ECCCNT_D_DMDEn_base> + 34 _H	8, 16, 32	KCPROT
	Test ECC Input Register	BUSDDECCSTSTEIN	<ECCCNT_D_DMDEn_base> + 40 _H	8, 16, 32	KCPROT
ECCCNT_SD_PEnCL0 (n = 0, 1)	ECC Control Register	BUSDECCCTL	<ECCCNT_SD_PEnCL0_base> + 00 _H	8, 16, 32	KCPROT

Table 38.202 List of Registers (3/4)

Module Name	Register Name	Symbol	Address	Access	Access Protection
ECCCNT_MD_PEnCL0 (n = 0, 1)	ECC Control Register	BUSDECCCTL	<ECCCNT_MD_PEnCL0_base> + 00 _H	8, 16, 32	KCPROT
ECCCNT_D_BARR	ECC Control Register	BUSDECCCTL	<ECCCNT_D_BARR_base> + 00 _H	8, 16, 32	KCPROT
ECCCNT_D_IPIR	ECC Control Register	BUSDECCCTL	<ECCCNT_D_IPIR_base> + 00 _H	8, 16, 32	KCPROT
ECCCNT_D_MEV	ECC Control Register	BUSDECCCTL	<ECCCNT_D_MEV_base> + 00 _H	8, 16, 32	KCPROT
ECCCNT_D_TPTM	ECC Control Register	BUSDECCCTL	<ECCCNT_D_TPTM_base> + 00 _H	8, 16, 32	KCPROT
ECCCNT_D_DTS	ECC Control Register	BUSDECCCTL	<ECCCNT_D_DTS_base> + 00 _H	8, 16, 32	KCPROT
ECCCNT_D_DMA _n (n = 0, 1)	ECC Control Register	BUSDECCCTL	<ECCCNT_D_DMA _n _base> + 00 _H	8, 16, 32	KCPROT
ECCCNT_D_V2AnW (n = 1-9)	ECC Control Register	BUSDVCECCCTL	<ECCCNT_D_V2AnW_base> + 00 _H	8, 16, 32	APECKCPROT*1
	ECC Test Control Register	BUSDVCECCTSTCTL	<ECCCNT_D_V2AnW_base> + 10 _H	8, 16, 32	APECKCPROT*1
	Test Data Input Register	BUSDVCECCTSTDIN0	<ECCCNT_D_V2AnW_base> + 30 _H	8, 16, 32	APECKCPROT*1
	Test ECC Input Register	BUSDVCECCTSTEIN	<ECCCNT_D_V2AnW_base> + 40 _H	8, 16, 32	APECKCPROT*1
ECCCNT_D_V2AnR (n = 1-9)	ECC Control Register	BUSDV1ECCTL	<ECCCNT_D_V2AnR_base> + 00 _H	8, 16, 32	APECKCPROT*1
	ECC Test Control Register	BUSDV1ECCTSTCTL	<ECCCNT_D_V2AnR_base> + 10 _H	8, 16, 32	APECKCPROT*1
	Test Data Input Register	BUSDV1ECCTSTDIN0	<ECCCNT_D_V2AnR_base> + 30 _H	8, 16, 32	APECKCPROT*1
	Test ECC Input Register	BUSDV1ECCTSTEIN	<ECCCNT_D_V2AnR_base> + 40 _H	8, 16, 32	APECKCPROT*1
BECCCAP_LRAM	Error Notification Control Register	LR_BUSERRINT	<BECCCAP_LRAM_base> + 00 _H	8, 16, 32	KCPROT
	Address 1-bit Error Status Clear Register	LR_BUSASSTCLR	<BECCCAP_LRAM_base> + 10 _H	8, 16, 32	
	Address 2-bit Error Status Clear Register	LR_BUSADSTCLR	<BECCCAP_LRAM_base> + 20 _H	8, 16, 32	
	Data 1-bit Error Status Clear Register	LR_BUSDSSTCLR	<BECCCAP_LRAM_base> + 30 _H	8, 16, 32	
	Data 2-bit Error Status Clear Register	LR_BUSDDSTCLR	<BECCCAP_LRAM_base> + 40 _H	8, 16, 32	
	Address 1-bit Error Status Register	LR_BUSASERSTR	<BECCCAP_LRAM_base> + 50 _H	8, 16, 32	
	Address 2-bit Error Status Register	LR_BUSADERSTR	<BECCCAP_LRAM_base> + 60 _H	8, 16, 32	
	Data 1-bit Error Status Register	LR_BUSDSERSTR	<BECCCAP_LRAM_base> + 70 _H	8, 16, 32	
BECCCAP_CRAM	Error Notification Control Register	CR_BUSERRINT	<BECCCAP_CRAM_base> + 00 _H	8, 16, 32	KCPROT
	Address 1-bit Error Status Clear Register	CR_BUSASSTCLR	<BECCCAP_CRAM_base> + 10 _H	8, 16, 32	
	Address 2-bit Error Status Clear Register	CR_BUSADSTCLR	<BECCCAP_CRAM_base> + 20 _H	8, 16, 32	
	Data 1-bit Error Status Clear Register	CR_BUSDSSTCLR	<BECCCAP_CRAM_base> + 30 _H	8, 16, 32	
	Data 2-bit Error Status Clear Register	CR_BUSDDSTCLR	<BECCCAP_CRAM_base> + 40 _H	8, 16, 32	
	Address 1-bit Error Status Register	CR_BUSASERSTR	<BECCCAP_CRAM_base> + 50 _H	8, 16, 32	
	Address 2-bit Error Status Register	CR_BUSADERSTR	<BECCCAP_CRAM_base> + 60 _H	8, 16, 32	
	Data 1-bit Error Status Register	CR_BUSDSERSTR	<BECCCAP_CRAM_base> + 70 _H	8, 16, 32	
BECCCAP_CFL	Error Notification Control Register	CF_BUSERRINT	<BECCCAP_CFL_base> + 00 _H	8, 16, 32	KCPROT
	Address 1-bit Error Status Clear Register	CF_BUSASSTCLR	<BECCCAP_CFL_base> + 10 _H	8, 16, 32	
	Address 2-bit Error Status Clear Register	CF_BUSADSTCLR	<BECCCAP_CFL_base> + 20 _H	8, 16, 32	
	Address 1-bit Error Status Register	CF_BUSASERSTR	<BECCCAP_CFL_base> + 50 _H	8, 16, 32	
	Address 2-bit Error Status Register	CF_BUSADERSTR	<BECCCAP_CFL_base> + 60 _H	8, 16, 32	
	Data 1-bit Error Status Register	CF_BUSDSERSTR	<BECCCAP_CFL_base> + 70 _H	8, 16, 32	
	Data 2-bit Error Status Register	CF_BUSDDERSTR	<BECCCAP_CFL_base> + 80 _H	8, 16, 32	
	BECCCAP_PERI	Error Notification Control Register	PH_BUSERRINT	<BECCCAP_PERI_base> + 00 _H	8, 16, 32
Address 1-bit Error Status Clear Register		PH_BUSASSTCLR	<BECCCAP_PERI_base> + 10 _H	8, 16, 32	
Address 2-bit Error Status Clear Register		PH_BUSADSTCLR	<BECCCAP_PERI_base> + 20 _H	8, 16, 32	
Data 1-bit Error Status Clear Register		PH_BUSDSSTCLR	<BECCCAP_PERI_base> + 30 _H	8, 16, 32	

Table 38.202 List of Registers (4/4)

Module Name	Register Name	Symbol	Address	Access	Access Protection
BECCCAP_PERI	Data 2-bit Error Status Clear Register	PH_BUSDDSTCLR	<BECCCAP_PERI_base> + 40 _H	8, 16, 32	
	Address 1-bit Error Status Register	PH_BUSASERSTR	<BECCCAP_PERI_base> + 50 _H	8, 16, 32	
	Address 2-bit Error Status Register	PH_BUSADERSTR	<BECCCAP_PERI_base> + 60 _H	8, 16, 32	
	Data 1-bit Error Status Register	PH_BUSDSERSTR	<BECCCAP_PERI_base> + 70 _H	8, 16, 32	
	Data 2-bit Error Status Register	PH_BUSDDERSTR	<BECCCAP_PERI_base> + 80 _H	8, 16, 32	
BECCCAP_DMDT	Error Notification Control Register	DM_BUSERRINT	<BECCCAP_DMDT_base> + 00 _H	8, 16, 32	KCPROT
	Data 1-bit Error Status Clear Register	DM_BUSDSSTCLR	<BECCCAP_DMDT_base> + 30 _H	8, 16, 32	
	Data 2-bit Error Status Clear Register	DM_BUSDDSTCLR	<BECCCAP_DMDT_base> + 40 _H	8, 16, 32	
	Data 1-bit Error Status Register	DM_BUSDSERSTR	<BECCCAP_DMDT_base> + 70 _H	8, 16, 32	
	Data 2-bit Error Status Register	DM_BUSDDERSTR	<BECCCAP_DMDT_base> + 80 _H	8, 16, 32	
BECCCAP_EMU	Error Notification Control Register	EM_BUSERRINT	<BECCCAP_EMU_base> + 00 _H	8, 16, 32	KCPROT
	Address 1-bit Error Status Clear Register	EM_BUSASSTCLR	<BECCCAP_EMU_base> + 10 _H	8, 16, 32	
	Address 2-bit Error Status Clear Register	EM_BUSADSTCLR	<BECCCAP_EMU_base> + 20 _H	8, 16, 32	
	Data 1-bit Error Status Clear Register	EM_BUSDSSTCLR	<BECCCAP_EMU_base> + 30 _H	8, 16, 32	
	Data 2-bit Error Status Clear Register	EM_BUSDDSTCLR	<BECCCAP_EMU_base> + 40 _H	8, 16, 32	
	Address 1-bit Error Status Register	EM_BUSASERSTR	<BECCCAP_EMU_base> + 50 _H	8, 16, 32	
	Address 2-bit Error Status Register	EM_BUSADERSTR	<BECCCAP_EMU_base> + 60 _H	8, 16, 32	
	Data 1-bit Error Status Register	EM_BUSDSERSTR	<BECCCAP_EMU_base> + 70 _H	8, 16, 32	
	Data 2-bit Error Status Register	EM_BUSDDERSTR	<BECCCAP_EMU_base> + 80 _H	8, 16, 32	
PBnECC (n = 0, 1, 2L, 2H, 3-9)	ECC Control Register	APECECCCTL	<PBnECC_base> + 00 _H	16, 32	APECKCPROT*1
	Error Notification Control Register	APECERRINT	<PBnECC_base> + 04 _H	8, 16, 32	APECKCPROT*1
	Error Status Clear Register	APECSTCLR	<PBnECC_base> + 08 _H	8, 16, 32	
	Error Status Register	APEC1STERSTR	<PBnECC_base> + 10 _H	8, 16, 32	
	ECC Test Control Register	APECTSTCTL	<PBnECC_base> + 150 _H	8, 16, 32	APECKCPROT*1
	Test Data Input Register	APECTSTDIN0	<PBnECC_base> + 154 _H	8, 16, 32	APECKCPROT*1
	Test ECC Input Register	APECTSTEIN	<PBnECC_base> + 158 _H	8, 16, 32	APECKCPROT*1
	Key Code Protection Register	APECKCPROT	<PBnECC_base> + 15C _H	32	
HBnECC (n = 91M/S, 93M/S, 95M/S, 96M)	ECC Control Register	HBECECCCTL	<HBnECC_base> + 00 _H	16, 32	HBECKCPROT
	Error Notification Control Register	HBECERRINT	<HBnECC_base> + 04 _H	8, 16, 32	HBECKCPROT
	Error Status Clear Register	HBECSTCLR	<HBnECC_base> + 08 _H	8, 16, 32	
	Error Status Register	HBEC1STERSTR	<HBnECC_base> + 10 _H	8, 16, 32	
	ECC Test Control Register	HBECTSTCTL	<HBnECC_base> + 150 _H	8, 16, 32	HBECKCPROT
	Test Data Input Register	HBECTSTDIN0	<HBnECC_base> + 154 _H	8, 16, 32	HBECKCPROT
	Test ECC Input Register	HBECTSTEIN	<HBnECC_base> + 158 _H	8, 16, 32	HBECKCPROT
	Key Code Protection Register	HBECKCPROT	<HBnECC_base> + 15C _H	32	
BECCCAP_V2An (n = 1-9)	Error Notification Control Register	PHC_BUSERRINT	<BECCCAP_V2An_base> + 00 _H	8, 16, 32	APECKCPROT*1
	Address 1-bit Error Status Clear Register	PHC_BUSASSTCLR	<BECCCAP_V2An_base> + 10 _H	8, 16, 32	
	Address 2-bit Error Status Clear Register	PHC_BUSADSTCLR	<BECCCAP_V2An_base> + 20 _H	8, 16, 32	
	Data 1-bit Error Status Clear Register	PHC_BUSDSSTCLR	<BECCCAP_V2An_base> + 30 _H	8, 16, 32	
	Data 2-bit Error Status Clear Register	PHC_BUSDDSTCLR	<BECCCAP_V2An_base> + 40 _H	8, 16, 32	
	Address 1-bit Error Status Register	PHC_BUSASERSTR	<BECCCAP_V2An_base> + 50 _H	8, 16, 32	
	Address 2-bit Error Status Register	PHC_BUSADERSTR	<BECCCAP_V2An_base> + 60 _H	8, 16, 32	
	Data 1-bit Error Status Register	PHC_BUSDSERSTR	<BECCCAP_V2An_base> + 70 _H	8, 16, 32	
	Data 2-bit Error Status Register	PHC_BUSDDERSTR	<BECCCAP_V2An_base> + 80 _H	8, 16, 32	

Note 1. For details of which peripheral group each module belongs to, see **Table 38.201, Register Base Addresses**.

Table 38.203 Relation between Data Transfer Path ECC and Modules (1/4)

Base Address Name	Master (from)	Slave (to)	Bus/access	Error Capture Target
<ECCCNT_A_CCIB0CLO_base>	All masters*1	Code Flash	Address	
<ECCCNT_A_CCIB1CLO_base>	All masters*1	Code Flash	Address	
<ECCCNT_SA_PE0CLO_base>	All Masters except for PE0*4	PE0	Address	
<ECCCNT_A_GCFU0ICLO_base>	PE0	Code Flash	Address (only address conversion on instruction)	
<ECCCNT_A_GCFU0DCL0_base>	PE0	Code Flash	Address (only address conversion on data access)	
<ECCCNT_SA_PE1CLO_base>	All Masters except for PE1*4	PE1	Address	
<ECCCNT_A_GCFU1ICLO_base>	PE1	Code Flash	Address (only address conversion on instruction)	
<ECCCNT_A_GCFU1DCL0_base>	PE1	Code Flash	Address (only address conversion on data access)	
<ECCCNT_A_X2VCLO_base>	Except PE	PE0/1, IPIR, MEV, BARR, TPTM	Address	
<ECCCNT_A_BARR_base>	All masters	BARR	Address	
<ECCCNT_A_IPIR_base>	All masters	IPIR	Address	
<ECCCNT_A_MEV_base>	All masters	MEV	Address	
<ECCCNT_A_TPTM_base>	All masters	TPTM	Address	
<ECCCNT_A_CRAM_base>	All masters	Cluster RAM	Address	
<ECCCNT_A_SG0_base>	All masters	P-Bus Group 0	Address (only double word access)	
<ECCCNT_A_SX2PV_base>	Except PE	All P-Bus Groups	Address	
<ECCCNT_A_SX2FX_base>	Except PE	Code Flash*2	Address	
<ECCCNT_A_FX2SX_base>	All masters	Cluster RAM	Address (for debug)	
<ECCCNT_A_GCFUF_base>	Except PE	Code Flash	Address (only address conversion)	
<ECCCNT_A_SX2MB_base>	All Masters except for PE*5	Cluster RAM	Address	
<ECCCNT_A_V2A1_base>	All masters	P-Bus Group 1	Address (only double word access)	
<ECCCNT_A_V2A2_base>	All masters	P-Bus Group 2	Address (only double word access)	
<ECCCNT_A_V2A3_base>	All masters	P-Bus Group 3	Address (only double word access)	
<ECCCNT_A_V2A4_base>	All masters	P-Bus Group 4	Address (only double word access)	
<ECCCNT_A_V2A5_base>	All masters	P-Bus Group 5	Address (only double word access)	
<ECCCNT_A_V2A6_base>	All masters	P-Bus Group 6	Address (only double word access)	
<ECCCNT_A_V2A7_base>	All masters	P-Bus Group 7	Address (only double word access)	
<ECCCNT_A_V2A8_base>	All masters	P-Bus Group 8	Address (only double word access)	
<ECCCNT_A_V2A9_base>	All masters	P-Bus Group 9	Address (only double word access)	
<ECCCNT_D_V2XWCL0_base>	PE0/1	H-Bus modules, Global/Cluster ERAM*7, Trace filter RAM*7,	Write data*3	
<ECCCNT_D_V2XRCL0_base>	PE0/1	H-Bus modules	Read data*3	
<ECCCNT_D_PV2APBW_base>	PE	P-Bus Group 0	Write data*3	
<ECCCNT_D_PV2APBR_base>	PE	P-Bus Group 0	Read data*3	
<ECCCNT_D_CRAM_base>	All masters	Cluster RAM	Write data (read-modify-write)	
<ECCCNT_D_EMU_base>	All masters	Cluster RAM	Read data (for debug)	
<ECCCNT_D_DMDE0_base>	sDMAC0	sDMAC0 internal buffer	Read data	
<ECCCNT_D_DMDE1_base>	sDMAC1	sDMAC1 internal buffer	Read data	
<ECCCNT_SD_PE0CLO_base>	Except PE0	PE0	Write data	

Table 38.203 Relation between Data Transfer Path ECC and Modules (2/4)

Base Address Name	Master (from)	Slave (to)	Bus/access	Error Capture Target
<ECCCNT_MD_PE0CL0_base>	PE0	All slaves except for Code Flash*6	Read data	
<ECCCNT_SD_PE1CL0_base>	Except PE1	PE1	Write data	
<ECCCNT_MD_PE1CL0_base>	PE1	All slaves except for Code Flash*6	Read data	
<ECCCNT_D_BARR_base>	All masters	BARR	Write data	
<ECCCNT_D_IPIR_base>	All masters	IPIR	Write data	
<ECCCNT_D_MEV_base>	All masters	MEV	Write data	
<ECCCNT_D_TPTM_base>	All masters	TPTM	Write data	
<ECCCNT_D_DTS_base>	DTS	All slaves	Read data(byte access and half word access)	
<ECCCNT_D_DMA0_base>	sDMAC0	All slaves	Read data	
<ECCCNT_D_DMA1_base>	sDMAC1	All slaves	Read data	
<ECCCNT_D_V2A1W_base>	PE	P-Bus Group 1	Write data (read-modify-write)*3	
<ECCCNT_D_V2A2W_base>	PE	P-Bus Group 2	Write data (read-modify-write)*3	
<ECCCNT_D_V2A3W_base>	PE	P-Bus Group 3	Write data (read-modify-write)*3	
<ECCCNT_D_V2A4W_base>	PE	P-Bus Group 4	Write data (read-modify-write)*3	
<ECCCNT_D_V2A5W_base>	PE	P-Bus Group 5	Write data (read-modify-write)*3	
<ECCCNT_D_V2A6W_base>	PE	P-Bus Group 6	Write data (read-modify-write)*3	
<ECCCNT_D_V2A7W_base>	PE	P-Bus Group 7	Write data (read-modify-write)*3	
<ECCCNT_D_V2A8W_base>	PE	P-Bus Group 8	Write data (read-modify-write)*3	
<ECCCNT_D_V2A9W_base>	PE	P-Bus Group 9	Write data (read-modify-write)*3	
<ECCCNT_D_V2A1R_base>	PE	P-Bus Group 1	Read data*3	
<ECCCNT_D_V2A2R_base>	PE	P-Bus Group 2	Read data*3	
<ECCCNT_D_V2A3R_base>	PE	P-Bus Group 3	Read data*3	
<ECCCNT_D_V2A4R_base>	PE	P-Bus Group 4	Read data*3	
<ECCCNT_D_V2A5R_base>	PE	P-Bus Group 5	Read data*3	
<ECCCNT_D_V2A6R_base>	PE	P-Bus Group 6	Read data*3	
<ECCCNT_D_V2A7R_base>	PE	P-Bus Group 7	Read data*3	
<ECCCNT_D_V2A8R_base>	PE	P-Bus Group 8	Read data*3	
<ECCCNT_D_V2A9R_base>	PE	P-Bus Group 9	Read data*3	
<BECCCAP_LRAM_base>	Error capture module related Local RAM			ECCCNT_SA_PEnCL0 (n = 0, 1) ECCCNT_SD_PEnCL0 (n = 0, 1) ECCCNT_A_X2VCL0
<BECCCAP_CRAM_base>	Error capture module related Cluster RAM			ECCCNT_A_CRAM ECCCNT_A_SX2MB ECCCNT_D_CRAM
<BECCCAP_CFL_base>	Error capture module related Code Flash			ECCCNT_A_CCIBnCL0 (n = 0 to 1) ECCCNT_A_SX2FX

Table 38.203 Relation between Data Transfer Path ECC and Modules (3/4)

Base Address Name	Master (from)	Slave (to)	Bus/access	Error Capture Target
<BECCAP_PERI_base>	Error capture module related P-Bus Group 0			ECCCNT_A_BARR ECCCNT_A_IPIR ECCCNT_A_MEV ECCCNT_A_TPTM ECCCNT_A_SG0 ECCCNT_A_SX2PV ECCCNT_MD_PEnCL0 (n = 0, 1) ECCCNT_D_V2XWCL0 ECCCNT_D_V2XRCL0 ECCCNT_D_PV2APBW ECCCNT_D_PV2APBR ECCCNT_D_BARR ECCCNT_D_IPIR ECCCNT_D_MEV ECCCNT_D_TPTM
<BECCAP_DMDT_base>	Error capture module related DTS/sDMAC			ECCCNT_D_DMDEn (n = 0, 1) ECCCNT_D_DTS ECCCNT_D_DMAAn (n = 0, 1)
<BECCAP_EMU_base>	Error capture module related Cluster RAM (for debug)			ECCCNT_A_FX2SX ECCCNT_A_GCFUF ECCCNT_D_EMU ECCCNT_A_GCFUnICL0 (n = 0, 1) ECCCNT_A_GCFUnDCL0 (n = 0, 1)
<PB0ECC_base>	All masters	P-Bus group 0	Address and Write data	Including Error capture module related P-Bus groups
<PB1ECC_base>	All masters	P-Bus group 1	Address and Write data	Including Error capture module related P-Bus groups
<PB2LECC_base>	All masters	P-Bus group 2L	Address and Write data	Including Error capture module related P-Bus groups
<PB2HECC_base>	All masters	P-Bus group 2H	Address and Write data	Including Error capture module related P-Bus groups
<PB3ECC_base>	All masters	P-Bus group 3	Address and Write data	Including Error capture module related P-Bus groups
<PB4ECC_base>	All masters	P-Bus group 4	Address and Write data	Including Error capture module related P-Bus groups
<PB5ECC_base>	All masters	P-Bus group 5	Address and Write data	Including Error capture module related P-Bus groups
<PB6ECC_base>	All masters	P-Bus group 6	Address and Write data	Including Error capture module related P-Bus groups
<PB7ECC_base>	All masters	P-Bus group 7	Address and Write data	Including Error capture module related P-Bus groups
<PB8ECC_base>	All masters	P-Bus group 8	Address and Write data	Including Error capture module related P-Bus groups
<PB9ECC_base>	All masters	P-Bus group 9	Address and Write data	Including Error capture module related P-Bus groups
<HB91MECC_base>	RHSIF0	All slaves	Read data	
<HB91SECC_base>	All masters	RHSIF0	Address and Write data	
<HB93MECC_base>	FlexRay0	All slaves	Read data	
<HB93SECC_base>	All masters	FlexRay0	Address and Write data	
<HB95MECC_base>	Ethernet	All slaves	Read data	
<HB95SECC_base>	All masters	Ethernet	Address and Write data	
<HB96MECC_base>	HSSPI0	All slaves	Read data	
<BECCAP_V2A1_base>	Error capture module related P-Bus group 1			ECCCNT_A_V2A1 ECCCNT_D_V2A1W ECCCNT_D_V2A1R
<BECCAP_V2A2_base>	Error capture module related P-Bus group 2			ECCCNT_A_V2A2 ECCCNT_D_V2A2W ECCCNT_D_V2A2R
<BECCAP_V2A3_base>	Error capture module related P-Bus group 3			ECCCNT_A_V2A3 ECCCNT_D_V2A3W ECCCNT_D_V2A3R

Table 38.203 Relation between Data Transfer Path ECC and Modules (4/4)

Base Address Name	Master (from)	Slave (to)	Bus/access	Error Capture Target
<BECCAP_V2A4_base>	Error capture module related P-Bus group 4			ECCCNT_A_V2A4 ECCCNT_D_V2A4W ECCCNT_D_V2A4R
<BECCAP_V2A5_base>	Error capture module related P-Bus group 5			ECCCNT_A_V2A5 ECCCNT_D_V2A5W ECCCNT_D_V2A5R
<BECCAP_V2A6_base>	Error capture module related P-Bus group 6			ECCCNT_A_V2A6 ECCCNT_D_V2A6W ECCCNT_D_V2A6R
<BECCAP_V2A7_base>	Error capture module related P-Bus group 7			ECCCNT_A_V2A7 ECCCNT_D_V2A7W ECCCNT_D_V2A7R
<BECCAP_V2A8_base>	Error capture module related P-Bus group 8			ECCCNT_A_V2A8 ECCCNT_D_V2A8W ECCCNT_D_V2A8R
<BECCAP_V2A9_base>	Error capture module related P-Bus group 9			ECCCNT_A_V2A9 ECCCNT_D_V2A9W ECCCNT_D_V2A9R

Note 1. For an access by using a master except for PE, the path from Global FLASH Bus to Code Flash is covered.

Note 2. For an access by using a master except for PE, the path from a master to Global FLASH Bus is covered.

Note 3. An access by specific instructions (CLR1, NOT1, SET1, and CAXI) is targeted. For details, see *the RH850G4MH User's Manual: Software*.

Note 4. Instruction fetch access from PEn to PEn's Local RAM is covered. (n = 0 to 1)

Note 5. Instruction fetch access from PEn to Cluster RAM is covered.

Note 6. Data access from PEn to Cluster RAM and PEn's Local RAM is not covered. (n = 0 to 1)

Note 7. This function is implemented in E2x-FCC1 only.

38.3.11.4 BUSAECCCTL — ECC Control Register

This register controls ECC error detection for requested address.

This register is protected by the Key Code Protection Register. For details whether this register is protected by either KCPROT or APECKCPROT, see **Table 38.202, List of Registers**.

This register can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 38.204 BUSAECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ECCDIS	Address ECC disable bit Sets ECC error detection to enable/disable. 0: ECC error detection is enabled 1: ECC error detection is disabled

38.3.11.5 BUSAECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), arbitrary data value can be input to ECC decoder.

This register is protected by the Key Code Protection Register. For details whether this register is protected by either KCPROT or APECKCPROT, see **Table 38.202, List of Registers**.

This register can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 38.205 BUSAECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for the address ECC decoder. 0: Normal mode 1: Test mode The values of BUSAECCTSTDINO and BUSAECCTSTEIN registers are provided to data and ECC fields of the ECC decoder in test mode.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.11.6 BUSAECCTSTDIN0 — Test Data Input Register

This register is used to inject errors into data field of ECC decoder for self-diagnosis. The values of BUSAECCTSTDIN0 and BUSAECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the Key Code Protection Register. For details whether this register is protected by either KCPROT or APECKCPROT, see **Table 38.202, List of Registers**.

This register can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.206 BUSAECCTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify test data for the data ECC decoder. The data is provided to data field of the ECC decoder when ECCTST = 1.

38.3.11.7 BUSAECCTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of ECC decoder for self-diagnosis. The values of BUSAECCTSTDIN0 and BUSAECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the Key Code Protection Register. For details whether this register is protected by either KCPROT or APECKCPROT, see **Table 38.202, List of Registers**.

This register can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.207 BUSAECCTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECC[6:0]	Specify test data for ECC decoder. The data is provided to ECC field of the ECC decoder when ECCTST = 1.

38.3.11.8 BUSDVCECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction.

This register is protected by the Key Code Protection Register. For details whether this register is protected by either KCPROT or APECKCPROT, see **Table 38.202, List of Registers**.

This register can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.208 BUSDVCECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	Data ECC error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when a 1-bit error is detected
0	ECCDIS	Data ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

38.3.11.9 BUSDVCECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), arbitrary data value can be input to ECC decoder.

This register is protected by the Key Code Protection Register. For details whether this register is protected by either KCPROT or APECKCPROT, see **Table 38.202, List of Registers**.

This register can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	DAT SEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.209 BUSDVCECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for the data ECC decoder. 0: Normal mode 1: Test mode The values of BUSDVCECCTSTDIN0 and BUSDVCECCTSTEIN registers are provided to data and ECC fields of one of the ECC decoders along with DATSEL setting in test mode.
0	DATSEL	Selects ECC decoder to replace input data when ECCTST = 1. 0: ECC decoder for lower data 1: ECC decoder for upper data

38.3.11.10 BUSDVCECCTSTDIN0 — Test Data Input Register

This register is used to inject errors into data field of ECC decoder for self-diagnosis. The values of BUSDVCECCTSTDIN0 and BUSDVCECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the Key Code Protection Register. For details whether this register is protected by either KCPROT or APECKCPROT, see **Table 38.202, List of Registers**.

This register can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.210 BUSDVCECCTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for the data ECC decoder. The data is provided to data field of the ECC decoder when ECCTST = 1.

38.3.11.11 BUSDVCECCTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of ECC decoder for self-diagnosis. The values of BUSDVCECCTSTDIN0 and BUSDVCECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the Key Code Protection Register. For details whether this register is protected by either KCPROT or APECKCPROT, see **Table 38.202, List of Registers**.

This register can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.211 BUSDVCECCTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECC[6:0]	Specify the test data for ECC decoder. The data is provided to ECC field of the ECC decoder when ECCTST = 1.

38.3.11.12 BUSDV1ECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for data read to VCI2APB bridge.

This register is protected by the Key Code Protection Register. For details whether this register is protected by either KCPROT or APECKCPROT, see **Table 38.202, List of Registers**.

This register can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.212 BUSDV1ECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	Data ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	Data ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

38.3.11.13 BUSDV1ECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), arbitrary data value can be input to ECC decoder.

This register is protected by the Key Code Protection Register. For details whether this register is protected by either KCPROT or APECKCPROT, see **Table 38.202, List of Registers**.

This register can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 38.213 BUSDV1ECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for the data ECC decoder. 0: Normal mode 1: Test mode The values of BUSDV1ECCTSTDIN0 and BUSDV1ECCTSTEIN registers are provided to data and ECC fields of the ECC decoder in test mode.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.11.14 BUSDV1ECCTSTDIN0 — Test Data Input Register

This register is used to inject errors into data field of ECC decoder for self-diagnosis. The values of BUSDV1ECCTSTDIN0 and BUSDV1ECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the Key Code Protection Register. For details whether this register is protected by either KCPROT or APECKCPROT, see **Table 38.202, List of Registers**.

This register can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.214 BUSDV1ECCTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for the data ECC decoder. The data is provided to data field of the ECC decoder when ECCTST = 1.

38.3.11.15 BUSDV1ECCTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of ECC decoder for self-diagnosis. The values of BUSDV1ECCTSTDIN0 and BUSDV1ECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the Key Code Protection Register. For details whether this register is protected by either KCPROT or APECKCPROT, see **Table 38.202, List of Registers**.

This register can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.215 BUSDV1ECCTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECC[6:0]	Specify the test data for ECC decoder. The data is provided to ECC field of the ECC decoder when ECCTST = 1.

38.3.11.16 BUSDEMECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for data read to FAXI2SAXI bridge.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.216 BUSDEMECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	Data ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	Data ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

38.3.11.17 BUSDEMECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After ECC test mode (ECCTST = 1) setting, arbitrary data value can be input to ECC decoder.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	DAT SEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.217 BUSDEMECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for the data ECC decoder. 0: Normal mode 1: Test mode The values of BUSDEMECCTSTDIN0 and BUSDEMECCTSTEIN registers are provided to data and ECC field of one of the ECC decoders along with DATSEL setting in test mode.
0	DATSEL	Selects ECC decoder to replace input data under ECCTST = 1. 0: ECC decoder for lower data 1: ECC decoder for upper data

38.3.11.18 BUSDEMECCTSTDIN0 — Test Data Input Register

This register is used to inject errors into data field of ECC decoder for self-diagnosis. The values of BUSDEMECCTSTDIN0 and BUSDEMECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.218 BUSDEMECCTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for the data ECC decoder. The data is provided to data field of the ECC decoder when ECCTST = 1.

38.3.11.19 BUSDEMECCTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of ECC decoder for self-diagnosis. The values of BUSDEMECCTSTDIN0 and BUSDEMECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.219 BUSDEMECCTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECC[6:0]	Specify the test data for ECC decoder. The data is provided to ECC field of the ECC decoder when ECCTST = 1.

38.3.11.20 BUSDDEECCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for internal buffer of sDMAC.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.220 BUSDDEECCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	Data ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	Data ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

38.3.11.21 BUSDDEECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), arbitrary data value can be input to ECC decoder.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 38.221 BUSDDEECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for the data ECC decoder. 0: Normal mode 1: Test mode The values of BUSDDEECCTSTDIN0, BUSDDEECCTSTDIN1 and BUSDDEECCTSTEIN registers are provided to data and ECC fields of the ECC decoder in test mode.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.11.22 BUSDDEECCTSTDIN0 — Test Data Input Register

This register is used to inject errors into lower data field of ECC decoder for self-diagnosis. The values of BUSDDEECCTSTDIN0, BUSDDEECCTSTDIN1 and BUSDDEECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.222 BUSDDEECCTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for the data ECC decoder. The data is provided to lower data field of the ECC decoder when ECCTST = 1.

38.3.11.23 BUSDDEECCTSTDIN1 — Test Data Input Register

This register is used to inject errors into upper data field of ECC decoder for self-diagnosis. The values of BUSDDEECCTSTDIN0, BUSDDEECCTSTDIN1 and BUSDDEECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.223 BUSDDEECCTSTDIN1 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for the data ECC decoder. The data is provided to upper data field of the ECC decoder when ECCTST = 1.

38.3.11.24 BUSDDEECCTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of ECC decoder for self-diagnosis. The values of BUSDDEECCTSTDIN0, BUSDDEECCTSTDIN1 and BUSDDEECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ECC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.224 BUSDDEECCTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	ECC[7:0]	Specify the test data for ECC decoder. The data is provided to ECC field of the ECC decoder when ECCTST = 1.

38.3.11.25 BUSDECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.225 BUSDECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	Data ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	Data ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

38.3.11.26 LR_BUSERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address/data ECC 2-bit error and address/data ECC 1-bit error are detected.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 000F_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADED IE	ASED IE	DDED IE	DSED IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.226 LR_BUSERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	ADEDIE	Controls error reports when address ECC 2-bit error is detected. 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled
2	ASEDIE	Controls error reports when address ECC 1-bit error is detected. 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled
1	DDEDIE	Controls error reports when data ECC 2-bit error is detected. 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled
0	DSEDIE	Controls error reports when data ECC 1-bit error is detected. 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled

38.3.11.27 LR_BUSASSTCLR — Address 1-bit Error Status Clear Register

This register is used to clear error flags in LR_BUSASERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASSTC LR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASSTC LR01	ASSTC LR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.227 LR_BUSASSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	ASSTCLR16	Writing 1 to this bit clears ASED16 in LR_BUSASERSTR.
15 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ASSTCLR01	Writing 1 to this bit clears ASED01 in LR_BUSASERSTR.
0	ASSTCLR00	Writing 1 to this bit clears ASED00 in LR_BUSASERSTR.

38.3.11.28 LR_BUSADSTCLR — Address 2-bit Error Status Clear Register

This register is used to clear error flags in LR_BUSADERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADSTC LR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADSTC LR01	ADSTC LR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.228 LR_BUSADSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	ADSTCLR16	Writing 1 to this bit clears ADEDF16 in LR_BUSADERSTR.
15 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ADSTCLR01	Writing 1 to this bit clears ADEDF01 in LR_BUSADERSTR.
0	ADSTCLR00	Writing 1 to this bit clears ADEDF00 in LR_BUSADERSTR.

38.3.11.29 LR_BUSDSSTCLR — Data 1-bit Error Status Clear Register

This register is used to clear error flags in LR_BUSDSESTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSSTC LR01	DSSTC LR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.229 LR_BUSDSSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DSSTCLR01	Writing 1 to this bit clears DSEDF01 in LR_BUSDSESTR.
0	DSSTCLR00	Writing 1 to this bit clears DSEDF00 in LR_BUSDSESTR.

38.3.11.30 LR_BUSDDSTCLR — Data 2-bit Error Status Clear Register

This register is used to clear error flags in LR_BUSDDERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDSTC LR01	DDSTC LR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.230 LR_BUSDDSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DDSTCLR01	Writing 1 to this bit clears DDED01 in LR_BUSDDERSTR.
0	DDSTCLR00	Writing 1 to this bit clears DDED00 in LR_BUSDDERSTR.

38.3.11.31 LR_BUSASERSTR — Address 1-bit Error Status Register

This register indicates that an address ECC 1-bit error has occurred in a path related to Local RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, flags of the detected errors are all set.

This register can be cleared by LR_BUSASSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASEDF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASEDF01	ASEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.231 LR_BUSASERSTR Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	ASEDF16	Indicates that an address ECC 1-bit error was detected in request address to AXI2VCI bridge in cluster 0.
15 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ASEDF01	Indicates that an address ECC 1-bit error was detected in request address to PE1.
0	ASEDF00	Indicates that an address ECC 1-bit error was detected in request address to PE0.

38.3.11.32 LR_BUSADERSTR — Address 2-bit Error Status Register

This register indicates that an address ECC 2-bit error has occurred in a path related to Local RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected errors are all set.

This register can be cleared by LR_BUSADSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADEDF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADEDF01	ADEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.232 LR_BUSADERSTR Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	ADEDF16	Indicates that an address ECC 2-bit error was detected in request address to AXI2VCI bridge in cluster 0.
15 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ADEDF01	Indicates that an address ECC 2-bit error was detected in request address to PE1.
0	ADEDF00	Indicates that an address ECC 2-bit error was detected in request address to PE0.

38.3.11.33 LR_BUSDSESTR — Data 1-bit Error Status Register

This register indicates that a data ECC 1-bit error has occurred in a path related to Local RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected errors are all set.

This register can be cleared by LR_BUSDSESTR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSEDF01	DSEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.233 LR_BUSDSESTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DSEDF01	Indicates that a data ECC 1-bit error was detected in write data to PE1.
0	DSEDF00	Indicates that a data ECC 1-bit error was detected in write data to PE0.

38.3.11.34 LR_BUSDDERSTR — Data 2-bit Error Status Register

This register indicates that a data ECC 2-bit error has occurred in a path related to Local RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected errors are all set.

This register can be cleared by LR_BUSDDSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDED F01	DDED F00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.234 LR_BUSDDERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DDED01	Indicates that a data ECC 2-bit error has been detected in write data to PE1.
0	DDED00	Indicates that a data ECC 2-bit error has been detected in write data to PE0.

38.3.11.35 CR_BUSERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address/data ECC 2-bit error and address/data ECC 1-bit error are detected.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 000F_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADED IE	ASED IE	DDED IE	DSED IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.235 CR_BUSERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	ADEDIE	Controls error reports when address ECC 2-bit error is detected. 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled
2	ASEDIE	Controls error reports when address ECC 1-bit error is detected. 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled
1	DDEDIE	Controls error reports when data ECC 2-bit error is detected. 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled
0	DSEDIE	Controls error reports when data ECC 1-bit error is detected. 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled

38.3.11.36 CR_BUSASSTCLR — Address 1-bit Error Status Clear Register

This register is used to clear error flags in CR_BUSASERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ASSTCLR25	ASSTCLR24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ASSTCLR06	—	—	—	ASSTCLR02	ASSTCLR01	ASSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	W	R	R	R	W	W	W

Table 38.236 CR_BUSASSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25	ASSTCLR25	Writing 1 to this bit clears ASED25 in CR_BUSASERSTR.
24	ASSTCLR24	Writing 1 to this bit clears ASED24 in CR_BUSASERSTR.
23 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	ASSTCLR06	Writing 1 to this bit clears ASED06 in CR_BUSASERSTR.
5 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	ASSTCLR02	Writing 1 to this bit clears ASED02 in CR_BUSASERSTR.
1	ASSTCLR01	Writing 1 to this bit clears ASED01 in CR_BUSASERSTR.
0	ASSTCLR00	Writing 1 to this bit clears ASED00 in CR_BUSASERSTR.

38.3.11.37 CR_BUSADSTCLR — Address 2-bit Error Status Clear Register

This register is used to clear error flags in CR_BUSADERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ADSTC LR25	ADSTC LR24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ADSTC LR06	—	—	—	ADSTC LR02	ADSTC LR01	ADSTC LR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	W	R	R	R	W	W	W

Table 38.237 CR_BUSADSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25	ADSTCLR25	Writing 1 to this bit clears ADEDF25 in CR_BUSADERSTR.
24	ADSTCLR24	Writing 1 to this bit clears ADEDF24 in CR_BUSADERSTR.
23 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	ADSTCLR06	Writing 1 to this bit clears ADEDF06 in CR_BUSADERSTR.
5 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	ADSTCLR02	Writing 1 to this bit clears ADEDF02 in CR_BUSADERSTR.
1	ADSTCLR01	Writing 1 to this bit clears ADEDF01 in CR_BUSADERSTR.
0	ADSTCLR00	Writing 1 to this bit clears ADEDF00 in CR_BUSADERSTR.

38.3.11.38 CR_BUSDSSTCLR — Data 1-bit Error Status Clear Register

This register is used to clear error flags in CR_BUSDSERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	DSSTC LR06	—	—	—	DSSTC LR02	DSSTC LR01	DSSTC LR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	W	R	R	R	W	W	W

Table 38.238 CR_BUSDSSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	DSSTCLR06	Writing 1 to this bit clears DSEDF06 in CR_BUSDSERSTR.
5 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	DSSTCLR02	Writing 1 to this bit clears DSEDF02 in CR_BUSDSERSTR.
1	DSSTCLR01	Writing 1 to this bit clears DSEDF01 in CR_BUSDSERSTR.
0	DSSTCLR00	Writing 1 to this bit clears DSEDF00 in CR_BUSDSERSTR.

38.3.11.39 CR_BUSDDSTCLR — Data 2-bit Error Status Clear Register

This register is used to clear error flags in CR_BUSDDERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	DDSTC LR06	—	—	—	DDSTC LR02	DDSTC LR01	DDSTC LR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	W	R	R	R	W	W	W

Table 38.239 CR_BUSDDSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	DDSTCLR06	Writing 1 to this bit clears DDED06 in CR_BUSDDERSTR.
5 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	DDSTCLR02	Writing 1 to this bit clears DDED02 in CR_BUSDDERSTR.
1	DDSTCLR01	Writing 1 to this bit clears DDED01 in CR_BUSDDERSTR.
0	DDSTCLR00	Writing 1 to this bit clears DDED00 in CR_BUSDDERSTR.

38.3.11.40 CR_BUSASERSTR — Address 1-bit Error Status Register

This register indicates that an address ECC 1-bit error has occurred in a path related to Cluster RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected errors are all set.

This register can be cleared by CR_BUSASSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ASEDF 25	ASEDF 24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ASEDF 06	—	—	—	ASEDF 02	ASEDF 01	ASEDF 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.240 CR_BUSASERSTR Register Contents

Bit Position	Bit Name	Function
31 to 26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25	ASEDF25	Indicates that an address ECC 1-bit error was detected in request address to SAXI2MBI bridge for cluster 1.
24	ASEDF24	Indicates that an address ECC 1-bit error was detected in request address to SAXI2MBI bridge for cluster 0.
23 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	ASEDF06	Indicates that an address ECC 1-bit error was detected in request address to Cluster RAM in cluster 1 from outside.
5 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	ASEDF02	Indicates that an address ECC 1-bit error was detected in request address to Cluster RAM in cluster 0 from outside.
1	ASEDF01	Indicates that an address ECC 1-bit error was detected in request address from PE1 to Cluster RAM in cluster 0.
0	ASEDF00	Indicates that an address ECC 1-bit error was detected in request address from PE0 to Cluster RAM in cluster 0.

38.3.11.41 CR_BUSADERSTR — Address 2-bit Error Status Register

This register indicates that an address ECC 2-bit error has occurred in a path related to Cluster RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected errors are all set.

This register can be cleared by CR_BUSADSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ADEDF 25	ADEDF 24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ADEDF 06	—	—	—	ADEDF 02	ADEDF 01	ADEDF 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.241 CR_BUSADERSTR Register Contents

Bit Position	Bit Name	Function
31 to 26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25	ADEDF25	Indicates that an address ECC 2-bit error was detected in request address to SAXI2MBI bridge for cluster 1.
24	ADEDF24	Indicates that an address ECC 2-bit error was detected in request address to SAXI2MBI bridge for cluster 0.
23 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	ADEDF06	Indicates that an address ECC 2-bit error was detected in request address to Cluster RAM in cluster 1 from outside.
5 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	ADEDF02	Indicates that an address ECC 2-bit error was detected in request address to Cluster RAM in cluster 0 from outside.
1	ADEDF01	Indicates that an address ECC 2-bit error was detected in request address from PE1 to Cluster RAM in cluster 0.
0	ADEDF00	Indicates that an address ECC 2-bit error was detected in request address from PE0 to Cluster RAM in cluster 0.

38.3.11.42 CR_BUSDSESTR — Data 1-bit Error Status Register

This register indicates that a data ECC 1-bit error has occurred in a path related to Cluster RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected errors are all set.

This register can be cleared by CR_BUSDSESTR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	DSEDF06	—	—	—	DSEDF02	DSEDF01	DSEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.242 CR_BUSDSESTR Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	DSEDF06	Indicates that a data ECC 1-bit error was detected in write data to Cluster RAM in cluster 1 from outside.
5 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	DSEDF02	Indicates that a data ECC 1-bit error was detected in write data to Cluster RAM in cluster 0 from outside.
1	DSEDF01	Indicates that a data ECC 1-bit error was detected in write data from PE1 to Cluster RAM in cluster 0.
0	DSEDF00	Indicates that a data ECC 1-bit error was detected in write data from PE0 to Cluster RAM in cluster 0.

38.3.11.43 CR_BUSDDERSTR — Data 2-bit Error Status Register

This register indicates that a data ECC 2-bit error has occurred in a path related to Cluster RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected errors are all set.

This register can be cleared by CR_BUSDDSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	DDED F06	—	—	—	DDED F02	DDED F01	DDED F00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.243 CR_BUSDDERSTR Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	DDED F06	Indicates that a data ECC 2-bit error was detected in write data to Cluster RAM in cluster 1 from outside.
5 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	DDED F02	Indicates that a data ECC 2-bit error was detected in write data to Cluster RAM in cluster 0 from outside.
1	DDED F01	Indicates that a data ECC 2-bit error was detected in write data from PE1 to Cluster RAM in cluster 0.
0	DDED F00	Indicates that a data ECC 2-bit error was detected in write data from PE0 to Cluster RAM in cluster 0.

38.3.11.44 CF_BUSERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address ECC 2-bit error and address ECC 1-bit error are detected.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 000CH

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADED IE	ASED IE	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R

Table 38.244 CF_BUSERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	ADEDIE	Controls error reports when address ECC 2-bit error is detected. 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled
2	ASEDIE	Controls error reports when address ECC 1-bit error is detected. 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled
1, 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.11.45 CF_BUSASSTCLR — Address 1-bit Error Status Clear Register

This register is used to clear error flags in CF_BUSASERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	ASSTCLR24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASSTCLR01	ASSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.245 CF_BUSASSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 25	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
24	ASSTCLR24	Writing 1 to this bit clears ASED24 in CF_BUSASERSTR.
23 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ASSTCLR01	Writing 1 to this bit clears ASED01 in CF_BUSASERSTR.
0	ASSTCLR00	Writing 1 to this bit clears ASED00 in CF_BUSASERSTR.

38.3.11.46 CF_BUSADSTCLR — Address 2-bit Error Status Clear Register

This register is used to clear error flags in CF_BUSADERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	ADSTC LR24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADSTC LR01	ADSTC LR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.246 CF_BUSADSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 25	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
24	ADSTCLR24	Writing 1 to this bit clears ADEDF24 in CF_BUSADERSTR.
23 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ADSTCLR01	Writing 1 to this bit clears ADEDF01 in CF_BUSADERSTR.
0	ADSTCLR00	Writing 1 to this bit clears ADEDF00 in CF_BUSADERSTR.

38.3.11.47 CF_BUSASERSTR — Address 1-bit Error Status Register

This register indicates that an address ECC 1-bit error has occurred in a path related to Code Flash. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected error are all set.

This register can be cleared by CF_BUSASSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	ASEDF 24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASEDF 01	ASEDF 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.247 CF_BUSASERSTR Register Contents

Bit Position	Bit Name	Function
31 to 25	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
24	ASEDF24	Indicates that an address ECC 1-bit error was detected in request address to SAXI2FAXI bridge.
23 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ASEDF01	Indicates that an address ECC 1-bit error was detected in request address to Code Flash bank B in cluster 0.
0	ASEDF00	Indicates that an address ECC 1-bit error was detected in request address to Code Flash bank A in cluster 0.

38.3.11.48 CF_BUSADERSTR — Address 2-bit Error Status Register

This register indicates that an address ECC 2-bit error has occurred in a path related to Code Flash. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by CF_BUSADSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	ADEDF 24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADEDF 01	ADEDF 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.248 CF_BUSADERSTR Register Contents

Bit Position	Bit Name	Function
31 to 25	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
24	ADEDF24	Indicates that an address ECC 2-bit error was detected in request address to SAXI2FAXI bridge.
23 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ADEDF01	Indicates that an address ECC 2-bit error was detected in request address to Code Flash bank B in cluster 0.
0	ADEDF00	Indicates that an address ECC 2-bit error was detected in request address to Code Flash bank A in cluster 0.

38.3.11.49 PH_BUSERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address/data ECC 2-bit error and address/data ECC 1-bit error are detected.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 000F_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADED IE	ASED IE	DDED IE	DSED IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.249 PH_BUSERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	ADEDIE	Controls error reports when address ECC 2-bit error was detected. 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled
2	ASEDIE	Controls error reports when address ECC 1-bit error was detected. 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled
1	DDEDIE	Controls error reports when data ECC 2-bit error was detected. 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled
0	DSEDIE	Controls error reports when data ECC 1-bit error was detected. 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled

38.3.11.50 PH_BUSASSTCLR — Address 1-bit Error Status Clear Register

This register is used to clear error flags in PH_BUSASERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ASSTCLR25	—	—	—	—	ASSTCLR20	ASSTCLR19	ASSTCLR18	ASSTCLR17	ASSTCLR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	R	R	R	R	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.250 PH_BUSASSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25	ASSTCLR25	Writing 1 to this bit clears ASED25 in PH_BUSASERSTR.
24 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	ASSTCLR20	Writing 1 to this bit clears ASED20 in PH_BUSASERSTR.
19	ASSTCLR19	Writing 1 to this bit clears ASED19 in PH_BUSASERSTR.
18	ASSTCLR18	Writing 1 to this bit clears ASED18 in PH_BUSASERSTR.
17	ASSTCLR17	Writing 1 to this bit clears ASED17 in PH_BUSASERSTR.
16	ASSTCLR16	Writing 1 to this bit clears ASED16 in PH_BUSASERSTR.
15 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.11.51 PH_BUSADSTCLR — Address 2-bit Error Status Clear Register

This register is used to clear error flags in PH_BUSADERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ADSTCLR25	—	—	—	—	ADSTCLR20	ADSTCLR19	ADSTCLR18	ADSTCLR17	ADSTCLR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	R	R	R	R	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.251 PH_BUSADSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25	ADSTCLR25	Writing 1 to this bit clears ADEDF25 in PH_BUSADERSTR.
24 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	ADSTCLR20	Writing 1 to this bit clears ADEDF20 in PH_BUSADERSTR.
19	ADSTCLR19	Writing 1 to this bit clears ADEDF19 in PH_BUSADERSTR.
18	ADSTCLR18	Writing 1 to this bit clears ADEDF18 in PH_BUSADERSTR.
17	ADSTCLR17	Writing 1 to this bit clears ADEDF17 in PH_BUSADERSTR.
16	ADSTCLR16	Writing 1 to this bit clears ADEDF16 in PH_BUSADERSTR.
15 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.11.52 PH_BUSDSSTCLR — Data 1-bit Error Status Clear Register

This register is used to clear error flags in PH_BUSDSESTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	DSSTCLR26	DSSTCLR25	—	—	—	—	—	DSSTCLR19	DSSTCLR18	DSSTCLR17	DSSTCLR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	W	W	R	R	R	R	R	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DSSTCLR09	DSSTCLR08	—	—	—	—	—	—	DSSTCLR01	DSSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W	R	R	R	R	R	R	W	W

Table 38.252 PH_BUSDSSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 27	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
26	DSSTCLR26	Writing 1 to this bit clears DSEDF26 in PH_BUSDSESTR.
25	DSSTCLR25	Writing 1 to this bit clears DSEDF25 in PH_BUSDSESTR.
24 to 20	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
19	DSSTCLR19	Writing 1 to this bit clears DSEDF19 in PH_BUSDSESTR.
18	DSSTCLR18	Writing 1 to this bit clears DSEDF18 in PH_BUSDSESTR.
17	DSSTCLR17	Writing 1 to this bit clears DSEDF17 in PH_BUSDSESTR.
16	DSSTCLR16	Writing 1 to this bit clears DSEDF16 in PH_BUSDSESTR.
15 to 10	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
9	DSSTCLR09	Writing 1 to this bit clears DSEDF09 in PH_BUSDSESTR.
8	DSSTCLR08	Writing 1 to this bit clears DSEDF08 in PH_BUSDSESTR.
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DSSTCLR01	Writing 1 to this bit clears DSEDF01 in PH_BUSDSESTR.
0	DSSTCLR00	Writing 1 to this bit clears DSEDF00 in PH_BUSDSESTR.

38.3.11.53 PH_BUSDDSTCLR — Data 2-bit Error Status Clear Register

This register is used to clear error flags in PH_BUSDDERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	DDSTC LR26	DDSTC LR25	—	—	—	—	—	DDSTC LR19	DDSTC LR18	DDSTC LR17	DDSTC LR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	W	W	R	R	R	R	R	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DDSTC LR09	DDSTC LR08	—	—	—	—	—	—	DDSTC LR01	DDSTC LR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W	R	R	R	R	R	R	W	W

Table 38.253 PH_BUSDDSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 27	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
26	DDSTCLR26	Writing 1 to this bit clears DDED26 in PH_BUSDDERSTR.
25	DDSTCLR25	Writing 1 to this bit clears DDED25 in PH_BUSDDERSTR.
24 to 20	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
19	DDSTCLR19	Writing 1 to this bit clears DDED19 in PH_BUSDDERSTR.
18	DDSTCLR18	Writing 1 to this bit clears DDED18 in PH_BUSDDERSTR.
17	DDSTCLR17	Writing 1 to this bit clears DDED17 in PH_BUSDDERSTR.
16	DDSTCLR16	Writing 1 to this bit clears DDED16 in PH_BUSDDERSTR.
15 to 10	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
9	DDSTCLR09	Writing 1 to this bit clears DDED09 in PH_BUSDDERSTR.
8	DDSTCLR08	Writing 1 to this bit clears DDED08 in PH_BUSDDERSTR.
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DDSTCLR01	Writing 1 to this bit clears DDED01 in PH_BUSDDERSTR.
0	DDSTCLR00	Writing 1 to this bit clears DDED00 in PH_BUSDDERSTR.

38.3.11.54 PH_BUSASERSTR — Address 1-bit Error Status Register

This register indicates that an address ECC 1-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PH_BUSASSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ASEDF 25	—	—	—	—	ASEDF 20	ASEDF 19	ASEDF 18	ASEDF 17	ASEDF 16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.254 PH_BUSASERSTR Register Contents

Bit Position	Bit Name	Function
31 to 26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25	ASEDF25	Indicates that an address ECC 1-bit error was detected in request address to VC12APB bridge for peripheral group 0.
24 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	ASEDF20	Indicates that an address ECC 1-bit error was detected in request address to AXI2PVC1 bridge.
19	ASEDF19	Indicates that an address ECC 1-bit error was detected in request address to MEV.
18	ASEDF18	Indicates that an address ECC 1-bit error was detected in request address to IPIR.
17	ASEDF17	Indicates that an address ECC 1-bit error was detected in request address to BarrierSync.
16	ASEDF16	Indicates that an address ECC 1-bit error was detected in request address to TPTM.
15 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.11.55 PH_BUSADERSTR — Address 2-bit Error Status Register

This register indicates that an address ECC 2-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PH_BUSADSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ADEDF 25	—	—	—	—	ADEDF 20	ADEDF 19	ADEDF 18	ADEDF 17	ADEDF 16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.255 PH_BUSADERSTR Register Contents

Bit Position	Bit Name	Function
31 to 26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25	ADEDF25	Indicates that an address ECC 2-bit error was detected in request address to VC12APB bridge for peripheral group 0.
24 to 21	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
20	ADEDF20	Indicates that an address ECC 2-bit error was detected in request address to AXI2PVC1 bridge.
19	ADEDF19	Indicates that an address ECC 2-bit error was detected in request address to MEV.
18	ADEDF18	Indicates that an address ECC 2-bit error was detected in request address to IPIR.
17	ADEDF17	Indicates that an address ECC 2-bit error was detected in request address to BarrierSync.
16	ADEDF16	Indicates that an address ECC 2-bit error was detected in request address to TPTM.
15 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.11.56 PH_BUSDSESTR — Data 1-bit Error Status Register

This register indicates that a data ECC 1-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PH_BUSDSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	DSEDF 26	DSEDF 25	—	—	—	—	—	DSEDF 19	DSEDF 18	DSEDF 17	DSEDF 16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DSEDF 09	DSEDF 08	—	—	—	—	—	—	DSEDF 01	DSEDF 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.256 PH_BUSDSESTR Register Contents

Bit Position	Bit Name	Function
31 to 27	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
26	DSEDF26	Indicates that a data ECC 1-bit error was detected in read data to VCI2APB bridge for peripheral group 0.
25	DSEDF25	Indicates that a data ECC 1-bit error was detected in write data to VCI2APB bridge for peripheral group 0.
24 to 20	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
19	DSEDF19	Indicates that a data ECC 1-bit error was detected in write data to MEV.
18	DSEDF18	Indicates that a data ECC 1-bit error was detected in write data to IPIR.
17	DSEDF17	Indicates that a data ECC 1-bit error was detected in write data to BarrierSync.
16	DSEDF16	Indicates that a data ECC 1-bit error was detected in write data to TPTM.
15 to 10	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
9	DSEDF09	Indicates that a data ECC 1-bit error was detected in read data to VCI2AXI bridge in cluster 0.
8	DSEDF08	Indicates that a data ECC 1-bit error was detected in write data to VCI2AXI bridge in cluster 0.
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DSEDF01	Indicates that a data ECC 1-bit error was detected in read data to PE1 from peripherals.
0	DSEDF00	Indicates that a data ECC 1-bit error was detected in read data to PE0 from peripherals.

38.3.11.57 PH_BUSDDERSTR — Data 2-bit Error Status Register

This register indicates that a data ECC 2-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PH_BUSDDSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	DDED 26	DDED 25	—	—	—	—	—	DDED 19	DDED 18	DDED 17	DDED 16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DDED 09	DDED 08	—	—	—	—	—	—	DDED 01	DDED 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.257 PH_BUSDDERSTR Register Contents

Bit Position	Bit Name	Function
31 to 27	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
26	DDED26	Indicates that a data ECC 2-bit error was detected in read data to VCI2APB bridge for peripheral group 0.
25	DDED25	Indicates that a data ECC 2-bit error was detected in write data to VCI2APB bridge for peripheral group 0.
24 to 20	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
19	DDED19	Indicates that a data ECC 2-bit error was detected in write data to MEV.
18	DDED18	Indicates that a data ECC 2-bit error was detected in write data to IPIR.
17	DDED17	Indicates that a data ECC 2-bit error was detected in write data to BarrierSync.
16	DDED16	Indicates that a data ECC 2-bit error was detected in write data to TPTM.
15 to 10	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
9	DDED09	Indicates that a data ECC 2-bit error was detected in read data to VCI2AXI bridge in cluster 0.
8	DDED08	Indicates that a data ECC 2-bit error was detected in write data to VCI2AXI bridge in cluster 0.
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DDED01	Indicates that a data ECC 2-bit error was detected in read data to PE1 from peripherals.
0	DDED00	Indicates that a data ECC 2-bit error was detected in read data to PE0 from peripherals.

38.3.11.58 DM_BUSERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when data ECC 2-bit error and data ECC 1-bit error are detected.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDED IE	DSED IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.258 DM_BUSERRINT Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DDEDIE	Controls error reports when data ECC 2-bit error was detected. 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled
0	DSEDIE	Controls error reports when data ECC 1-bit error was detected. 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled

38.3.11.59 DM_BUSDSSTCLR — Data 1-bit Error Status Clear Register

This register is used to clear error flags in DM_BUSDSERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DSSTC LR08	—	—	—	—	DSSTC LR03	DSSTC LR02	DSSTC LR01	DSSTC LR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	R	W	W	W	W

Table 38.259 DM_BUSDSSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	DSSTCLR08	Writing 1 to this bit clears DSEDF08 in DM_BUSDSERSTR.
7 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	DSSTCLR03	Writing 1 to this bit clears DSEDF03 in DM_BUSDSERSTR.
2	DSSTCLR02	Writing 1 to this bit clears DSEDF02 in DM_BUSDSERSTR.
1	DSSTCLR01	Writing 1 to this bit clears DSEDF01 in DM_BUSDSERSTR.
0	DSSTCLR00	Writing 1 to this bit clears DSEDF00 in DM_BUSDSERSTR.

38.3.11.60 DM_BUSDDSTCLR — Data 2-bit Error Status Clear Register

This register is used to clear error flags in DM_BUSDDERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DDSTC LR08	—	—	—	—	DDSTC LR03	DDSTC LR02	DDSTC LR01	DDSTC LR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	R	W	W	W	W

Table 38.260 DM_BUSDDSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	DDSTCLR08	Writing 1 to this bit clears DDED08 in DM_BUSDDERSTR.
7 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	DDSTCLR03	Writing 1 to this bit clears DDED03 in DM_BUSDDERSTR.
2	DDSTCLR02	Writing 1 to this bit clears DDED02 in DM_BUSDDERSTR.
1	DDSTCLR01	Writing 1 to this bit clears DDED01 in DM_BUSDDERSTR.
0	DDSTCLR00	Writing 1 to this bit clears DDED00 in DM_BUSDDERSTR.

38.3.11.61 DM_BUSDSERSTR — Data 1-bit Error Status Register

This register indicates that a data ECC 1-bit error has occurred in a path related to DMA/DTS. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by DM_BUSDSSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DSEDF08	—	—	—	—	DSEDF03	DSEDF02	DSEDF01	DSEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.261 DM_BUSDSERSTR Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	DSEDF08	Indicates that a data ECC 1-bit error was detected in read data to DTS.
7 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	DSEDF03	Indicates that a data ECC 1-bit error was detected in internal buffer of sDMAC 1.
2	DSEDF02	Indicates that a data ECC 1-bit error was detected in read data to sDMAC 1.
1	DSEDF01	Indicates that a data ECC 1-bit error was detected in internal buffer of sDMAC 0.
0	DSEDF00	Indicates that a data ECC 1-bit error was detected in read data to sDMAC 0.

38.3.11.62 DM_BUSDDERSTR — Data 2-bit Error Status Register

This register indicates that a data ECC 2-bit error has occurred in a path related to DMA/DTS. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by DM_BUSDDSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DDED F08	—	—	—	—	DDED F03	DDED F02	DDED F01	DDED F00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.262 DM_BUSDDERSTR Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	DDED F08	Indicates that a data ECC 2-bit error was detected in read data to DTS.
7 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	DDED F03	Indicates that a data ECC 2-bit error was detected in internal buffer of sDMAC 1.
2	DDED F02	Indicates that a data ECC 2-bit error was detected in read data to sDMAC 1.
1	DDED F01	Indicates that a data ECC 2-bit error was detected in internal buffer of sDMAC 0.
0	DDED F00	Indicates that a data ECC 2-bit error was detected in read data to sDMAC 0.

38.3.11.63 EM_BUSERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address/data ECC 2-bit error and address/data ECC 1-bit error are detected.

This register is protected by the KCPROT register. When KCPROT.KCE = 0, it cannot be written. It can be written only when KCPROT.KCE = 1.

Value after reset: 0000 000F_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADED IE	ASED IE	DDED IE	DSED IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.263 EM_BUSERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	ADEDIE	Controls error reports when address ECC 2-bit error was detected. 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled
2	ASEDIE	Controls error reports when address ECC 1-bit error was detected. 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled
1	DDEDIE	Controls error reports when data ECC 2-bit error was detected. 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled
0	DSEDIE	Controls error reports when data ECC 1-bit error was detected. 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled

38.3.11.64 EM_BUSASSTCLR — Address 1-bit Error Status Clear Register

This register is used to clear error flags in EM_BUSASERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASSTCLR17	ASSTCLR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ASSTCLR03	ASSTCLR02	ASSTCLR01	ASSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 38.264 EM_BUSASSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 18	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
17	ASSTCLR17	Writing 1 to this bit clears ASEDf17 in EM_BUSASERSTR.
16	ASSTCLR16	Writing 1 to this bit clears ASEDf16 in EM_BUSASERSTR.
15 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	ASSTCLR03	Writing 1 to this bit clears ASEDf03 in EM_BUSASERSTR.
2	ASSTCLR02	Writing 1 to this bit clears ASEDf02 in EM_BUSASERSTR.
1	ASSTCLR01	Writing 1 to this bit clears ASEDf01 in EM_BUSASERSTR.
0	ASSTCLR00	Writing 1 to this bit clears ASEDf00 in EM_BUSASERSTR.

38.3.11.65 EM_BUSADSTCLR — Address 2-bit Error Status Clear Register

This register is used to clear error flags in EM_BUSADERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADSTC LR17	ADSTC LR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADSTC LR03	ADSTC LR02	ADSTC LR01	ADSTC LR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 38.265 EM_BUSADSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 18	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
17	ADSTCLR17	Writing 1 to this bit clears ADEDF17 in EM_BUSADERSTR.
16	ADSTCLR16	Writing 1 to this bit clears ADEDF16 in EM_BUSADERSTR.
15 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	ADSTCLR03	Writing 1 to this bit clears ADEDF03 in EM_BUSADERSTR.
2	ADSTCLR02	Writing 1 to this bit clears ADEDF02 in EM_BUSADERSTR.
1	ADSTCLR01	Writing 1 to this bit clears ADEDF01 in EM_BUSADERSTR.
0	ADSTCLR00	Writing 1 to this bit clears ADEDF00 in EM_BUSADERSTR.

38.3.11.66 EM_BUSDSSTCLR — Data 1-bit Error Status Clear Register

This register is used to clear error flags in EM_BUSDSERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSSTCLR17	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.266 EM_BUSDSSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 18	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
17	DSSTCLR17	Writing 1 to this bit clears DSEDF17 in EM_BUSDSERSTR.
16 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.11.67 EM_BUSDDSTCLR — Data 2-bit Error Status Clear Register

This register is used to clear error flags in EM_BUSDDERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDSTC LR17	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.267 EM_BUSDDSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 18	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
17	DDSTCLR17	Writing 1 to this bit clears DDED17 in EM_BUSDDERSTR.
16 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.11.68 EM_BUSASERSTR — Address 1-bit Error Status Register

This register indicates that an address ECC 1-bit error has occurred in a path related to emulation. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by EM_BUSASSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASEDF 17	ASEDF 16	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	—	—	—	—	—	—	—	—	—	—	—	ASEDF 03	ASEDF 02	ASEDF 01	ASEDF 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Table 38.268 EM_BUSASERSTR Register Contents

Bit Position	Bit Name	Function
31 to 18	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
17	ASEDF17	Indicates that an address ECC 1-bit error was detected in request address to FAXI2SAXI bridge.
16	ASEDF16	Indicates that an address ECC 1-bit error was detected in request address to the remap module from a SAXI master.
15 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	ASEDF03	Indicates that an address ECC 1-bit error was detected in request address to the remap module for data read from PE1.
2	ASEDF02	Indicates that an address ECC 1-bit error was detected in request address to the remap module for instruction fetch from PE1.
1	ASEDF01	Indicates that an address ECC 1-bit error was detected in request address to the remap module for data read from PE0.
0	ASEDF00	Indicates that an address ECC 1-bit error was detected in request address to the remap module for instruction fetch from PE0.

38.3.11.69 EM_BUSADERSTR — Address 2-bit Error Status Register

This register indicates that an address ECC 2-bit error has occurred in a path related to emulation. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by EM_BUSADSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADEDF17	ADEDF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADEDF03	ADEDF02	ADEDF01	ADEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.269 EM_BUSADERSTR Register Contents

Bit Position	Bit Name	Function
31 to 18	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
17	ADEDF17	Indicates that an address ECC 2-bit error was detected in request address to FAXI2SAXI bridge.
16	ADEDF16	Indicates that an address ECC 2-bit error was detected in request address to the remap module from a SAXI master.
15 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	ADEDF03	Indicates that an address ECC 2-bit error was detected in request address to the remap module for data read from PE1.
2	ADEDF02	Indicates that an address ECC 2-bit error was detected in request address to remap module for instruction fetch from PE1.
1	ADEDF01	Indicates that an address ECC 2-bit error was detected in request address to remap module for data read from PE0.
0	ADEDF00	Indicates that an address ECC 2-bit error was detected in request address to remap module for instruction fetch from PE0.

38.3.11.70 EM_BUSDSERSTR — Data 1-bit Error Status Register

This register indicates that a data ECC 1-bit error has occurred in a path related to emulation. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by EM_BUSDSSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSEDF 17	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.270 EM_BUSDSERSTR Register Contents

Bit Position	Bit Name	Function
31 to 18	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
17	DSEDF17	Indicates that a data ECC 1-bit error was detected in read data to FAXI2SAXI bridge.
16 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.11.71 EM_BUSDDERSTR — Data 2-bit Error Status Register

This register indicates that a data ECC 2-bit error has occurred in a path related to emulation. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by EM_BUSDDSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDED17	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.271 EM_BUSDDERSTR Register Contents

Bit Position	Bit Name	Function
31 to 18	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
17	DDED17	Indicates that a data ECC 2-bit error was detected in read data to FAXI2SAXI bridge.
16 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.11.72 APECECCCTL — ECC Control Register

This register controls the ECC error detection/correction and 1-bit error correction.

This register is protected by the key-code protection register. When KCE = 0, it cannot be written. It can be written only when KCE = 1.

Value after reset: 0000 0008_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	AECC DIS	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 38.272 APECECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	AECCDIS	Address ECC disable bit Sets address ECC error detection to enable/disable. 0: Address ECC error detection is enabled 1: Address ECC error detection is disabled
1	SECDIS	ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

38.3.11.73 APECERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address/data ECC 2-bit error, address/data ECC 1-bit error are detected.

This register is protected by the key-code protection register. When KCE = 0, it cannot be written. It can be written only when KCE = 1.

Value after reset: 0000 0033_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	ADED IE	ASED IE	—	—	DED IE	SED IE
Value after reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Table 38.273 APECERRINT Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	ADEDIE	Controls error reports when address ECC 2-bit error is detected. 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled
4	ASEDIE	Controls error reports when address ECC 1-bit error is detected. 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when data ECC 2-bit error is detected. 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled
0	SEDIE	Controls error reports when data ECC 1-bit error is detected. 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled

38.3.11.74 APECSTCLR — Error Status Clear Register

This register is used to clear error flag in APEC1STERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ST CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.274 APECSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	STCLR0	Writing 1 to this bit clears AECDEDF0/AECSEDF0/DEDF0/SEDF0 in APEC1STERSTR.

38.3.11.75 APEC1STERSTR — Error Status Register

This register indicates whether an error has occurred. When all error flags are “0” and a new error occurs, error status flag is set.

If address ECC 1-bit error or data ECC 1-bit error is set and the new error is address ECC 2-bit error or data ECC 2-bit error, the new error is set (does not clear the previous error flag).

If multiple causes of errors are detected, the flags of the detected errors are all set (e.g. if data ECC 2-bit error and address ECC 2-bit error are detected at the same access, DEDF0 and AECDEDF0 are both set.).

This register can be cleared by STCLR0 in APECSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	AEC DEDFO	AEC SEDF0	—	—	—	—	DEDFO	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.275 APEC1STERSTR Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	AECDEDF0	Indicates that an address ECC 2-bit error was detected. This is set when address ECC 2-bit error is detected when AECDEDF0 and DEDFO are both “0”.
6	AECSEDF0	Indicates that an address ECC 1-bit error was detected. This is set when address ECC 1-bit error is detected when all error flags are “0”.
5 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDFO	Indicates that a data ECC 2-bit error was detected. This is set when data ECC 2-bit error is detected when AECDEDF0 and DEDFO are both “0”.
0	SEDF0	Indicates that a data ECC 1-bit error was detected. This is set when data ECC 1-bit error is detected when all error flags are “0”.

38.3.11.76 APECTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After ECC test mode (ECCTST = 1) setting, arbitrary data value can be input to ECC decoder.

This register is protected by the key-code protection register. When KCE = 0, it cannot be written. It can be written only when KCE = 1.

When accessing any registers in the same peripheral group, ECCTST is cleared to 0.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R

Table 38.276 APECTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for the address ECC decoder. 0:Normal mode 1:Test mode The values of APECTSTDIN0 and APECTSTEIN registers are provided to data and ECC field of the ECC decoder respectively when accessing any registers in the same peripheral group in test mode. This bit is always read as 0.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.11.77 APECTSTDIN0 — Test Data Input Register

This register is used to inject errors into data field of ECC decoder for self-diagnosis. The values of APECTSTDIN0 and APECTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the key-code protection register. When KCE = 0, it cannot be written. It can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.277 APECTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for the address ECC decoder. The data is provided to data field of the ECC decoder when accessing any registers in the same peripheral group when ECCTST = 1.

38.3.11.78 APECTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of ECC decoder for self-diagnosis. The values of APECTSTDIN0 and APECTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the key-code protection register. When KCE = 0, it cannot be written. It can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.278 APECTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECC[6:0]	Specify the test data for the address ECC decoder. The data is provided to ECC field of the ECC decoder when accessing any registers in the same peripheral group when ECCTST = 1.

38.3.11.79 APECKCPROT — Key Code Protection Register

This register is used for protection against writing to ECC control registers and Error notification control registers due to program malfunction and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.279 APECKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit*2 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

Note 2. Unlocking this protection is one of necessary conditions to enable writing the protected registers. It is also necessary to unlock security protection driven by the ICUM when ICUM function is active. For details, see *the RH850/E2x-FCC1 ICUMD User's Manual: Hardware, Section 9*.

The security protection covers the following modules:

ECCCNT_A_V2An (n = 1, 6, 9)
ECCCNT_D_V2AnW (n = 1, 6, 9)
ECCCNT_D_V2AnR (n = 1, 6, 9)
PBnECC (n = 0, 1, 6, 9)
BECCCAP_V2An (n = 1, 6, 9)

38.3.11.80 PHC_BUSERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address/data ECC 2-bit error and address/data ECC 1-bit error are detected.

This register is protected by the PBnECC.APECKCPROT register. When PBnECC.APECKCPROT.KCE = 0, it cannot be written. It can be written only when PBnECC.APECKCPROT.KCE = 1.

Note: For BECCCAP_V2An of peripheral group2, PB2HECC.APECKCPROT controls access protection.

Value after reset: 0000 000F_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADED IE	ASED IE	DDED IE	DSED IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.280 PHC_BUSERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	ADEDIE	Controls error reports when address ECC 2-bit error is detected. 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled
2	ASEDIE	Controls error reports when address ECC 1-bit error is detected. 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled
1	DDEDIE	Controls error reports when data ECC 2-bit error is detected. 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled
0	DSEDIE	Controls error reports when data ECC 1-bit error is detected. 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled

38.3.11.81 PHC_BUSASSTCLR — Address 1-bit Error Status Clear Register

This register is used to clear error flags in PHC_BUSASERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.281 PHC_BUSASSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ASSTCLR00	Writing 1 to this bit clears ASSEDF00 in PHC_BUSASERSTR.

38.3.11.82 PHC_BUSADSTCLR — Address 2-bit Error Status Clear Register

This register is used to clear error flags in PHC_BUSADERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.282 PHC_BUSADSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ADSTCLR00	Writing 1 to this bit clears ADEDF00 in PHC_BUSADERSTR.

38.3.11.83 PHC_BUSDSSTCLR — Data 1-bit Error Status Clear Register

This register is used to clear error flags in PHC_BUSDSERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSST CLR01	DSST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.283 PHC_BUSDSSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DSSTCLR01	Writing 1 to this bit clears DSEDF01 in PHC_BUSDSERSTR.
0	DSSTCLR00	Writing 1 to this bit clears DSEDF00 in PHC_BUSDSERSTR.

38.3.11.84 PHC_BUSDDSTCLR — Data 2-bit Error Status Clear Register

This register is used to clear error flags in PHC_BUSDDERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDST CLR01	DDST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.284 PHC_BUSDDSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DDSTCLR01	Writing 1 to this bit clears DDEDF01 in PHC_BUSDDERSTR.
0	DDSTCLR00	Writing 1 to this bit clears DDEDF00 in PHC_BUSDDERSTR.

38.3.11.85 PHC_BUSASERSTR — Address 1-bit Error Status Register

This register indicates that an address ECC 1-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PHC_BUSASSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.285 PHC_BUSASERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ASEDF00	Indicates that an address ECC 1-bit error was detected in request address to VC12APB bridge for peripheral group n.

38.3.11.86 PHC_BUSADERSTR — Address 2-bit Error Status Register

This register indicates that an address ECC 2-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PHC_BUSADSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADEF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.286 PHC_BUSADERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ADEF00	Indicates that an address ECC 2-bit error was detected in request address to VC12APB bridge for peripheral group n.

38.3.11.87 PHC_BUSDSESTR — Data 1-bit Error Status Register

This register indicates that a data ECC 1-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PHC_BUSDSSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSEDF01	DSEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.287 PHC_BUSDSESTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DSEDF01	Indicates that a data ECC 1-bit error was detected in read data to VCI2APB bridge for peripheral group n.
0	DSEDF00	Indicates that a data ECC 1-bit error was detected in write data to VCI2APB bridge for peripheral group n.

38.3.11.88 PHC_BUSDDERSTR — Data 2-bit Error Status Register

This register indicates that a data ECC 2-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PHC_BUSDDSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDED01	DDED00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.288 PHC_BUSDDERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DDED01	Indicates that a data ECC 2-bit error was detected in read data to VCI2APB bridge for peripheral group n.
0	DDED00	Indicates that a data ECC 2-bit error was detected in write data to VCI2APB bridge for peripheral group n.

38.3.11.89 HBECECCCTL — ECC Control Register

This register controls the ECC error detection/correction and 1-bit error correction.

This register is protected by the key-code protection register. When KCE = 0, it cannot be written. It can be written only when KCE = 1.

Value after reset: 0000 0008_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	AECC DIS	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 38.289 HBECECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
2	AECCDIS	Address ECC disable bit (for H-Bus slave) Sets address ECC error detection to enable/disable. 0: Address ECC error detection is enabled 1: Address ECC error detection is disabled For H-Bus master, when writing, write the value after reset because this bit is reserved.
1	SECDIS	ECC 1 bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

38.3.11.90 HBECERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address/data ECC 2-bit error, address/data ECC 1-bit error are detected.

This register is protected by the key-code protection register. When KCE = 0, it cannot be written. It can be written only when KCE = 1.

Value after reset: 0000 0033_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	ADED IE	ASED IE	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Table 38.290 HBECERRINT Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5	ADEDIE	Controls error reports when address ECC 2-bit error is detected (for H-Bus slave). 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled For H-Bus master, when writing, write the value after reset because this bit is reserved.
4	ASEDIE	Controls error reports when address ECC 1-bit error is detected (for H-Bus slave). 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled For H-Bus master, when writing, write the value after reset because this bit is reserved.
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when data ECC 2-bit error is detected. 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled
0	SEDIE	Controls error reports when data ECC 1-bit error is detected. 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled

38.3.11.91 HBECSTCLR — Error Status Clear Register

This register is used to clear error flag in HBEC1STERSTR. This is write only register and read value is always “0”.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ST CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.291 HBECSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	STCLR0	Writing 1 to this bit clears AECDEDF0/AECSEDF0/DEDF0/SEDF0 in HBEC1STERSTR.

38.3.11.92 HBEC1STERSTR — Error Status Register

This register indicates whether error has occurred. When all error flags are “0” and a new error occurs, error status flag is set.

If address ECC 1-bit error or data ECC 1-bit error are set and the new error is address ECC 2-bit error or data ECC 2-bit error, the new error is set (does not clear the previous error flag).

If multiple causes of errors are detected, the errors that are detected are all set (e.g. if data ECC 2-bit error and address ECC 2-bit error are detected at the same access, DEDF0 and AECDEDF0 are both set.).

This register can be cleared by STCLR0 in HBECSTCLR register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	AEC DEDFO	AEC SEDF0	—	—	—	—	DEDFO	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.292 HBEC1STERSTR Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7	AECDEDF0	Indicates address ECC 2-bit error was detected (for H-Bus slave). This is set when address ECC 2-bit error is detected when AECDEDF0 and DEDFO are all “0”. For H-Bus master, when read, the value after reset is returned because this bit is reserved.
6	AECSEDF0	Indicates address ECC 1-bit error was detected (for H-Bus slave). This is set when address ECC 1-bit error is detected when all error flags are “0”. For H-Bus master, when read, the value after reset is returned because this bit is reserved.
5 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DEDFO	Indicates data ECC 2-bit error was detected. This is set when data ECC 2-bit error is detected when AECDEDF0 and DEDFO are all “0”.
0	SEDF0	Indicates data ECC 1-bit error was detected. This is set when data ECC 1-bit error is detected when all error flags are “0”.

38.3.11.93 HBECTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After ECC test mode (ECCTST = 1) setting, arbitrary data value can be input to ECC decoder of H-Bus slave. For H-Bus master, keep the value after reset because this register is reserved.

This register is protected by the key-code protection register. When KCE = 0, it cannot be written. It can be written only when KCE = 1.

When accessing any registers in the same peripheral group, ECCTST is cleared to 0.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R

Table 38.293 HBECTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for the address ECC decoder. 0:Normal mode 1:Test mode The values of HBECTSTDIN0 and HBECTSTEIN registers are provided to data and ECC field of the ECC decoder respectively when accessing any registers in the same peripheral group in test mode. This bit is always read as 0.
0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.3.11.94 HBECTSTDIN0 — Test Data Input Register

This register is used to inject errors into data field of ECC decoder for self-diagnosis. The values of HBECTSTDIN0 and HBECTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the key-code protection register. When KCE = 0, it cannot be written. It can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.294 HBECTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for the address ECC decoder. The data is provided to data field of the ECC decoder of H-Bus slave when accessing any registers in the same peripheral group when ECCTST = 1. For H-Bus master, keep the value after reset because this register is reserved.

38.3.11.95 HBECTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of ECC decoder for self-diagnosis. The values of HBECTSTDIN0 and HBECTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the key-code protection register. When KCE = 0, it cannot be written. It can be written only when KCE = 1.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.295 HBEECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECC[6:0]	Specify the test data for the address ECC decoder. The data is provided to ECC field of the ECC decoder of H-Bus slave when accessing any registers in the same peripheral group when ECCTST = 1. For H-Bus master, keep the value after reset because this register is reserved.

38.3.11.96 HBECKCPROT — Key Code Protection Register

This register is used for protection against writing to the HBnECC control registers due to program malfunction and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.296 HBECKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

38.3.11.97 Test Function

ECC on the write data, the read data and the address are checked by the each ECC decoder. Each ECC decoder has a self-diagnosis function that uses ERRGEN module.

For details, refer to **Section 38.3.12.8, Usage of ERRGEN.**

38.3.12 Error Injection for ECC Function

38.3.12.1 Overview

ERRGEN is a utility module for self-diagnosis tests of ECC decoders. ERRGEN can be used to generate a data pairs and ECC including errors in any bit. ERRGEN has the following features.

Table 38.297 Error Injection for ECC Function

Item	Description
ECC calculation	Provides the correct 7-bit and 8-bit ECC value of arbitrary data.
Error injection	<p>Injects various errors into read data include ECC field.</p> <ul style="list-style-type: none"> • Error injection is only into the 32-bit data and 7-bit ECC fields. • No function to inject errors into 64-bit data and 8-bit ECC fields directly but ECC calculation register is available.
Register set	A set of registers in odd and even words respectively used to inject errors to 64-bit bus consisting of 2 sets of 32-bit data and 7-bit ECC are implemented, and also a register for users to calculate 8-bit ECC field.

38.3.12.2 List of Registers

Table 38.298 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
ERRGEN (Peripheral Group0)	Error Data Register	EGDATORG _n (n = 0-1)	FFC5 8400 _H + n × 04 _H	8, 16, 32	
	7-bit ECC Calculation Register	EGECCCAL _n (n = 0-1)	FFC5 8408 _H + n × 04 _H	8, 16, 32	
	Error Injection Register	EGDATINV _n (n = 0-1)	FFC5 8410 _H + n × 04 _H	8, 16, 32	
	Error Injection Register	EGECCINV _n (n = 0-1)	FFC5 8418 _H + n × 04 _H	8, 16, 32	
	8-bit ECC Calculation Register	EGECC8CAL	FFC5 8420 _H	8, 16, 32	

38.3.12.3 EGDATORGn — Error Data Register

This register is used to set original data before the error injection and to obtain the modified data containing injected errors according to the settings of EGDATINVn and EGECCINVn. The read data is the value of this register XOR-ed with EGDATINVn and the read value of ECC is EGECCCALn XOR-ed with EGECCINVn when reading this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATORG[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATORG[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.299 EGDATORGn Register Contents

Bit Position	Bit Name	Function
31 to 0	DATORG[31:0]	Write data before injecting errors and read data after errors are injected

38.3.12.4 EGECCALn — 7-bit ECC Calculation Register

This register is used to obtain a 7-bit ECC value for the arbitrary value written to EGDATORGn. This ECC value is then injected into an ECC decoder on the address channel. ECC value calculated from EGDATORGn can be read when reading this register. The value of this register is updated simultaneously when EGDATORGn is updated.

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECCCAL[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.300 EGECCALn Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECCCAL[6:0]	Read ECC values calculated from EGDATORGn

38.3.12.5 EGDATINVn — Error Injection Register

This register is used to inject errors into the read data signal when reading EGDATORGn. The read data signal is driven by EGDATORGn XOR-ed with this register.

Value after reset: 0000 0000_H

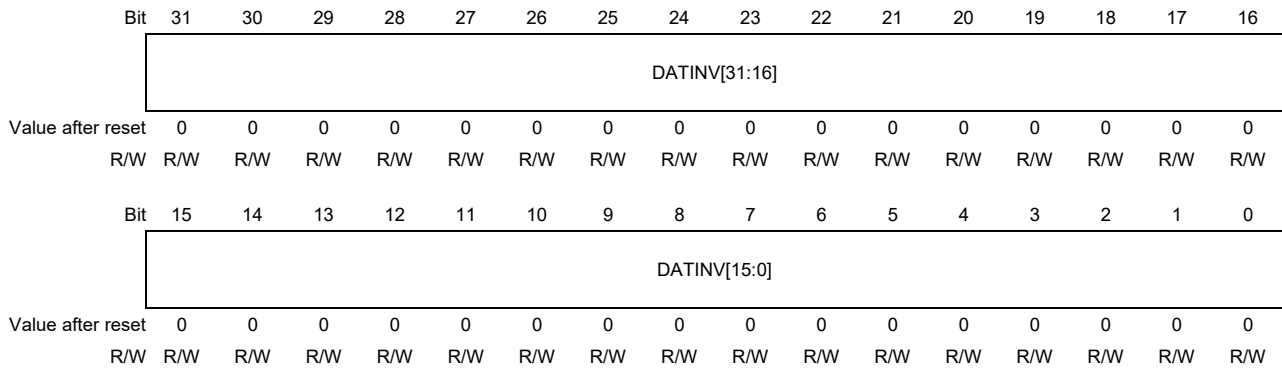


Table 38.301 EGDATINVn Register Contents

Bit Position	Bit Name	Function
31 to 0	DATINV[31:0]	Bit pattern to invert EGDATORGn value when reading EGDATORGn

38.3.12.6 EGECCINVn — Error Injection Register

This register is used to inject errors into the ECC signal when reading EGDATORGn. The ECC signal is driven by EGECCCALn XOR-ed with this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECCINV[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.302 EGECCINVn Register Contents

Bit Position	Bit Name	Function
31 to 7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECCINV[6:0]	Bit pattern to invert ECC value of EGDATORGn when reading EGDATORGn

38.3.12.7 EGEC8CAL — 8-bit ECC Calculation Register

This register is used to obtain an ECC values for the arbitrary values written to EGDATORG0 and EGDATORG1. These ECC values are then injected into ECC decoders on Data RAM of sDMAC. ECC values calculated from {EGDATORG1, EGDATORG0} can be read when reading this register. The value of this register is updated simultaneously when either EGDATORG0 or EGDATORG1 is updated.

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ECC8CAL[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.303 EGEC8CAL Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	ECC8CAL[7:0]	Read ECC values calculated from {EGDATORG1,EGDATORG0}

38.3.12.8 Usage of ERRGEN

(1) Error Injection into Read Data Channel

ERRGEN can be used to run self-diagnosis tests for ECC decoders of some bus masters on read data channel. Self-diagnosis tests for an ECC decoder on read data channel are made in the following way.

1. Write arbitrary data to EGDATORG0 and EGDATORG1.
2. Set any 1 or 2 bits among EGDATINV0, EGDATINV1, EGECCINV0 and EGECCINV1 to 1.
3. Read EGDATORG0 and EGDATORG1 at the same time in double word access from a target bus master.
4. Check ECC error notification asserted from the bus master.

(2) Error Injection into Write Data Channel

ERRGEN can be used to run self-diagnosis tests for ECC decoders of some bus slaves on write data channel. Self-diagnosis tests for an ECC decoder on write data channel are made in the following way.

1. Write arbitrary data to EGDATORG0 and EGDATORG1.
2. Set any 1 or 2 bits among EGDATINV0, EGDATINV1, EGECCINV0 and EGECCINV1 to 1.
3. Make DTS transfer data from EGDATORG0 and EGDATORG1 to a target bus slave in 64-bit mode.
4. Check ECC error notification asserted from the bus slave.

(3) Error Injection into Address Channel

This case uses a different feature of ERRGEN, which can calculate ECC value of data. It is necessary for users to know correct ECC value for any data to inject various errors into ECC decoders of bus slaves on address channel. Self-diagnosis tests for an ECC decoder on address channel are made in the following way.

1. Write arbitrary data to EGDATORGn.
2. Read EGECCCALn to obtain correct ECC value of the data.
3. Invert any 1 or 2 bits among the data and the ECC value.
4. Set the modified data and ECC value to related registers*¹ of the Address EDC decoder and enable its test function.
5. Read data from a target bus slave including the ECC decoder.
6. Check ECC error notification asserted from the ECC decoder.

(4) Error Injection into Data RAM of sDMAC

ERRGEN can also calculate ECC value of 64-bit data. It is necessary for users to learn correct ECC value for any data to inject various errors into ECC decoders on Data RAM of sDMAC. Self-diagnosis tests for an ECC decoder on Data RAM of sDMAC are made in the following way.

1. Write arbitrary data to EGDATORG0 and EGDATORG1.
2. Read EGECC8CAL to obtain correct ECC value of the data.
3. Invert any 1 or 2 bits among the data and the ECC value.
4. Set the modified data and ECC value to related registers*² of the sDMAC Data RAM ECC module and enable its test function.
5. Make sDMAC including the ECC decoder transfer data from anywhere to anywhere.

6. Check ECC error notification asserted from the ECC decoder.

Note 1: For details, refer to **Section 38.3.11, Safety Mechanism on Data Transfer Path.**

Note 2: For details, refer to **Section 38.3.8, sDMAC/DTS RAM ECC and Address Feedback.**

38.4 Hardware Redundancy

38.4.1 Overview

This product implements the CPU0/1 and DTS with the lockstep function to quickly detect failures without software interaction. The CPU0/1 and DTS execute the function using two different cores; that is, a master core and checker core, and constantly compare the execution results of the two cores. When the results do not match, an error is reported to the ECM. The comparison targets of the CPU's bus outputs are own Local RAM, Cluster RAM, Inter-processor element Bus, Local FLASH Bus, and the tag RAM and data RAM of Instruction cache.

The lockstep function of the CPU0/1 and DTS features of failure insertion, through which errors can be intentionally triggered and thus self-diagnosis of the lockstep operation is possible. For details on CPU0/1's failure insertion, refer to **Section 3, CPU System**.

Following table shows redundant structure implemented in this product.

Table 38.304 Redundant Blocks

Block Name	Features	Comparator Error Injection
CPU0, CPU1 (G4MHPE)	Dual lock step with 2-cycle delay and inverted input for comparator	Yes
DTS	Dual lock step with 2-cycle delay and inverted input for comparator	Yes

38.4.2 List of Registers

Table 38.305 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
DTS_COMPC (Peripheral Group0)	Error Injection Control Register	DTS_COMP_CNTRL	FFC5 8200 _H	32	Self

38.4.3 DTS_COMP_CNTRL — Error Injection Control Register

This register can control the output signals on the checker side of the DTS. A DTS comparison error can be generated by setting this register.

Writing to this register should be executed with $PROT[31:30] = 01_B$.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PROT[31:30]		DTSCMPERR[29:16]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTSCMPERR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.306 DTS_COMP_CNTRL Register Contents

Bit Position	Bit Name	Function
31 to 30	PROT[31:30]	Protection Bit 01: Writing is enabled Other: Writing is disabled These bits are always read as 0.
29 to 0	DTSCMPERR[29:0]	A DMA comparison error can be generated by writing 111111_11111011_11111111_11111111 to DTSCMPERR together with the PROT bit. Clear all these bits to 0 if there is no need to generate this error.

38.4.4 Usage Note

Reading a register with a value that is undefined after a reset without initializing the register may lead to a CPU comparison error. Accordingly, such registers must be initialized with the desired settings. Even if the branch instruction and the subsequent instruction is issued in parallel, the CPU comparison error might be occurred by undefined register after the reset. It should be applied as specified below until the register which refer by subsequent instruction is initialized in case of branching in the preceding instruction.

- Insert the SYNCI instruction or the RIE instruction following the branch instruction. (It has to be added by assembler language. When C language is used, it could be optimized.)
- Applicable branch instructions: Bcond except BR, JARL, JMP

38.5 Memory Protection

38.5.1 Overview

This product incorporates the memory protection/guard functions to prevent erroneous access to data in the memory and control registers of the peripheral modules. **Figure 38.1** shows the overall memory protection architecture. Each programmable core (bus master) has a Memory Protection Unit (MPU) that defines the software access protection. In addition, each resource (bus slave) has a guard that controls the access by any bus master, including ones that do not have an MPU such as the DMA.

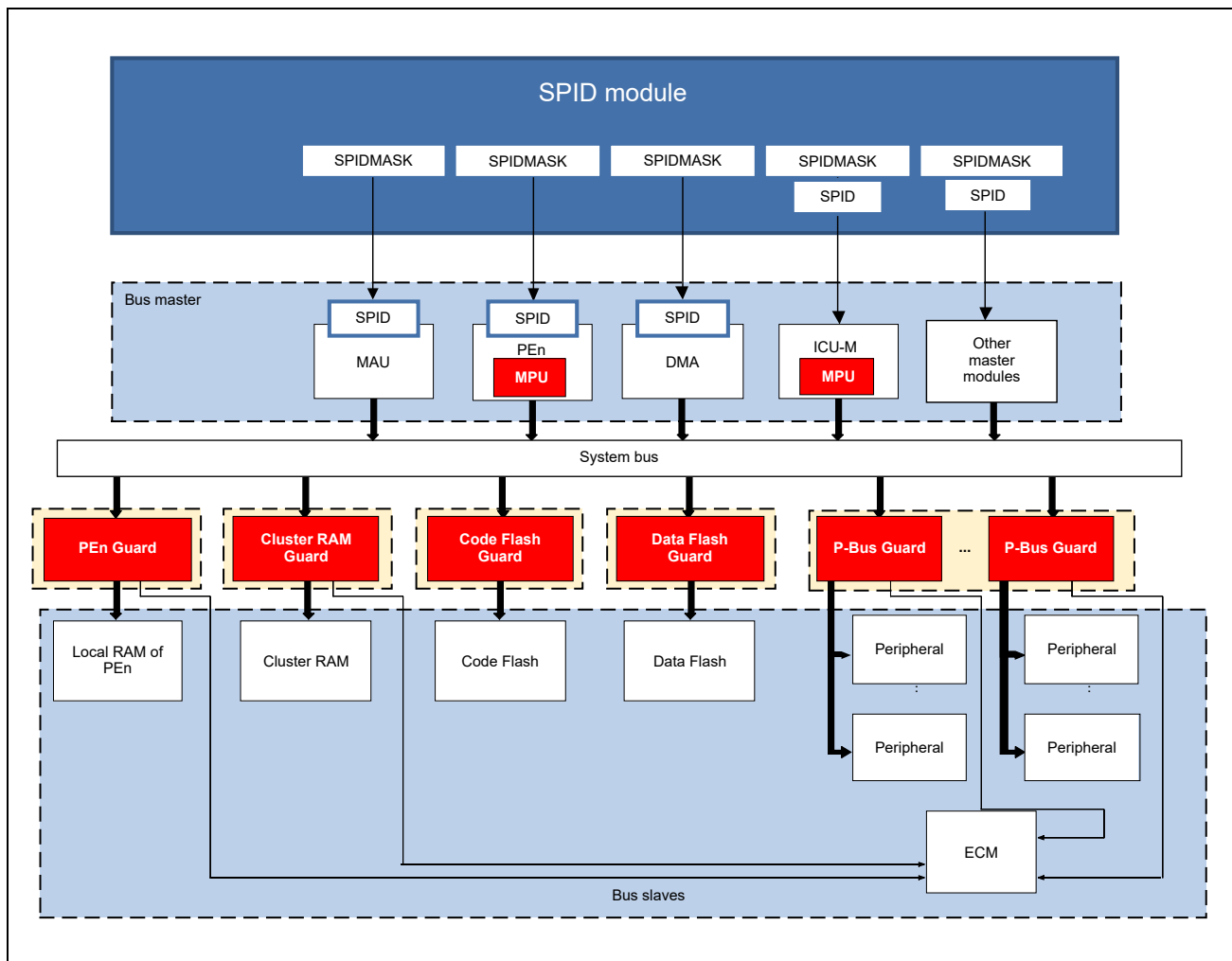


Figure 38.1 Memory Protection Architecture

Note: For details of Code Flash Guard and Data Flash Guard, See *RH850/E2x-FCC1 ICUMD User's Manual: Hardware*.

• MPU

All CPU cores and ICU-M protect the memory against unintended access from the CPU cores and ICU-M themselves. Requests to access addresses that are prohibited by the MPU are never issued by the CPU cores or ICU-M. For details, refer to **Section 3, CPU System** and *the RH850ICUMD User's Manual: Hardware*.

• Slave Guard

A specific memory is protected against unintended access from any bus master. Slave guard includes the following guard types. The details of each type are given in the following sections.

– PEG

Local RAM and INTC1 are protected against unintended access except from the CPU incorporating the Local RAM and INTC1 themselves. For example, access from the CPU0 to Local RAM in the CPU0 is not rejected by the PEG.

For details, refer to **Section 38.5.3, PEG**.

– CRG

Cluster RAM is protected against unintended access.

– Peripheral Guard

The control registers in the peripheral circuits and memory are protected against unintended access.

Peripheral Guard includes INTC2 Guard, DTS Guard, sDMAC Guard, IBG, HBG and PBG.

For details, refer to **Section 38.5.5, INTC2 Guard** to **Section 38.5.10, HBG**.

For this purpose, each bus access contains the following context information that identifies the bus master in addition to access address.

Table 38.307 Identifiers for Memory Protection

Identifier	Function
UM	Indicates the operating mode of the CPU or ICU-M when the CPU core accesses an address. 0: Supervisor mode 1: User mode When DTS or sDMAC or MAU ¹ accesses an address, this identifier indicates the value set in each dedicated register. When the other masters accesses an address, the value of this identifier is always 0.
SEC	Indicates the access mode of the ICU-M when ICU-M accesses an address. 0: Non-secure 1: Secure When the other master accesses an address, the value of this identifier is always 0.
DBG	Indicates the access mode of the debug modules when the debug module accesses an address. 0: Normal mode 1: Debug mode When the other master accesses an address, the value of this identifier is always 0.
SPID	Indicates the system protection identifier SPID that is assigned by the system register when the CPU core or ICU-M or DTS/sDMAC accesses an address. When DTS or sDMAC or MAU ¹ accesses an address, this identifier indicates the value set in each dedicated register. When the other masters accesses an address, this identifier indicates the value set in SPID module. See Section 38.5.2, SPID for details.
WRITE	Indicates the access type of bus master. 0: Read access 1: Write access

Note 1. MAU is one of the debug functions to read and write the memory and registers.

38.5.2 SPID

38.5.2.1 Overview

SPID (System Protection identifier) indicates an identifier of the software task. The value of SPID for CPU cores, DTS/sDMAC and MAU*² can be set individually in each configuration register. For the other masters, it can be set in SPID module.

SPID module supervises all SPIDs except for some bus masters which have own configuration register. These SPID registers present in SPID module are configurable by software. In addition, the SPID module also supervises SPIDs which all bus masters use through SPID mask registers.

SPID mask registers can be locked by SPID mask lock registers for security purposes. Once SPID mask lock register locks a SPID mask register, it can be cleared by Power Up Reset, System Reset1, System Reset2, and Application Reset.

Table 38.308 shows the initial values of SPID for each bus master.

Table 38.308 Initial Value and Register Location of SPID

Initial value of SPID	Bus master	Register location
0	CPU0	CPU0
1	CPU1	CPU1
2 to 18	Reserved	—
19	RHSIF0	SPID module
20	Reserved	—
21	HSSPI0	SPID module
22	Reserved	—
23	FlexRay0	SPID module
24	Ethernet	SPID module
25	Reserved	—
26	ICU-M	SPID module
27	sDMAC1	sDMAC1
28	sDMAC0	sDMAC0
29	DTS* ¹	DTS
30	Reserved	—
31	MAU	MAU

Note 1. Initial value of SPID for DTS is stored in DTS RAM. Therefore, it is undefined after reset and it becomes 0 after RAM initialization. Please refer to **Section 10.3.6**, about RAM initialization in detail.

Note 2. MAU is one of the debug functions to read and write the memory and registers.

Features of SPID:

- SPID of each bus master is configurable by software with high flexibility according to user's system.
- It is possible to assign one SPID to one bus master or assign one SPID to several bus masters.
- It is possible to limit SPID that each bus master uses by SPIDMASK register. Moreover this configuration can be locked.
- The lock function above prevents unauthorized use of SPID for security purpose.

Table 38.309 shows a case which eight SPIDs are allocated only to CPU0 and other SPIDs are allocated to each remained bus master. **Table 38.310** shows a case which some SPIDs are allocated to each CPU and some peripheral bus masters use one same SPID.

The system integrator should define the SPID assignment according to the system.

Table 38.309 Example of SPID setting (Several SPIDs are allocated only to CPU0)

SPID value	Bus master	Lock/Unlock
0	CPU0	Unlocked
1	CPU1	Locked
2 to 5	Reserved	—
6 to 12	CPU0	Unlocked
13 to 18	Reserved	—
19	RHSIF0	Locked
20	Reserved	—
21	HSSPI0	Locked
22	Reserved	—
23	FlexRay0	Locked
24	Ethernet	Locked
25	Reserved	—
26	ICU-M	Locked
27	sDMAC1	Locked
28	sDMAC0	Locked
29	DTS	Locked
30	Reserved	—
31	MAU	Locked

Table 38.310 Example of SPID Setting (Several SPIDs are Allocated to Each CPU)

SPID value	Bus master	Lock/Unlock
0	CPU0	Unlocked
1	CPU1	Unlocked
2 to 5	Reserved	—
6 to 12	CPU0	Unlocked
13 to 19	CPU1	Unlocked
20 to 23	Reserved	—
24	RHSIF0/HSSPI0/FlexRay0/Ethernet	Locked
25	Reserved	—
26	ICU-M	Locked
27	sDMAC1	Locked
28	sDMAC0	Locked
29	DTS	Locked
30	Reserved	—
31	MAU	Locked

38.5.2.2 List of Registers

Table 38.311 List of SPID Registers (1/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection
SPIDCTL (Peripheral Group9)	Reserved	—	FF0A 8000 _H to FF0A 804B _H	—	
	BusMaster 19 SPID Register	BM19SPID	FF0A 804C _H	8, 16, 32	
	Reserved	—	FF0A 8050 _H	—	
	BusMaster 21 SPID Register	BM21SPID	FF0A 8054 _H	8, 16, 32	
	Reserved	—	FF0A 8058 _H	—	
	BusMaster 23 SPID Register	BM23SPID	FF0A 805C _H	8, 16, 32	
	BusMaster 24 SPID Register	BM24SPID	FF0A 8060 _H	8, 16, 32	
	Reserved	—	FF0A 8064 _H	—	
	BusMaster 26 SPID Register	BM26SPID	FF0A 8068 _H	8, 16, 32	
	Reserved	—	FF0A 806C _H to FF0A 80FF _H	—	
	BusMaster 00 SPID Mask Register	BM00SPIDMSK	FF0A 8100 _H	8, 16, 32	SPIDKCProt
	BusMaster 01 SPID Mask Register	BM01SPIDMSK	FF0A 8104 _H	8, 16, 32	SPIDKCProt
	Reserved	—	FF0A 8108 _H to FF0A 814B _H	—	
	BusMaster 19 SPID Mask Register	BM19SPIDMSK	FF0A 814C _H	8, 16, 32	SPIDKCProt
	Reserved	—	FF0A 8150 _H	—	
	BusMaster 21 SPID Mask Register	BM21SPIDMSK	FF0A 8154 _H	8, 16, 32	SPIDKCProt
	Reserved	—	FF0A 8158 _H	—	
	BusMaster 23 SPID Mask Register	BM23SPIDMSK	FF0A 815C _H	8, 16, 32	SPIDKCProt
	BusMaster 24 SPID Mask Register	BM24SPIDMSK	FF0A 8160 _H	8, 16, 32	SPIDKCProt
	Reserved	—	FF0A 8164 _H	—	
	BusMaster 26 SPID Mask Register	BM26SPIDMSK	FF0A 8168 _H	8, 16, 32	SPIDKCProt
	BusMaster 27 SPID Mask Register	BM27SPIDMSK	FF0A 816C _H	8, 16, 32	SPIDKCProt
	BusMaster 28 SPID Mask Register	BM28SPIDMSK	FF0A 8170 _H	8, 16, 32	SPIDKCProt
	BusMaster 29 SPID Mask Register	BM29SPIDMSK	FF0A 8174 _H	8, 16, 32	SPIDKCProt
	Reserved	—	FF0A 8178 _H	—	
	BusMaster 31 SPID Mask Register	BM31SPIDMSK	FF0A 817C _H	8, 16, 32	SPIDKCProt
	Reserved	—	FF0A 8180 _H to FF0A 81FF _H	—	
	BusMaster 00 SPID Mask Lock Register	BM00SPIDMSKLOCK	FF0A 8200 _H	8, 16, 32	
	BusMaster 01 SPID Mask Lock Register	BM01SPIDMSKLOCK	FF0A 8204 _H	8, 16, 32	
	Reserved	—	FF0A 8208 _H to FF0A 824B _H	—	
	BusMaster 19 SPID Mask Lock Register	BM19SPIDMSKLOCK	FF0A 824C _H	8, 16, 32	
	Reserved	—	FF0A 8250 _H	—	
	BusMaster 21 SPID Mask Lock Register	BM21SPIDMSKLOCK	FF0A 8254 _H	8, 16, 32	
Reserved	—	FF0A 8258 _H	—		
BusMaster 23 SPID Mask Lock Register	BM23SPIDMSKLOCK	FF0A 825C _H	8, 16, 32		

Table 38.311 List of SPID Registers (2/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection
SPIDCTL (Peripheral Group9)	BusMaster 24 SPID Mask Lock Register	BM24SPIDMSKLOCK	FF0A 8260 _H	8, 16, 32	
	Reserved	—	FF0A 8264 _H	—	
	BusMaster 26 SPID Mask Lock Register	BM26SPIDMSKLOCK	FF0A 8268 _H	8, 16, 32	
	BusMaster 27 SPID Mask Lock Register	BM27SPIDMSKLOCK	FF0A 826C _H	8, 16, 32	
	BusMaster 28 SPID Mask Lock Register	BM28SPIDMSKLOCK	FF0A 8270 _H	8, 16, 32	
	BusMaster 29 SPID Mask Lock Register	BM29SPIDMSKLOCK	FF0A 8274 _H	8, 16, 32	
	Reserved	—	FF0A 8278 _H	—	
	BusMaster 31 SPID Mask Lock Register	BM31SPIDMSKLOCK	FF0A 827C _H	8, 16, 32	
	Reserved	—	FF0A 8280 _H to FF0A 82FF _H	—	
	SPID Key Code Protect Register	SPIDKCPROT	FF0A 8300 _H	32	
	Reserved	—	FF0A 8304 _H to FF0A 83FF _H	—	

38.5.2.3 BMnSPID — BusMaster n SPID Register

This register is used to set the SPID of each bus master.

Value after reset: 0000 00XX_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SPID[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	(defined for each bus master)				
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W				

Note 1. See **Section 38.5.2.1, Overview** for the initial value of each SPID.

Table 38.312 BMnSPID Register Contents

Bit Position	Bit Name	Function
31 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	SPID[4:0]	SPID for each bus master.

38.5.2.4 BMnSPIDMSK — BusMaster n SPID Mask Register

This register is used to set the SPID that each bus-master can use.

This register is protected by key-code register.

Value after reset: FFFF FFFF_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPIDMSK[31:16]																
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPIDMSK[15:0]																
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.313 BMnSPIDMSK Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDMSK[31:0]	Set inhibition of SPID value which that each bus master can use. Each bit corresponds to a single SPID value. 0: The bus-master cannot use this SPID. Setting of this SPID is inhibited. 1: The bus-master can use this SPID. Setting of this SPID is not inhibited.

38.5.2.5 BMnSPIDMSKLOCK — BusMaster n SPID Mask Lock Register

This register is used to set write-mask to each BMnSPIDMSK.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LOCK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 38.314 BMnSPIDMSKLOCK Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	LOCK	Lock of register 0: Register (BMnSPIDMSK) can be re-written 1: Any further write to the register (BMnSPIDMSK) is ignored. LOCK can be cleared by Power Up Reset, System Reset1, System Reset2, and Application Reset.

38.5.2.6 SPIDKCPROT — SPID Key Code Protect Register

This register is used to set the key-code lock or unlock of BMnSPIDMSK.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.315 SPIDKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

38.5.2.7 Usage

SPID is used for one of the identifiers of the memory protection/guard functions. SPID setting procedure is as follows.

1. Set the SPID key code protection register (SPIDKCPROT) to enable writing to SPID registers.
2. Set the SPID mask register (BMnSPIDMSK) to define the SPID values that the bus master can set according to system FFI.
3. Set the SPID mask lock register (BMnSPIDMSKLOCK) to prohibit changing the SPID mask register setting for security purposes.
4. Set the value allowed by the SPID mask register (BMnSPIDMSK) to the SPID register (BMnSPID) for the next time the bus master is accessed.
5. Set the SPID key code protection register (SPIDKCPROT) to disable writing to SPID registers.
6. Configure the memory protection/guard settings to prevent erroneous access identified by SPID and other Identifiers.

38.5.3 PEG

38.5.3.1 Overview

Each PE has a guard that controls the access to the Local RAM and INTC1 from other bus masters. Access by bus masters can be controlled via their bus context for four regions through this PE Guard (PEG). Bus masters can only access the Local RAM area and INTC1 register area if permission is granted. PEG can prevent read*¹ and write accesses against which the Local RAM area and INTC1 register area should be protected. If PEG detects illegal access, guard error is reported to ECM.

If the guard enable bit (GEN) of all PEGPROT_m registers is disabled, all masters can access to both the Local RAM area and INTC1 registers.

If the guard enable bit (GEN) of either PEGPROT_m registers is enabled, an access to Local RAM area and INTC1 registers is limited according to that setting.

Note 1. including fetch access.

38.5.3.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as an offset from the base addresses.

Table 38.316 Register Base Addresses

Base Address Name	Base Address	Bus Group
<GUARD_PE0CL0_base>	FFC6 C000 _H	Peripheral Group 0
<GUARD_PE1CL0_base>	FFC6 C100 _H	Peripheral Group 0
<PEGCAP_M_PE0CL0_base>	FFC6 C800 _H	Peripheral Group 0
<PEGCAP_S_PE0CL0_base>	FFC6 C810 _H	Peripheral Group 0
<PEGCAP_M_PE1CL0_base>	FFC6 C820 _H	Peripheral Group 0
<PEGCAP_S_PE1CL0_base>	FFC6 C830 _H	Peripheral Group 0

38.5.3.3 List of Registers

Table 38.317 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
GUARD_PEnCL0 (n = 0, 1)*1	Key Code Protection Register	PEGKCPROT	<GUARD_PEnCL0_base> + 00 _H	32	
	Channel Protection Control Register	PEGPROTm (m = 0 to 3)	<GUARD_PEnCL0_base> + 40 _H + m × 10 _H	8, 16, 32	PEGKCPROT
	Channel SPID Setting Register	PEGSPIDm (m = 0 to 3)	<GUARD_PEnCL0_base> + 44 _H + m × 10 _H	8, 16, 32	PEGKCPROT
	Channel Base Address Setting Register	PEGBADm (m = 0 to 3)	<GUARD_PEnCL0_base> + 48 _H + m × 10 _H	32	PEGKCPROT
	Channel Valid Bit Setting Register	PEGADVm (m = 0 to 3)	<GUARD_PEnCL0_base> + 4C _H + m × 10 _H	32	PEGKCPROT
PEGCAP_M_PEnCL0 (n = 0, 1)	Guard Error Clear Register	PEGERRCLR	<PEGCAP_M_PEnCL0_base> + 00 _H	8, 16, 32	
	Guard Error Status Register	PEGERRSTAT	<PEGCAP_M_PEnCL0_base> + 04 _H	8, 32	
	Guard Error Address Register	PEGERRADDR	<PEGCAP_M_PEnCL0_base> + 08 _H	32	
	Guard Error Access Information Register	PEGERRTYPE	<PEGCAP_M_PEnCL0_base> + 0C _H	16, 32	
PEGCAP_S_PEnCL0 (n = 0, 1)	Guard Error Clear Register	PEGERRCLR	<PEGCAP_S_PEnCL0_base> + 00 _H	8, 16, 32	
	Guard Error Status Register	PEGERRSTAT	<PEGCAP_S_PEnCL0_base> + 04 _H	8, 32	
	Guard Error Address Register	PEGERRADDR	<PEGCAP_S_PEnCL0_base> + 08 _H	32	
	Guard Error Access Information Register	PEGERRTYPE	<PEGCAP_S_PEnCL0_base> + 0C _H	16, 32	

Note 1. It is necessary to give read permission even when fetching instructions from its own Local RAM.

38.5.3.4 PEGKCPROT — Key Code Protection Register

This register is used to unlock/lock key protection of the PE guard register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.318 PEGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

38.5.3.5 PEGPROTm — Channel Protection Control Register

Specifies the PE resource access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the settings other than SPID.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 38.319 PEGPROTm Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, PEGSPIDm is used as a judgment condition. 1: During write, PEGSPIDm is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, PEGSPIDm is used as the judgment condition. 1: During read, PEGSPIDm is not used as a judgment condition.

38.5.3.6 PEGSPIDm — Channel SPID Setting Register

Specifies the PE resource access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the SPID settings.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPID[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.320 PEGSPIDm Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	<p>R/W enable setting based on SPID</p> <p>PEGSPIDm is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits.</p> <p>For example, setting PEGSPIDm to 0101_B enables access to areas SPID = 0 and SPID = 2.</p> <p>0: Reading/writing the area with SPID = m depends on the RG and WG bit setting</p> <p>1: Enables reading/writing the area with SPID = m</p>

38.5.3.7 PEGBADm — Channel Base Address Setting Register

This register is used to define area m when the setting of PEGPROTm and PEGSPIDm is valid. PEGBADm specifies base address of area m and PEGADVm specifies valid bits of PEGBADm.

Value after reset: FC00 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	BAD[25:16]									
Value after reset	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAD[15:12]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.321 PEGBADm Register Contents

Bit Position	Bit Name	Function
31 to 26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25 to 12	BAD[25:12]	Base address
11 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.5.3.8 PEGADV_m — Channel Valid Bit Setting Register

This register is used to define area *m* when the setting of PEGPROT_m and PEGSPID_m is valid. PEGBAD_m specifies base address of area *m* and PEGADV_m specifies valid bits of PEGBAD_m.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ADV[25:16]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADV[15:12]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.322 PEGADV_m Register Contents

Bit Position	Bit Name	Function
31 to 26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25 to 12	ADV[25:12]	Valid bits of PEGBAD _m When an ADV bit is 1, the address of each access will be compared with the corresponding BAD bit of PEGBAD _m . When all ADV bits are set to 1, 4 KB (the minimum unit) is targeted for protection based on the address specified by PEGBAD _m . When all ADV bits are set to 0, the target of protection is the entire Local RAM.
11 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Example: When PEGBAD_m[25:12] is set to 1C00_H and PEGADV_m[25:12] is set to 3FFE_H, PE guard protection area *m* is FDC0 0000_H to FDC0 0FFF_H and FDC0 1000_H to FDC0 1FFF_H.

Commentary: When PEGBAD_m[25:12] is set to 1C00_H, the base address is FDC0 0000_H and the configurable range of bits is enclosed in [] as shown below.

1111 11[01 1100 0000 0000] 0000 0000 0000

F D C 0 0 0 0 0

When PEGADV_m[25:12] is 3FFE_H, bits set to 0 in that range and the 12 lower-order bits are ignored, so we have

1111 11[01 1100 0000 000X] XXXX XXXX XXXX

which means that the 4 KB ranges (a total of 8 KB) from

F D C 0 0 0 0 0 to

F D C 0 0 F F F,

and

F D C 0 1 0 0 0 to

F D C 0 1 F F F

will be protected.

38.5.3.9 PEGERRCLR — Guard Error Clear Register

This register is used to clear the PEGERRSTAT register. The PEGERRSTAT.OVF and/or PEGERRSTAT.ERR bits are cleared immediately after writing this register. 0 is always returned when reading this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.323 PEGERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CLRO	Clears the PEGERRSTAT.OVF bit
0	CLRE	Clears the PEGERRSTAT.ERR bit

38.5.3.10 PEGERRSTAT — Guard Error Status Register

This register is used to notify users whether guard errors occurred. The ERR bit is set when a guard error is reported by an error detection module, etc. The OVF bit is set when a guard error is reported again when the ERR bit is already set. This register is not writable and is cleared when writing the PEGERRCLR register. See **Section 38.5.3.9, PEGERRCLR — Guard Error Clear Register.**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.324 PEGERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	OVF	Overflow status flag
0	ERR	Error status flag

38.5.3.11 PEGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when PEGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.325 PEGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

38.5.3.12 PEGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when PEGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]				—	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.326 PEGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

38.5.4 CRG

38.5.4.1 Overview

The Cluster RAM is protected by a memory guard. Access to the Cluster RAM can be controlled via the bus context for eight regions by this Cluster RAM Guard (CRG). Bus masters can only access the Cluster RAM area if permission is granted. CRG can prevent read*¹ and write accesses against which the Cluster RAM area should be protected. If CRG detects illegal access, guard error notification is signaled to ECM.

If the guard enable bit (GEN) of all CRGPROTm registers is disabled, all masters can access to the Cluster RAM area.

If the guard enable bit (GEN) of either CRGPROTm registers is enabled, an access to Cluster RAM area is limited according to that setting.

Note 1. including fetch access.

38.5.4.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offsets from the base addresses.

Table 38.327 Register Base Addresses

Base Address Name	Base Address	Bus Group
<GUARD_CRAMCL0_base>	FFC6 D000 _H	Peripheral Group 0
<GUARD_CRAMCL1_base>	FFC6 D100 _H	Peripheral Group 0
<CRGCAP_PE0CL0_base>	FFC6 D800 _H	Peripheral Group 0
<CRGCAP_PE1CL0_base>	FFC6 D810 _H	Peripheral Group 0
<CRGCAP_CRAMHCL0_base>	FFC6 D900 _H	Peripheral Group 0
<CRGCAP_CRAMLCL0_base>	FFC6 D910 _H	Peripheral Group 0
<CRGCAP_SX2MBHCL0_base>	FFC6 D920 _H	Peripheral Group 0
<CRGCAP_SX2MBLCL0_base>	FFC6 D930 _H	Peripheral Group 0
<CRGCAP_CRAMHCL1_base>	FFC6 D940 _H	Peripheral Group 0
<CRGCAP_CRAMLCL1_base>	FFC6 D950 _H	Peripheral Group 0
<CRGCAP_SX2MBHCL1_base>	FFC6 D960 _H	Peripheral Group 0
<CRGCAP_SX2MBLCL1_base>	FFC6 D970 _H	Peripheral Group 0

38.5.4.3 List of Registers

Table 38.328 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
GUARD_CRAMCLn (n = 0, 1)	Key Code Protection Register	CRGKCPROT	<GUARD_CRAMCLn_base> + 00H	32	
	Channel Protection Control Register	CRGPROTm (m = 0-7)	<GUARD_CRAMCLn_base> + 10H + m × 10H	8, 16, 32	CRGKCPROT
	Channel SPID Setting Register	CRGSPIDm (m = 0-7)	<GUARD_CRAMCLn_base> + 14H + m × 10H	8, 16, 32	CRGKCPROT
	Channel Base Address Setting Register	CRGBADm (m = 0-7)	<GUARD_CRAMCLn_base> + 18H + m × 10H	8, 16, 32	CRGKCPROT
	Channel Valid Bit Setting Register	CRGADVm (m = 0-7)	<GUARD_CRAMCLn_base> + 1CH + m × 10H	8, 16, 32	CRGKCPROT
CRGCAP_PEnCL0 (n = 0, 1)	Guard Error Clear Register	CRGERRCLR	<CRGCAP_PEnCL0_base> + 00H	8, 16, 32	
	Guard Error Status Register	CRGERRSTAT	<CRGCAP_PEnCL0_base> + 04H	8, 32	
	Guard Error Address Register	CRGERRADDR	<CRGCAP_PEnCL0_base> + 08H	32	
	Guard Error Access Information Register	CRGERRTYPE	<CRGCAP_PEnCL0_base> + 0CH	16, 32	
CRGCAP_CRAMHCLn (n = 0, 1)	Guard Error Clear Register	CRGERRCLR	<CRGCAP_CRAMHCLn_base> + 00H	8, 16, 32	
	Guard Error Status Register	CRGERRSTAT	<CRGCAP_CRAMHCLn_base> + 04H	8, 32	
	Guard Error Address Register	CRGERRADDR	<CRGCAP_CRAMHCLn_base> + 08H	32	
	Guard Error Access Information Register	CRGERRTYPE	<CRGCAP_CRAMHCLn_base> + 0CH	16, 32	
CRGCAP_CRAMLCLn (n = 0, 1)	Guard Error Clear Register	CRGERRCLR	<CRGCAP_CRAMLCLn_base> + 00H	8, 16, 32	
	Guard Error Status Register	CRGERRSTAT	<CRGCAP_CRAMLCLn_base> + 04H	8, 32	
	Guard Error Address Register	CRGERRADDR	<CRGCAP_CRAMLCLn_base> + 08H	32	
	Guard Error Access Information Register	CRGERRTYPE	<CRGCAP_CRAMLCLn_base> + 0CH	16, 32	
CRGCAP_SX2MBHCLn (n = 0, 1)	Guard Error Clear Register	CRGERRCLR	<CRGCAP_SX2MBHCLn_base> + 00H	8, 16, 32	
	Guard Error Status Register	CRGERRSTAT	<CRGCAP_SX2MBHCLn_base> + 04H	8, 32	
	Guard Error Address Register	CRGERRADDR	<CRGCAP_SX2MBHCLn_base> + 08H	32	
	Guard Error Access Information Register	CRGERRTYPE	<CRGCAP_SX2MBHCLn_base> + 0CH	16, 32	
CRGCAP_SX2MBLCLn (n = 0, 1)	Guard Error Clear Register	CRGERRCLR	<CRGCAP_SX2MBLCLn_base> + 00H	8, 16, 32	
	Guard Error Status Register	CRGERRSTAT	<CRGCAP_SX2MBLCLn_base> + 04H	8, 32	
	Guard Error Address Register	CRGERRADDR	<CRGCAP_SX2MBLCLn_base> + 08H	32	
	Guard Error Access Information Register	CRGERRTYPE	<CRGCAP_SX2MBLCLn_base> + 0CH	16, 32	

38.5.4.4 CRGKCPROT — Key Code Protection Register

This register is used to unlock/lock key protection of the Cluster RAM guard register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.329 CRGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

38.5.4.5 CRGPROTm — Channel Protection Control Register

Specifies the Cluster RAM access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the settings other than SPID.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 38.330 CRGPROTm Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, CRGSPIDm is used as a judgment condition. 1: During write, CRGSPIDm is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, CRGSPIDm is used as the judgment condition. 1: During read, CRGSPIDm is not used as a judgment condition.

38.5.4.6 CRGSPIDm — Channel SPID Setting Register

Specifies the Cluster RAM access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the SPID settings.

Value after reset: 0000 0000_H

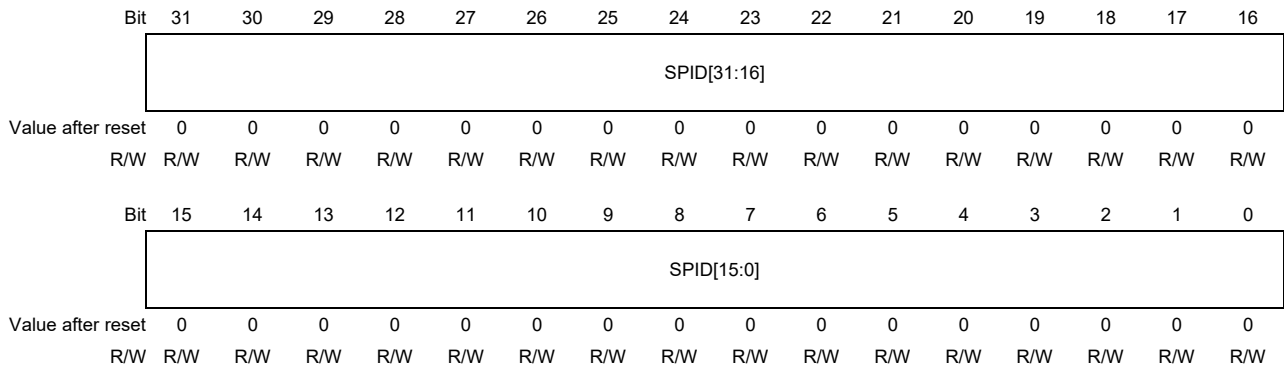


Table 38.331 CRGSPIDm Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	<p>R/W enable setting based on SPID</p> <p>CRGSPIDm is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits.</p> <p>For example, setting CRGSPIDm to 0101_B enables access to areas with SPID = 0 and SPID = 2.</p> <p>0: Reading/writing the area with SPID = m depends on the RG and WG bit setting</p> <p>1: Enables reading/writing the area with SPID = m</p>

38.5.4.7 CRGBADm — Channel Base Address Setting Register

This register is used to define area m when the setting of CRGPROTm and CRGSPIDm is valid. CRGBADm specifies base address of area m and CRGADVm specifies valid bits of CRGBADm.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	BAD[18:16]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAD[15:8]								—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 38.332 CRGBADm Register Contents

Bit Position	Bit Name	Function
31 to 19	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
18 to 8	BAD[18:8]	Base address
7 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

38.5.4.8 CRGADV_m — Channel Valid Bit Setting Register

This register is used to define area *m* when the setting of CRGPROT_m and CRGSPID_m is valid. CRGBAD_m specifies base address of area *m* and CRGADV_m specifies valid bits of CRGBAD_m

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	ADV[18:16]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADV[15:8]								—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 38.333 CRGADV_m Register Contents

Bit Position	Bit Name	Function
31 to 19	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
18 to 8	ADV[18:8]	Valid bits of CRGBAD _m When an ADV bit is 1, the address of each access will be compared with the corresponding BAD bit of CRGBAD _m . When all ADV bits are set to 1, 256 bytes (the minimum unit) are targeted for protection based on the address specified by CRGBAD _m . When all ADV bits are set to 0, the target of protection is the whole of the entire Cluster RAM.
7 to 0	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.

Example: When CRGBAD_m[18:8] is set to 000_H and CRGADV_m[18:8] is set to 7EF_H, Cluster RAM guard protection area *m* is FE00 0000_H to FE00 00FF_H and FE00 1000_H to FE00 10FF_H.

Commentary: When CRGBAD_m[18:8] is set to 000_H, the base address is FE00 0000_H and the configurable range of bits is enclosed in [] as shown below.

1111 1110 0000 0[000 0000 0000] 0000 0000

F E 0 0 0 0 0

When CRGADV_m[18:8] is 7EF_H, bits set to 0 in that range and the 8 lower-order bits are ignored, so we have

1111 1110 0000 0[000 000X 0000] XXXX XXXX

which means that the 256-byte ranges (a total of 512 bytes) from

F E 0 0 0 0 0 0 to

F E 0 0 0 0 F F,

and

F E 0 0 1 0 0 0 to

F E 0 0 1 0 F F

will be protected.

38.5.4.9 CRGERRCLR — Guard Error Clear Register

This register is used to clear the CRGERRSTAT register. The CRGERRSTAT.OVF and/or CRGERRSTAT.ERR bits are cleared immediately after writing this register. Response data is always 0 when reading this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.334 CRGERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CLRO	Clears the CRGERRSTAT.OVF bit
0	CLRE	Clears the CRGERRSTAT.ERR bit

38.5.4.10 CRGERRSTAT — Guard Error Status Register

This register is used to notify users whether guard errors occurred. The ERR bit is set when a guard error is reported by an error detection module, etc. The OVF bit is set when a guard error is reported again when the ERR bit is already set. This register is not writable and is cleared when writing the CRGERRCLR register. See **Section 38.5.4.9, CRGERRCLR — Guard Error Clear Register.**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.335 CRGERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	OVF	Overflow status flag
0	ERR	Error status flag

38.5.4.11 CRGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when CRGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.336 CRGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

38.5.4.12 CRGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when CRGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]				—	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.337 CRGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

38.5.5 INTC2 Guard

38.5.5.1 Overview

INTC2 guard is one of the slave guards used to control the access to INTC2 registers. INTC2 guard can protect respectively the read and write access against INTC2 registers of each channel. If INTC2 guard detects illegal access, guard error notification is signaled to ECM.

38.5.5.2 List of Registers

Table 38.338 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
GUARD_INTC2 (Peripheral Group0)	Key Code Protection Register	INTC2GKCPROT	FFC6 4000 _H	32	
	Guard Error Clear Register	INTC2GERRCLR	FFC6 4004 _H	8, 32	
	Guard Error Status Register	INTC2GERRSTAT	FFC6 4008 _H	8, 32	
	Guard Error Address Register	INTC2GERRADDR	FFC6 400C _H	32	
	Guard Error Access Information Register	INTC2GERRTYPE	FFC6 4010 _H	16, 32	
	SPID Setting Register	INTC2GMPID _n (n = 0-7)	FFC6 4040 _H + n × 04 _H	8, 32	INTC2GKCPROT
	INTC2 Protection Control Register	INTC2GPROT_GR	FFC6 40F0 _H	32	INTC2GKCPROT
	Channel Protection Control Register	INTC2GPROT_m (m = 032-511)	FFC6 4100 _H + m × 04 _H	32	INTC2GKCPROT

Table 38.339 Target Register of Protection Control Register

Symbol	Target Register
INTC2GPROT_m (m = 032 to 511)	EIC _n (n = 32 to 511)
INTC2GPROT_GR	IMR _n (n = 1 to 15), EIBD _n (n = 32 to 511)

38.5.5.3 INTC2GERRCLR — Guard Error Clear Register

This register is used to clear the INTC2GERRSTAT register. The INTC2GERRSTAT.OVF and/or INTC2GERRSTAT.ERR bits are cleared immediately after writing this register. Response data is always 0 when reading this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.340 INTC2GERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CLRO	Clears the INTC2GERRSTAT.OVF bit
0	CLRE	Clears the INTC2GERRSTAT.ERR bit

38.5.5.4 INTC2GERRSTAT — Guard Error Status Register

This register is used to notify users whether guard errors occurred. The ERR bit is set when a guard error is reported by an error detection module, etc. The OVF bit is set when a guard error is reported again when the ERR bit is already set. This register is not writable and is cleared when writing the INTC2GERRCLR register. See **Section 38.5.5.3, INTC2GERRCLR — Guard Error Clear Register.**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.341 INTC2GERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	OVF	Overflow status flag
0	ERR	Error status flag

38.5.5.5 INTC2GERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when INTC2GERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.342 INTC2GERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

38.5.5.6 INTC2GERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when INTC2GERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]				—	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.343 INTC2GERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

38.5.5.7 INTC2GKCPROT — Key Code Protection Register

This register is used for protection against writing to the channel/INTC2 protection control registers due to program malfunction and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.344 INTC2GKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

38.5.5.8 INTC2GMPIDn — SPID Setting Register

INTC2GMPIDn are used to specify a set of SPIDs, which are referred by INTC2GPROT_m/GR registers. Access protection can be controlled by combination of INTC2GMPIDn registers and INTC2GPROT_m/GR register. For example, masters with SPID=1 or SPID=7 can access to global registers when setting INTC2GPROT_GR.MPID=00000101_B, INTC2GMPID0.SPID=01_H and INTC2GMPID2.SPID=07_H.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SPID[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 38.345 INTC2GMPIDn Register Contents

Bit Position	Bit Name	Function
31 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	SPID[4:0]	SPID setting

38.5.5.9 INTC2GPROT_m/GR — Channel/INTC2 Protection Control Register

Specifies the INTC2 slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	MPID[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 38.346 INTC2GPROT_m/GR Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	MPID[7:0]	R/W enable setting based on MPID MPID is a list of bits each representing one MPID value. Multiple MPID values are enabled simultaneously by setting multiple bits. For example, setting MPID to 0101 _B enables access to areas MPID = 0 and MPID = 2. 0: Reading/writing the area with MPID = m depends on the RG and WG bit setting 1: Enables reading/writing the area with MPID = m
15 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, MPID is used as a judgment condition. 1: During write, MPID is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, MPID is used as the judgment condition. 1: During read, MPID is not used as a judgment condition.

38.5.6 DTS Guard

38.5.6.1 Overview

DTS guard is one of the slave guards used to control the access to the registers for DTS transfer. DTS guard can prevent read and write access to the DTS registers of each channel. If DTS guard detects illegal access, guard error notification is signaled to ECM.

38.5.6.2 List of Registers

Table 38.347 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
GUARD_DTS (Peripheral Group0)	Key Code Protection Register	DTSGKCPROT	FFC6 8000 _H	32	
	Guard Error Clear Register	DTSGERRCLR	FFC6 8004 _H	8, 32	
	Guard Error Status Register	DTSGERRSTAT	FFC6 8008 _H	8, 32	
	Guard Error Address Register	DTSGERRADDR	FFC6 800C _H	32	
	Guard Error Access Information Register	DTSGERRTYPE	FFC6 8010 _H	16, 32	
	SPID Setting Register	DTSGMPID _n (n = 0-7)	FFC6 8040 _H + n × 04 _H	8, 32	DTSGKCPROT
	DTS Protection Control Register	DTSGPROT_GR	FFC6 80F0 _H	32	DTSGKCPROT
	Channel Protection Control Register	DTSGPROT_m (m = 000-127)	FFC6 8100 _H + m × 04 _H	32	DTSGKCPROT

Table 38.348 Target Register of Protection Control Register

Symbol	Target Register
DTSGPROT_GR	DTSCTL1, DTSCTL2, DTSSTS, DTSER, DTSPR0, DTSPR1, DTSPR2, DTSPR3, DTSPR4, DTSPR5, DTSPR6, DTSPR7, DTS _{nnn} CM(nnn = 000 to 127)
DTSGPROT_m (m = 000 to 127)	DTSA _{nnn} , DTD _{nnn} , DTT _{nnn} , DTTCT _{nnn} , DTRS _{nnn} , DTRDA _{nnn} , DTRTC _{nnn} , DTTCC _{nnn} , DTFSL _{nnn} , DTFST _{nnn} , DTFSS _{nnn} , DTFSC _{nnn} (nnn = 000 to 127)

38.5.6.3 DTSGERRCLR — Guard Error Clear Register

This register is used to clear the DTSGERRSTAT register. The DTSGERRSTAT.OVF and/or DTSGERRSTAT.ERR bits are cleared immediately after writing this register. Response data is always 0 when reading this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.349 DTSGERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CLRO	Clears the DTSGERRSTAT.OVF bit
0	CLRE	Clears the DTSGERRSTAT.ERR bit

38.5.6.4 DTSGERRSTAT — Guard Error Status Register

This register is used to notify users whether guard errors occurred. The ERR bit is set when a guard error is reported by an error detection module, etc. The OVF bit is set when a guard error is reported again when the ERR bit is already set. This register is not writable and is cleared when writing the DTSGERRCLR register. See **Section 38.5.6.3, DTSGERRCLR — Guard Error Clear Register.**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.350 DTSGERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	OVF	Overflow status flag
0	ERR	Error status flag

38.5.6.5 DTSGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when DTSGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.351 DTSGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

38.5.6.6 DTSGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when DTSGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]				—	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.352 DTSGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

38.5.6.7 DTSGKCPROT — Key Code Protection Register

This register is used for protection against writing to the channel/DTS protection control registers due to program malfunction and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.353 DTSGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

38.5.6.8 DTSGMPIDn — SPID Setting Register

DTSGMPIDn are used to specify a set of SPIDs, which are referred by DTSGPROT_m/_GR registers. Access protection can be controlled by combination of DTSGMPIDn registers and DTSGPROT_m/GR register. For example, masters with SPID=1 or SPID=7 can access to global registers when setting DTSGPROT_GR.MPID=00000101_B, DTSGMPID0.SPID=01_H and DTSGMPID2.SPID=07_H.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SPID[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 38.354 DTSGMPIDn Register Contents

Bit Position	Bit Name	Function
31 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	SPID[4:0]	SPID setting

38.5.6.9 DTSGPROT_m/GR — Channel/DTS Protection Control Register

Specifies the DTS slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	MPID[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 38.355 DTSGPROT_m/GR Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	MPID[7:0]	R/W enable setting based on MPID MPID is a list of bits each representing one MPID value. Multiple MPID values are enabled simultaneously by setting multiple bits. For example, setting MPID to 0101 _B enables access to areas with MPID = 0 and MPID = 2. 0: Reading/writing the area with MPID = m depends on the RG and WG bit setting 1: Enables reading/writing the area with MPID = m
15 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, MPID is used as a judgment condition. 1: During write, MPID is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, MPID is used as the judgment condition. 1: During read, MPID is not used as a judgment condition.

38.5.7 sDMAC Guard

38.5.7.1 Overview

sDMAC guard is one of the slave guards used to control the access to the registers for sDMAC transfer. sDMAC guard can prevent read and write access to the sDMAC registers of each channel. If sDMAC guard detects illegal access, guard error notification is signaled to ECM.

38.5.7.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offsets from the base addresses.

Table 38.356 Register Base Addresses

Base Address Name	Base Address	Bus Group
<GUARD_DMACH_base>	FFC6 9000 _H	Peripheral Group 0
<GUARD_DMACH1_base>	FFC6 9400 _H	Peripheral Group 0

38.5.7.3 List of Registers

Table 38.357 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
GUARD_DMACH (j = 0, 1)	Key Code Protection Register	DMAGKCPROT	<GUARD_DMACH_base> + 00 _H	32	
	Guard Error Clear Register	DMAGERRCLR	<GUARD_DMACH_base> + 04 _H	8, 16, 32	
	Guard Error Status Register	DMAGERRSTAT	<GUARD_DMACH_base> + 08 _H	8, 32	
	Guard Error Address Register	DMAGERRADDR	<GUARD_DMACH_base> + 0C _H	32	
	Guard Error Access Information Register	DMAGERRTYPE	<GUARD_DMACH_base> + 10 _H	16, 32	
	SPID Setting Register	DMAGMPIDn (n = 0-7)	<GUARD_DMACH_base> + 40 _H + n × 04 _H	8, 16, 32	DMAGKCPROT
	DMA Protection Control Register	DMAGPROT_GR	<GUARD_DMACH_base> + F0 _H	8, 16, 32	DMAGKCPROT
	DMA Descriptor Protection Control Register	DMAGPROT_DP	<GUARD_DMACH_base> + F4 _H	8, 16, 32	DMAGKCPROT
	Channel Protection Control Register	DMAGPROT_m (m = 000-015)	<GUARD_DMACH_base> + 100 _H + m × 04 _H	8, 16, 32	DMAGKCPROT

Table 38.358 Target Register of Protection Control Register

Symbol	Target Register
DMAGPROT_GR	sDMAC global registers
DMAGPROT_DP	Descriptor RAM
DMAGPROT_m (m = 000 to 015)	sDMAC channel registers

38.5.7.4 DMAGERRCLR — Guard Error Clear Register

This register is used to clear the DMAGERRSTAT register. The DMAGERRSTAT.OVF and/or DMAGERRSTAT.ERR bits are cleared immediately after writing this register. Response data is always 0 when reading this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.359 DMAGERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CLRO	Clears the DMAGERRSTAT.OVF bit
0	CLRE	Clears the DMAGERRSTAT.ERR bit

38.5.7.5 DMAGERRSTAT — Guard Error Status Register

This register is used to notify users whether guard errors occurred. The ERR bit is set when a guard error is reported by an error detection module, etc. The OVF bit is set when a guard error is reported again when the ERR bit is already set. This register is not writable and is cleared when writing the DMAGERRCLR register. See **Section 38.5.7.4, DMAGERRCLR — Guard Error Clear Register.**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.360 DMAGERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	OVF	Overflow status flag
0	ERR	Error status flag

38.5.7.6 DMAGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when DMAGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.361 DMAGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

38.5.7.7 DMAGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when DMAGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]				—	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.362 DMAGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

38.5.7.8 DMAGKCPROT — Key Code Protection Register

This register is used for protection against writing to the channel/DMA/DMA descriptor protection control registers due to program malfunction and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.363 DMAGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

38.5.7.9 DMAGMPIDn — SPID Setting Register

DMAGMPIDn are used to specify a set of SPIDs, which are referred by DMAGPROT_m/GR/DP registers. Access protection can be controlled by combination of DMAGMPIDn registers and DMAGPROT_m/GR/DP register. For example, masters with SPID=1 or SPID=7 can access to global registers when setting DMAGPROT_GR.MPID=00000101_B, DMAGMPID0.SPID=01_H and DMAGMPID2.SPID=07_H.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SPID[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 38.364 DMAGMPIDn Register Contents

Bit Position	Bit Name	Function
31 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	SPID[4:0]	SPID setting

38.5.7.10 DMAGPROT_m/GR/DP — Channel/DMA/DMA Descriptor Protection Control Register

Specifies the sDMAC slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	MPID[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 38.365 DMAGPROT_m/GR/DP Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	MPID[7:0]	R/W enable setting based on MPID MPID is a list of bits each representing one MPID value. Multiple MPID values are enabled simultaneously by setting multiple bits. For example, setting MPID to 0101 _B enables access to areas with MPID = 0 and MPID = 2. 0: Reading/writing the area with MPID = m depends on the RG and WG bit setting 1: Enables reading/writing the area with MPID = m
15 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, MPID is used as a judgment condition. 1: During write, MPID is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, MPID is used as the judgment condition. 1: During read, MPID is not used as a judgment condition.

38.5.8 IBG

38.5.8.1 Overview

IPIR, MEV, BarrierSync and TPTM connect to Inter-cluster Bus (I-Bus). I-Bus Guard (IBG) is one of the slave guards used to control the access to these I-Bus modules. IBG can prevent read and write access to the registers of each module. If IBG detects illegal access, guard error notification is signaled to ECM.

38.5.8.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offsets from the base addresses.

Table 38.366 Register Base Addresses

Base Address Name	Base Address	Bus Group
<GUARD_IPIR_base>	FFC6 A200 _H	Peripheral Group 0
<GUARD_MEV_base>	FFC6 A400 _H	Peripheral Group 0
<GUARD_BARR_base>	FFC6 A000 _H	Peripheral Group 0
<GUARD_TPTM_base>	FFC6 A600 _H	Peripheral Group 0

38.5.8.3 List of Registers

Table 38.367 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection
GUARD_IPIR	Guard Error Clear Register	IPIGERRCLR	<GUARD_IPIR_base> + 00 _H	8, 32	
	Guard Error Status Register	IPIGERRSTAT	<GUARD_IPIR_base> + 04 _H	8, 32	
	Guard Error Address Register	IPIGERRADDR	<GUARD_IPIR_base> + 08 _H	32	
	Guard Error Access Information Register	IPIGERRTYPE	<GUARD_IPIR_base> + 0C _H	16, 32	
	Key Code Protection Register	IPIGKCPROT	<GUARD_IPIR_base> + 10 _H	32	
	Channel Protection Control Register	IPIGPROT0_n (n = 0-3)	<GUARD_IPIR_base> + 80 _H + n × 08 _H	8, 16, 32	IPIGKCPROT
	Channel SPID Setting Register	IPIGPROT1_n (n = 0-3)	<GUARD_IPIR_base> + 84 _H + n × 08 _H	8, 16, 32	IPIGKCPROT
	Common Protection Control Register	IPIGPROT0_4	<GUARD_IPIR_base> + A0 _H	8, 16, 32	IPIGKCPROT
	Common SPID Setting Register	IPIGPROT1_4	<GUARD_IPIR_base> + A4 _H	8, 16, 32	IPIGKCPROT
GUARD_MEV	Guard Error Clear Register	MEGERRCLR	<GUARD_MEV_base> + 00 _H	8, 32	
	Guard Error Status Register	MEGERRSTAT	<GUARD_MEV_base> + 04 _H	8, 32	
	Guard Error Address Register	MEGERRADDR	<GUARD_MEV_base> + 08 _H	32	
	Guard Error Access Information Register	MEGERRTYPE	<GUARD_MEV_base> + 0C _H	16, 32	
	Key Code Protection Register	MEGKCPROT	<GUARD_MEV_base> + 10 _H	32	
	Channel Protection Control Register	MEGPROT0_n (n = 0-7)	<GUARD_MEV_base> + 80 _H + n × 08 _H	8, 16, 32	MEGKCPROT
	Channel SPID Setting Register	MEGPROT1_n (n = 0-7)	<GUARD_MEV_base> + 84 _H + n × 08 _H	8, 16, 32	MEGKCPROT
GUARD_BARR	Guard Error Clear Register	BRGERRCLR	<GUARD_BARR_base> + 00 _H	8, 32	
	Guard Error Status Register	BRGERRSTAT	<GUARD_BARR_base> + 04 _H	8, 32	
	Guard Error Address Register	BRGERRADDR	<GUARD_BARR_base> + 08 _H	32	

Table 38.367 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection
GUARD_BARR	Guard Error Access Information Register	BRGERRTYPE	<GUARD_BARR_base> + 0C _H	16, 32	
	Key Code Protection Register	BRGKCPROT	<GUARD_BARR_base> + 10 _H	32	
	Channel Protection Control Register	BRGPROT0_n (n = 0-15)	<GUARD_BARR_base> + 80 _H + n × 08 _H	8, 16, 32	BRGKCPROT
	Channel SPID Setting Register	BRGPROT1_n (n = 0-15)	<GUARD_BARR_base> + 84 _H + n × 08 _H	8, 16, 32	BRGKCPROT
	Common Protection Control Register	BRGPROT0_16	<GUARD_BARR_base> + 100 _H	8, 16, 32	BRGKCPROT
	Common SPID Setting Register	BRGPROT1_16	<GUARD_BARR_base> + 104 _H	8, 16, 32	BRGKCPROT
GUARD_TPTM	Guard Error Clear Register	TPTGERRCLR	<GUARD_TPTM_base> + 00 _H	8, 32	
	Guard Error Status Register	TPTGERRSTAT	<GUARD_TPTM_base> + 04 _H	8, 32	
	Guard Error Address Register	TPTGERRADDR	<GUARD_TPTM_base> + 08 _H	32	
	Guard Error Access Information Register	TPTGERRTYPE	<GUARD_TPTM_base> + 0C _H	16, 32	
	Key Code Protection Register	TPTGKCPROT	<GUARD_TPTM_base> + 10 _H	32	
	Channel Protection Control Register	TPTGPROT0_n (n = 0, 1)	<GUARD_TPTM_base> + 80 _H + n × 08 _H	8, 16, 32	TPTGKCPROT
	Channel SPID Setting Register	TPTGPROT1_n (n = 0, 1)	<GUARD_TPTM_base> + 84 _H + n × 08 _H	8, 16, 32	TPTGKCPROT

Table 38.368 Target Register of Protection Control Register

Related Function	Symbol	Target Register
IPIR	IPIGPROT0_n (n = 0 to 3)	IPIInENS, IPIInFLGS, IPIInFCLRS, IPIInREQS, IPIInRCLRS *1
	IPIGPROT0_4	IPIInENm, IPIInFLGm, IPIInFCLRm, IPIInREQm, IPIInRCLRm *1
MEV	MEGPROT0_n (n = 0 to 7)	G0MEV(4n) to G0MEV(4n+3) (n = 0 to 7)
BARR	BRGPROT0_n (n = 0 to 15)	BRnINIT, BRnEN, BRnCHKS, BRnSYNCS *2
	BRGPROT0_16	BRnCHKm, BRnSYNCm *2
TPTM	TPTGPROT0_n (n = 0, 1)	TPTMnIRUN, TPTMnIRRUN, TPTMnIISTP, TPTMnIISTR, TPTMnIIEN, TPTMnIUSTR, TPTMnIDIV, TPTMnIFRUN, TPTMnIFRRUN, TPTMnIFSTP, TPTMnIFSTR, TPTMnIFDIV, TPTMnICNT0, TPTMnILD0, TPTMnICNT1, TPTMnILD1, TPTMnICNT2, TPTMnILD2, TPTMnICNT3, TPTMnILD3, TPTMnFCNT *3*4

Note 1. n = 0 to 3: n is the channel number of IPIR. m = 0 to 1: m is PE number.

Note 2. n = 0 to 15: n is the channel number of Barrier-Synchronization. m = 0 to 1: m is PE number.

Note 3. n = 0 to 1: n is PE number.

Note 4. TPTM Self region registers are protected according to TPTGPROT0_n setting.

38.5.8.4 IPIGERRCLR — Guard Error Clear Register

This register is used to clear the IPIGERRSTAT register. The IPIGERRSTAT.OVF and/or IPIGERRSTAT.ERR bits are cleared immediately after writing this register. Response data is always 0 when reading this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.369 IPIGERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CLRO	Clears the IPIGERRSTAT.OVF bit
0	CLRE	Clears the IPIGERRSTAT.ERR bit

38.5.8.5 IPIGERRSTAT — Guard Error Status Register

This register is used to notify users whether guard errors occurred. The ERR bit is set when a guard error is reported by an error detection module, etc. The OVF bit is set when a guard error is reported again when the ERR bit is already set. This register is not writable and is cleared when writing the IPIGERRCLR register. See **Section 38.5.8.4, IPIGERRCLR — Guard Error Clear Register.**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.370 IPIGERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	OVF	Overflow status flag
0	ERR	Error status flag

38.5.8.6 IPIGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when IPIGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.371 IPIGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

38.5.8.7 IPIGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when IPIGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]				—	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.372 IPIGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

38.5.8.8 IPIGKCPROT — Key Code Protection Register

This register is used for protection against writing to the channel/common protection control registers due to program malfunction and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.373 IPIGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

38.5.8.9 IPIGPROT0_n/4 — Channel/Common Protection Control Register (n = 0 to 3)

Specifies the IPIR slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the settings other than SPID.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 38.374 IPIGPROT0_n/4 Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, IPIGPROT1_n/4 is used as a judgment condition. 1: During write, IPIGPROT1_n/4 is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, IPIGPROT1_n/4 is used as the judgment condition. 1: During read, IPIGPROT1_n/4 is not used as a judgment condition.

38.5.8.10 IPIGPROT1_n/4 — Channel/Common SPID Setting Register (n = 0 to 3)

Specifies the IPIR slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the SPID settings.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPID[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.375 IPIGPROT1_n/4 Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	<p>R/W enable setting based on SPID</p> <p>IPIGPROT1_n/4 is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits.</p> <p>For example, setting IPIGPROT1_n/4 to 0101_B enables access to areas with SPID = 0 and SPID = 2.</p> <p>0: Reading/writing the area with SPID = m depends on the RG and WG bit setting</p> <p>1: Enables reading/writing the area with SPID = m</p>

38.5.8.11 MEGERRCLR — Guard Error Clear Register

This register is used to clear the MEGERRSTAT register. The MEGERRSTAT.OVF and/or MEGERRSTAT.ERR bits are cleared immediately after writing this register. Response data is always 0 when reading this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.376 MEGERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CLRO	Clears the MEGERRSTAT.OVF bit
0	CLRE	Clears the MEGERRSTAT.ERR bit

38.5.8.12 MEGERRSTAT — Guard Error Status Register

This register is used to notify users whether guard errors occurred. The ERR bit is set when a guard error is reported by an error detection module, etc. The OVF bit is set when a guard error is reported again when the ERR bit is already set. This register is not writable and is cleared when writing the MEGERRCLR register. See **Section 38.5.8.11, MEGERRCLR — Guard Error Clear Register.**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.377 MEGERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	OVF	Overflow status flag
0	ERR	Error status flag

38.5.8.13 MEGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when MEGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.378 MEGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

38.5.8.14 MEGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when MEGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]				—	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.379 MEGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	WRITE	Access type of read or write to the target slave when a guard has error occurred

38.5.8.15 MEGKCPROT — Key Code Protection Register

This register is used for protection against writing to the channel protection control registers due to program malfunction and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.380 MEGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

38.5.8.16 MEGPROT0_n — Channel Protection Control Register

Specifies the MEV slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the settings other than SPID.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 38.381 MEGPROT0_n Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, MEGPROT1_n is used as a judgment condition. 1: During write, MEGPROT1_n is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, MEGPROT1_n is used as the judgment condition. 1: During read, MEGPROT1_n is not used as a judgment condition.

38.5.8.17 MEGPROT1_n — Channel SPID Setting Register

Specifies the MEV slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the SPID settings.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPID[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.382 MEGPROT1_n Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	<p>R/W enable setting based on SPID</p> <p>MEGPROT1_n is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits.</p> <p>For example, setting MEGPROT1_n to 0101_B enables access to areas with SPID = 0 and SPID = 2.</p> <p>0: Reading/writing the area with SPID = m depends on the RG and WG bit setting</p> <p>1: Enables reading/writing the area with SPID = m</p>

38.5.8.18 BRGERRCLR — Guard Error Clear Register

This register is used to clear the BRGERRSTAT register. The BRGERRSTAT.OVF and/or BRGERRSTAT.ERR bits are cleared immediately after writing this register. Response data is always 0 when reading this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.383 BRGERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CLRO	Clears the BRGERRSTAT.OVF bit
0	CLRE	Clears the BRGERRSTAT.ERR bit

38.5.8.19 BRGERRSTAT — Guard Error Status Register

This register is used to notify users whether guard errors occurred. The ERR bit is set when a guard error is reported by an error detection module, etc. The OVF bit is set when a guard error is reported again when the ERR bit is already set. This register is not writable and is cleared when writing the BRGERRCLR register. See **Section 38.5.8.18, BRGERRCLR — Guard Error Clear Register.**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.384 BRGERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	OVF	Overflow status flag
0	ERR	Error status flag

38.5.8.20 BRGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when BRGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.385 BRGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

38.5.8.21 BRGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when BRGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]				—	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.386 BRGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

38.5.8.22 BRGKCPROT — Key Code Protection Register

This register is used for protection against writing to the channel/common protection control registers due to program malfunction and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.387 BRGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

38.5.8.23 BRGPROT0_n/16 — Channel/Common Protection Control Register (n = 0 to 15)

Specifies the BarrierSync slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the settings other than SPID.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 38.388 BRGPROT0_n/16 Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, BRGPROT1_n/16 is used as a judgment condition. 1: During write, BRGPROT1_n/16 is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, BRGPROT1_n/16 is used as the judgment condition. 1: During read, BRGPROT1_n/16 is not used as a judgment condition.

38.5.8.24 BRGPROT1_n/16 — Channel/Common SPID Setting Register (n = 0 to 15)

Specifies the BarrierSync slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the SPID settings.

Value after reset: 0000 0000_H

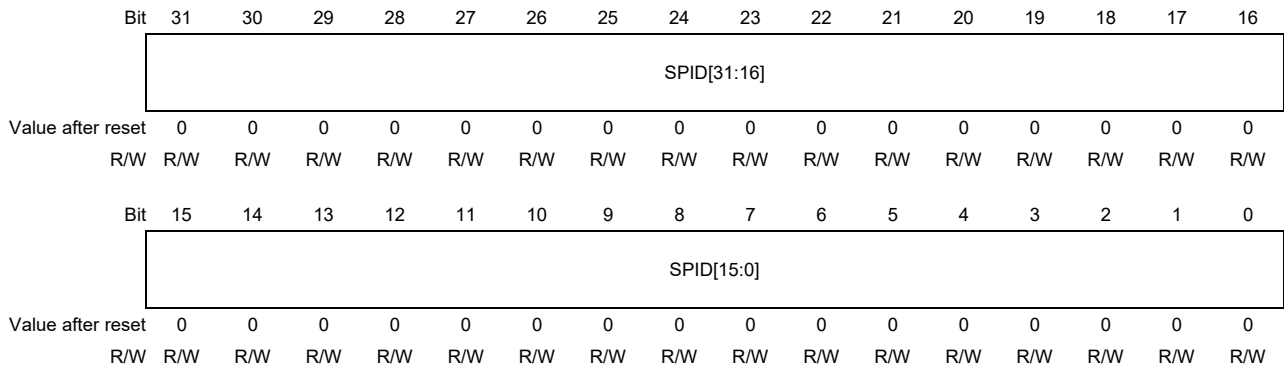


Table 38.389 BRGPROT1_n/16 Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	R/W enable setting based on SPID BRGPROT1_n/16 is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits. For example, setting BRGPROT1_n/16 to 0101 _B enables access to areas with SPID = 0 and SPID = 2. 0: Reading/writing the area with SPID = m depends on the RG and WG bit setting 1: Enables reading/writing the area with SPID = m

38.5.8.25 TPTGERRCLR — Guard Error Clear Register

This register is used to clear the TPTGERRSTAT register. The TPTGERRSTAT.OVF and/or TPTGERRSTAT.ERR bits are cleared immediately after writing this register. Response data is always 0 when reading this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.390 TPTGERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CLRO	Clears the TPTGERRSTAT.OVF bit
0	CLRE	Clears the TPTGERRSTAT.ERR bit

38.5.8.26 TPTGERRSTAT — Guard Error Status Register

This register is used to notify users whether guard errors occurred. The ERR bit is set when a guard error is notified by an error detection module, etc. The OVF bit is set when a guard error is notified again when the ERR bit is already set. This register is not writable and is cleared when writing the TPTGERRCLR register. See **Section 38.5.8.25, TPTGERRCLR — Guard Error Clear Register.**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.391 TPTGERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	OVF	Overflow status flag
0	ERR	Error status flag

38.5.8.27 TPTGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when TPTGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.392 TPTGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

38.5.8.28 TPTGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when TPTGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]				—	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.393 TPTGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

38.5.8.29 TPTGKCPROT — Key Code Protection Register

This register is used for protection against writing to the channel protection control registers due to program malfunction and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.394 TPTGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

38.5.8.30 TPTGPROT0_n — Channel Protection Control Register

Specifies the TPTM slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the settings other than SPID.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 38.395 TPTGPROT0_n Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, TPTGPROT1_n is used as a judgment condition. 1: During write, TPTGPROT1_n is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, TPTGPROT1_n is used as the judgment condition. 1: During read, TPTGPROT1_n is not used as a judgment condition.

38.5.8.31 TPTGPROT1_n — Channel SPID Setting Register

Specifies the TPTM slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the SPID settings.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPID[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.396 TPTGPROT1_n Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	<p>R/W enable setting based on SPID</p> <p>TPTGPROT1_n is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits.</p> <p>For example, setting TPTGPROT1_n to 0101_B enables access to areas with SPID = 0 and SPID = 2.</p> <p>0: Reading/writing the area with SPID = m depends on the RG and WG bit setting</p> <p>1: Enables reading/writing the area with SPID = m</p>

38.5.9 PBG

38.5.9.1 Overview

The PBG module is divided into several PBG groups, each of which is provided with a maximum of 16 protection channels. A single PBG channel can designate the access against which a single peripheral circuit should be protected. Each PBG group can hold the information of the access that has been rejected.

The following table lists the peripheral circuits to be protected, the corresponding PBG group names, and the PBG channel numbers.

Table 38.397 List of Peripheral Circuit Modules to be Protected (1/5)

PBG Group	PBG Channel Number	Module to be Protected
PBG00	0	ECC modules of Peripheral Group 0 (Group A) *3
	1	ECC modules of Peripheral Group 0 (Group B) *4
	3	MISG
	4	PFSYSTEM
	5	GCFU
PBG10	0	ECC module of Data Flash
	1	ECC modules of Peripheral Group 1
PBG2L0	0	SCI30
	1	SCI31
	2	SCI32
	3	SCI33
	4	RHSB XBAR0
	5	RHSB XBAR1
PBG2H0	0	ATU-V (Common, Prescaler)
	1	ATU-V (Timer A)
	2	ATU-V (Timer B)
	3	ATU-V (Timer C)
	4	ATU-V (Timer D)
	5	ATU-V (Timer E)
	6	ATU-V (Timer F)
	7	ATU-V (Timer G)
	8	ATU-V (Trigger)
	9	PIC20
	10	PIC21
	11	PIC22
	12	GTM0
	13	GTM0_1
	14	ENCA0
15	ENCA1	
PBG2H1	0	ECC modules of Peripheral Group 2H

Table 38.397 List of Peripheral Circuit Modules to be Protected (2/5)

PBG Group	PBG Channel Number	Module to be Protected
PBG30	0	PSI5-S
	1	RS-CANFD0
	2	CSIH4
	3	CSIH5
	4	ECC modules of Peripheral Group 3
PBG40	0	CSIH0
	1	CSIH1
	2	CSIH2
	3	CSIH3
	4	ECC modules of Peripheral Group 4
PBG50	0	DFE
	1	ECC modules of Peripheral Group 5
PBG60	0	ECC modules of Peripheral Group 6
	1	RHSB0
	2	RHSB1
	3	RSENT0
	4	RSENT1
	5	RSENT2
	6	RSENT3
	7	RSENT4
	8	RSENT5
	9	RSENT6
	10	RSENT7
	11	RSENT8
	12	RSENT9
	13	RSENT10
	14	RSENT11
	15	RSENT12
PBG61	0	RSENT13
	1	RSENT14
	2	RSENT15
	3	RSENT16
	4	EINT
	5	ECM (M)
	6	ECM (C)
	7	ECM (Common)
	8	Flash protection *2
	9	Flash protection *2
	10	Operating Modes, Reset Controller, Power Supply Voltage Monitor, Clock Controller, Standby Controller, Clock Monitor

Table 38.397 List of Peripheral Circuit Modules to be Protected (3/5)

PBG Group	PBG Channel Number	Module to be Protected
PBG62	0	Port group 00 (Group A) *1
	1	Port group 01 (Group A) *1
	2	Port group 02 (Group A) *1
	3	Port group 10 (Group A) *1
	4	Port group 11 (Group A) *1
	5	Port group 12 (Group A) *1
	6	Port group 13 (Group A) *1
	7	Port group 14 (Group A) *1
	8	Port group 15 (Group A) *1
	9	Port group 20 (Group A) *1
	10	Port group 21 (Group A) *1
	11	Port group 22 (Group A) *1
	12	Port group 23 (Group A) *1
	13	Port group 24 (Group A) *1
	14	Port group 25 (Group A) *1
15	Port group 27 (Group A) *1	
PBG63	0	Port group 30 (Group A) *1
	1	Port group 32 (Group A) *1
	2	Port group 33 (Group A) *1
	3	Port group 34 (Group A) *1
	4	Port group 40 (Group A) *1
	5	Port group 41 (Group A) *1
	6	Port group 42 (Group A) *1
	7	Port group 43 (Group A) *1
	8	Port group 50 (Group A) *1
	9	Port group 51 (Group A) *1
	10	Port group 52 (Group A) *1
	11	Port group 53 (Group A) *1
	12	Port group 54 (Group A) *1
	13	Port group 55 (Group A) *1
	14	Port group 56 (Group A) *1
15	Port group 57 (Group A) *1	

Table 38.397 List of Peripheral Circuit Modules to be Protected (4/5)

PBG Group	PBG Channel Number	Module to be Protected
PBG64	0	Port group 00 (Group B) *1
	1	Port group 01 (Group B) *1
	2	Port group 02 (Group B) *1
	3	Port group 10 (Group B) *1
	4	Port group 11 (Group B) *1
	5	Port group 12 (Group B) *1
	6	Port group 13 (Group B) *1
	7	Port group 14 (Group B) *1
	8	Port group 15 (P15_0 to P15_3, Group B) *1
	9	Port group 15 (P15_4 to P15_7, Group B) *1
	10	Port group 20 (P20_0 to P20_3, Group B) *1
	11	Port group 20 (P20_4 to P20_7, Group B) *1
	12	Port group 21 (Group B) *1
	13	Port group 22 (Group B) *1
	14	Port group 23 (Group B) *1
15	Port group 24 (Group B) *1	
PBG65	0	Port group 25 (Group B) *1
	1	Port group 27 (Group B) *1
	2	Port group 30 (Group B) *1
	3	Port group 32 (Group B) *1
	4	Port group 33 (Group B) *1
	5	Port group 34 (Group B) *1
	6	Port group 40 (Group B) *1
	7	Port group 41 (Group B) *1
	8	Port group 42 (Group B) *1
	9	Port group 43 (Group B) *1
	10	Port group 50 (Group B) *1
	11	Port group 51 (Group B) *1
	12	Port group 52 (Group B) *1
	13	Port group 53 (Group B) *1
	14	Port group 54 (Group B) *1
15	Port group 55 (Group B) *1	
PBG66	0	Port group 56 (Group B) *1
	1	Port group 57 (Group B) *1
	2	Port Keycode, Port Write enable, LVDS registers
	3	JTAG Port group 0 (Group A) *1
	4	JTAG Port group 0 (Group B) *1
	5	Alternative Functions DNF
	6	Edge Detection DNF

Table 38.397 List of Peripheral Circuit Modules to be Protected (5/5)

PBG Group	PBG Channel Number	Module to be Protected
PBG70	0	ECC modules of Peripheral Group 7
	1	ADC0
	2	ADC2
	3	ADC0, 2 (IFC)
	4	ADC (ASF)
	5	ADC (AVSEG)
	6	ADC (ABFG)
	7	ADC (AIR)
	8	DSADC (shared register)
	9	DSADC (CH00 register)
	10	DSADC (CH10 register)
	11	DSADC (CH20 register)
	12	DSADC (CH12 register)
	13	Cyclic ADC (shared register)
14	Cyclic ADC (CH00 register)	
PBG80	0	ECC modules of Peripheral Group 8
	1	ADC1
	2	ADC3
	3	ADC1, 3 (IFC)
	4	DSADC (CH13 register)
5	DSADC (CH11 register)	
PBG90	0	RHSIF
	1	HSSPI
	2	INTIF, DMATRGSEL
	3	ECC modules of Peripheral Group 9

Note 1. The port group registers are divided into the following two groups and controlled separately.

Group A: Registers except PCR_n_m/JPCR_n_m registers (see **Section 2.4.1, Table 2.11**)

Group B: PCR_n_m/JPCR_n_m

Note 2. For details of the target registers, *the RH850/E2x-FCC1 Flash Memory User's Manual: Hardware Interface, Section 4.2, Registers Related to Write and Erase Protection of Flash Memory.*

Note 3. This PBG channel protects the registers related to the following modules:

ECCCNT_* (except for ECCCNT_A_V2An (n = 1 to 9), ECCCNT_D_V2AnW (n = 1 to 9), and ECCCNT_D_V2AnR (n = 1 to 9))

PB0ECC

DTS_COMPC

Note 4. This PBG channel protects the registers related to the following modules:

MECCCAP_*

BECCCAP_* (except for BECCCAP_V2An (n = 1 to 9))

38.5.9.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offsets from the base addresses.

Table 38.398 Register Base Addresses

Base Address Name	Base Address	Bus Group
<PBG00_base>	FFC6 B000 _H	Peripheral Group 0
<PBG10_base>	FFC6 3000 _H	Peripheral Group 1
<PBG2L0_base>	FFDA 0800 _H	Peripheral Group 2L
<PBG2H0_base>	FFDE 1000 _H	Peripheral Group 2H
<PBG2H1_base>	FFDE 1100 _H	Peripheral Group 2H
<PBG30_base>	FFC7 2B00 _H	Peripheral Group 3
<PBG40_base>	FFC7 5300 _H	Peripheral Group 4
<PBG50_base>	FFC7 B000 _H	Peripheral Group 5
<PBG60_base>	FFC8 1000 _H	Peripheral Group 6
<PBG61_base>	FFC8 1100 _H	Peripheral Group 6
<PBG62_base>	FFC8 1200 _H	Peripheral Group 6
<PBG63_base>	FFC8 1300 _H	Peripheral Group 6
<PBG64_base>	FFC8 1400 _H	Peripheral Group 6
<PBG65_base>	FFC8 1500 _H	Peripheral Group 6
<PBG66_base>	FFC8 1600 _H	Peripheral Group 6
<PBG70_base>	FFF4 9000 _H	Peripheral Group 7
<PBG80_base>	FFF5 9000 _H	Peripheral Group 8
<PBG90_base>	FF0A 1300 _H	Peripheral Group 9

38.5.9.3 List of Registers

Table 38.399 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
PBGn (n = 00, 10, 2L0, 2H0, 2H1, 30, 40, 50, 60-66, 70, 80, 90)	Guard Error Clear Register	PBGERRCLR	<PBGn_base> + 00 _H	8, 16, 32	
	Guard Error Status Register	PBGERRSTAT	<PBGn_base> + 04 _H	8, 32	
	Guard Error Address Register	PBGERRADDR	<PBGn_base> + 08 _H	32	
	Guard Error Access Information Register	PBGERRTYPE	<PBGn_base> + 0C _H	16, 32	
	Key Code Protection Register	PBGKCPROT	<PBGn_base> + 10 _H	32	
	Channel Protection Control Register	PBGPROT0_m (m = 0-15)*1	<PBGn_base> + 80 _H + m × 08 _H	8, 16, 32	PBGKCPROT
	Channel SPID Setting Register	PBGPROT1_m (m = 0-15)*1	<PBGn_base> + 84 _H + m × 08 _H	8, 16, 32	PBGKCPROT

Note 1. "m" of PBGPROT0_m and PBGPROT1_m indicates the number of PBG channel. Each channel number is different from each other then please refer to **Section 38.5.9.1, Overview**.

38.5.9.4 PBGERRCLR —Guard Error Clear Register

This register is used to clear the PBGERRSTAT register. The PBGERRSTAT.OVF and/or PBGERRSTAT.ERR bits are cleared immediately after writing this register. Response data is always 0 when reading this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.400 PBGERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CLRO	Clears the PBGERRSTAT.OVF bit
0	CLRE	Clears the PBGERRSTAT.ERR bit

38.5.9.5 PBGERRSTAT — Guard Error Status Register

This register is used to notify users whether guard errors occurred. The ERR bit is set when a guard error is notified by an error detection module, etc. The OVF bit is set when a guard error is notified again when the ERR bit is already set. This register is not writable and is cleared when writing the PBGERRCLR register. See **Section 38.5.9.4, PBGERRCLR —Guard Error Clear Register.**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.401 PBGERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	OVF	Overflow status flag
0	ERR	Error status flag

38.5.9.6 PBGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when PBGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.402 PBGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

38.5.9.7 PBGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when PBGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]				—	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.403 PBGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

38.5.9.8 PBGKCPROT — Key Code Protection Register

This register is used to unlock/lock key protection of the P-bus guard register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.404 PBGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

38.5.9.9 PBGPROT0_m — Channel Protection Control Register

Specifies the P-Bus peripheral access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the settings other than SPID.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 38.405 PBGPROT0_m Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, PBGPROT1_m is used as a judgment condition. 1: During write, PBGPROT1_m is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, PBGPROT1_m is used as the judgment condition. 1: During read, PBGPROT1_m is not used as a judgment condition.

Note: An access by specific instructions (CLR1, NOT1, SET1 and CAXI) is protected by PBG if either reading or writing is disabled.

38.5.9.10 PBGPROT1_m — Channel SPID Setting Register

Specifies the P-Bus peripheral access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the SPID setting.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPID[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.406 PBGPROT1_m Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	<p>R/W enable setting based on SPID</p> <p>PBGPROT1_m is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits.</p> <p>For example, setting PBGPROT1_m to 0101_B enables access to areas with SPID = 0 and SPID = 2.</p> <p>0: Reading/writing the area with SPID = m depends on the RG and WG bit setting</p> <p>1: Enables reading/writing the area with SPID = m</p>

38.5.10 HBG

38.5.10.1 Overview

HBG can prevent read and write access against which a peripheral circuit on H-Bus should be protected. If HBG detects illegal access, guard error notification is signaled to ECM.

The following table lists the peripheral circuits to be protected and the corresponding HBG names.

Table 38.407 List of Peripheral Circuit Modules to be Protected

HBG	Module to be Protected
HBG91	RHSIF0
HBG93	FlexRay0
HBG95	Ethernet

38.5.10.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offsets from the base addresses.

Table 38.408 Register Base Addresses

Base Address Name	Base Address	Bus Group
<HBG91_base>	FF0D 0300 _H	Peripheral Group 9
<HBG93_base>	FF0D 0000 _H	Peripheral Group 9
<HBG95_base>	FF0D 0200 _H	Peripheral Group 9

38.5.10.3 List of Registers

Table 38.409 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
HBGn (n = 91, 93, 95)	Guard Error Clear Register	HBGERRCLR	<HBGn_base> + 00 _H	8, 16, 32	
	Guard Error Status Register	HBGERRSTAT	<HBGn_base> + 04 _H	8, 16, 32	
	Guard Error Address Register	HBGERRADDR	<HBGn_base> + 08 _H	32	
	Guard Error Access Information Register	HBGERRTYPE	<HBGn_base> + 0C _H	16, 32	
	Key Code Protection Register	HBGKCPROT	<HBGn_base> + 10 _H	32	
	Channel Protection Control Register	HBGPROT0	<HBGn_base> + 80 _H	8, 16, 32	HBGKCPROT
	Channel SPID Setting Register	HBGPROT1	<HBGn_base> + 84 _H	8, 16, 32	HBGKCPROT

38.5.10.4 HBGERRCLR — Guard Error Clear Register

This register is used to clear the HBGERRSTAT register. The HBGERRSTAT.OVF and/or HBGERRSTAT.ERR bits are cleared immediately after writing this register. Response data is always 0 when reading this register.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 38.410 HBGERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CLRO	Clears the HBGERRSTAT.OVF bit
0	CLRE	Clears the HBGERRSTAT.ERR bit

38.5.10.5 HBGERRSTAT — Guard Error Status Register

This register is used to notify users whether guard errors occurred. The ERR bit is set when a guard error is reported by an error detection module, etc. The OVF bit is set when a guard error is reported again when the ERR bit is already set. This register is not writable and is cleared when writing the HBGERRCLR register. See **Section 38.5.10.4, HBGERRCLR — Guard Error Clear Register.**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.411 HBGERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	OVF	Overflow status flag
0	ERR	Error status flag

38.5.10.6 HBGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when HBGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.412 HBGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

38.5.10.7 HBGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when HBGERRSTAT.ERR bit is not set.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]					—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.413 HBGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

38.5.10.8 HBGKCPROT — Key Code Protection Register

This register is used for protection against writing to the channel protection control registers due to program malfunction and the like.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 38.414 HBGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

38.5.10.9 HBGPROT0 — Channel Protection Control Register

Specifies the H-Bus peripheral access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the settings other than SPID.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 38.415 HBGPROT0 Register Contents

Bit Position	Bit Name	Function
31 to 9	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, HBGPROT1_n is used as a judgment condition. 1: During write, HBGPROT1_n is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, HBGPROT1_n is used as the judgment condition. 1: During read, HBGPROT1_n is not used as a judgment condition.

38.5.10.10 HBGPROT1 — Channel SPID Setting Register

Specifies the H-Bus peripheral access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the SPID setting.

Value after reset: 0000 0000_H

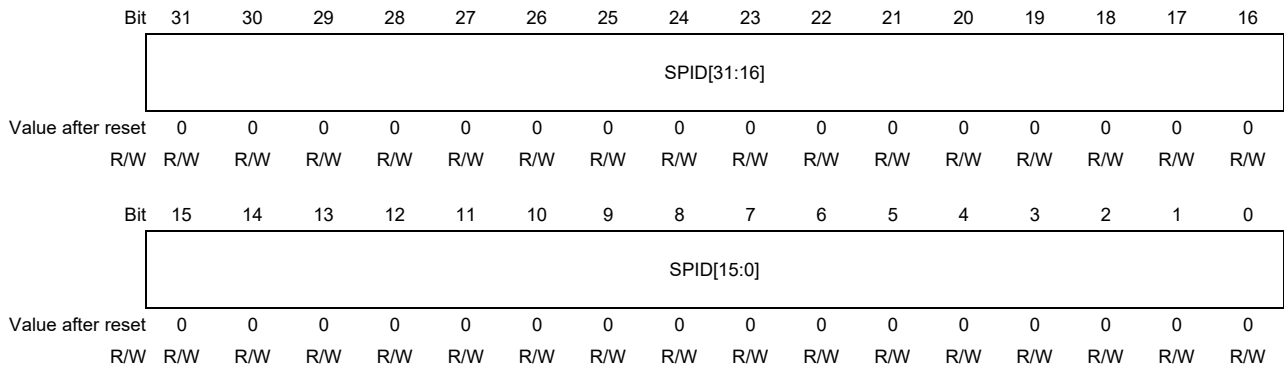


Table 38.416 HBGPROT1 Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	<p>R/W enable setting based on SPID</p> <p>HBGPROT1_n is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits.</p> <p>For example, setting HBGPROT1_n to 0101_B enables access to areas with SPID = 0 and SPID = 2.</p> <p>0: Reading/writing the area with SPID = m depends on the RG and WG bit setting</p> <p>1: Enables reading/writing the area with SPID = m</p>

38.6 BIST

38.6.1 Overview

This product incorporates Built-In-Self-Test (BIST) function to detect failures of the safety mechanism itself. The BIST consists of Logic BIST (LBIST) and Memory BIST (MBIST). LBIST is considered as the effective hardware safety mechanism to measure latent faults which can reduce software load for error injection tests. LBIST covers over 90% of the safety mechanisms for LFM. The BIST is executed during the reset sequence of System Reset 1 and System Reset 2 before the CPU starts operation. BIST execution results can be identified by the BIST result register (BSEQ0ST). In addition, by setting System Reset 2 before shutdown, BIST can be performed during the reset sequence in accordance with the setting of the BSEQ0CTL register. When running BIST, the scenario can be selected in accordance with the setting of the BSEQ0SEL register. For LBIST and MBIST, either a power-up BIST using System Reset 1 (set by using flash option byte 3) or a shutdown BIST using System Reset 2 (set by the BSEQ0SEL register) can be selected.

38.6.2 Registers

38.6.2.1 Register Protection

Write protected registers are protected from inadvertent write access due to erroneous program execution, etc. Write protected registers can be written by releasing the protection of RESKCPROT. For details of RESKCPROT, see **Section 10.2.16, RESKCPROT — Reset Controller Register Key Code Protection Register**.

38.6.2.2 List of Registers

Table 38.417 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
BIST (Peripheral Group6)	Logic BIST Reference Value Register 1	LBISTREF1	FF70 E100 _H	32	
	Logic BIST Reference Value Register 2	LBISTREF2	FF70 E104 _H	32	
	Memory BIST Reference Value Register 1	MBISTREF1	FF70 E108 _H	32	
	Logic BIST Signature Value Register 1	LBISTSIG1	FF70 E110 _H	32	
	Logic BIST Signature Value Register 2	LBISTSIG2	FF70 E114 _H	32	
	Memory BIST Signature Value Register 1	MBISTSIG1	FF70 E118 _H	32	
	Memory BIST1 FTAG Signature Value Register 0	MBIST1FTAG0	FF70 E120 _H	32	
	Memory BIST1 FTAG Signature Value Register 1	MBIST1FTAG1	FF70 E124 _H	32	
	Memory BIST1 FTAG Signature Value Register 2	MBIST1FTAG2	FF70 E128 _H	32	
	BIST Sequencer Status Register	BSEQ0ST	FF70 E160 _H	32	
	BIST Sequencer Inverted Status Register	BSEQ0STB	FF70 E164 _H	32	
	BIST Result Register	BISTST	FF70 E168 _H	32	
	BIST Scenario Select Register	BSEQ0SEL	FF70 E16C _H	32	
	SYSCTRL (Peripheral Group6)	BIST Skip Control Register	BSEQ0CTL	FF70 0400 _H	32

Note 1. See **Section 10.2.16, RESKCPROT — Reset Controller Register Key Code Protection Register.**

38.6.2.3 Reset of Registers

Register reset conditions are shown in **Table 38.418**.

Table 38.418 Reset sources

Register Name	Reset Source					
	Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
LBISTREF1	✓	✓	✓	—	—	—
LBISTREF2	✓	✓	✓	—	—	—
MBISTREF1	✓	✓	✓	—	—	—
LBISTSIG1	✓	✓	✓	—	—	—
LBISTSIG2	✓	✓	✓	—	—	—
MBISTSIG1	✓	✓	✓	—	—	—
MBIST1FTAG0	✓	✓	✓	—	—	—
MBIST1FTAG1	✓	✓	✓	—	—	—
MBIST1FTAG2	✓	✓	✓	—	—	—
BSEQ0ST	✓	✓	✓	—	—	—
BSEQ0STB	✓	✓	✓	—	—	—
BISTST	✓	✓	✓	—	—	—
BSEQ0SEL	✓	✓	—	—	—	—
BSEQ0CTL	✓	✓	—	—	—	—

38.6.2.4 LBISTREF1 — Logic BIST Reference Value Register 1

This register indicates the reference value of the Logic BIST. This register is automatically updated by the reference value after BIST execution.

Value after reset: BIST reference value *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	LBISTREF1[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LBISTREF1[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.419 LBISTREF1 Register Contents

Bit Position	Bit Name	Function
31 to 20	—	Reserved When reading, the value after reset is read. When writing, write the value after reset.
19 to 0	LBISTREF1[19:0]	Reference signature value (expected signature value)

Note 1. Value after reset depends on BIST execution condition. For details about BIST execution condition, see **Section 38.6.3.1, BIST Execution Condition.**

BIST is skipped: 000A 5A5A_H

BIST is executed: BIST reference value

38.6.2.5 LBISTREF2 — Logic BIST Reference Value Register 2

This register indicates the reference value of the Logic BIST. This register is automatically updated by the reference value after BIST execution.

Value after reset: BIST reference value *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	LBISTREF2[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LBISTREF2[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.420 LBISTREF2 Register Contents

Bit Position	Bit Name	Function
31 to 20	—	Reserved When reading, the value after reset is read. When writing, write the value after reset.
19 to 0	LBISTREF2[19:0]	Reference signature value (expected signature value)

Note 1. Value after reset depends on BIST execution condition. For details about BIST execution condition, see **Section 38.6.3.1, BIST Execution Condition.**

BIST is skipped: 0005 A5A5_H

BIST is executed: BIST reference value

38.6.2.6 MBISTREF1 — Memory BIST Reference Value Register 1

This register indicates the reference value of the Memory BIST. This register is automatically updated by the reference value after BIST execution.

Value after reset: BIST reference value *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	MBISTREF1[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBISTREF1[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.421 MBISTREF1 Register Contents

Bit Position	Bit Name	Function
31 to 20	—	Reserved When reading, the value after reset is read. When writing, write the value after reset.
19 to 0	MBISTREF1[19:0]	Reference signature value (expected signature value)

Note 1. Value after reset depends on BIST execution condition. For details about BIST execution condition, see **Section 38.6.3.1, BIST Execution Condition.**

BIST is skipped: 000A A55A_H

BIST is executed: BIST reference value

38.6.2.7 LBISTSIG1 — Logic BIST Signature Value Register 1

This register indicates the signature value of the Logic BIST. This register is automatically updated by the BIST result signature after BIST execution.

The user compares the reference signature of LBISTREF1 against the resulting signature LBISTSIG1. The Logic BIST is passed if these are equal.

Value after reset: BIST result *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	LBISTSIG1[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LBISTSIG1[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.422 LBISTSIG1 Register Contents

Bit Position	Bit Name	Function
31 to 20	—	Reserved When reading, the value after reset is read. When writing, write the value after reset.
19 to 0	LBISTSIG1[19:0]	Logic BIST1 result signature value

Note 1. Value after reset depends on BIST execution condition. For details about BIST execution condition, see **Section 38.6.3.1, BIST Execution Condition.**

BIST is skipped:	0005 A5A5 _H
BIST is executed without LBIST:	LBISTREF1
BIST is executed with LBIST:	BIST result

38.6.2.8 LBISTSIG2 — Logic BIST Signature Value Register 2

This register indicates the signature value of the Logic BIST. This register is automatically updated by the BIST result signature after BIST execution.

The user shall compare the reference signature of LBISTREF2 against the resulting signature LBISTSIG2. The Logic BIST is passed if these are equal.

Value after reset: BIST result *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	LBISTSIG2[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LBISTSIG2[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.423 LBISTSIG2 Register Contents

Bit Position	Bit Name	Function
31 to 20	—	Reserved When reading, the value after reset is read. When writing, write the value after reset.
19 to 0	LBISTSIG2[19:0]	Logic BIST2 result signature value

Note 1. Value after reset depends on BIST execution condition. For details about BIST execution condition, see **Section 38.6.3.1, BIST Execution Condition.**

BIST is skipped:	000A 5A5A _H
BIST is executed without LBIST:	LBISTREF2
BIST is executed with LBIST:	BIST result

38.6.2.9 MBISTSIG1 — Memory BIST Signature Value Register 1

This register indicates the signature value of the Memory BIST. This register is automatically updated by the BIST result signature after BIST execution.

The user shall compare the reference signature of MBISTREF1 against the resulting signature MBISTSIG1. The Memory BIST is passed if these are equal.

Value after reset: BIST result *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	MBISTSIG1[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBISTSIG1[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.424 MBISTSIG1 Register Contents

Bit Position	Bit Name	Function
31 to 20	—	Reserved When reading, the value after reset is read. When writing, write the value after reset.
19 to 0	MBISTSIG1[19:0]	Memory BIST1 result signature

Note 1. Value after reset depends on BIST execution condition. For details about BIST execution condition, see **Section 38.6.3.1, BIST Execution Condition.**

- BIST is skipped: 0005 5AA5_H
- BIST is executed without MBIST: MBISTREF1
- BIST is executed with MBIST: BIST result

38.6.2.10 MBIST1FTAG0 — Memory BIST1 FTAG Signature Value Register 0

This register indicates the Memory BIST status of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Value after reset:		BIST result *1																										
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	MBIST1FTAG0[31:16]											
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	MBIST1FTAG0[15:0]											
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Table 38.425 MBIST1FTAG0 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1FTAG0[31:0]	The state of self-diagnostic MBIST of Bridge No.n 0: MBIST status is PASS. 1: MBIST status is FAIL.

Note 1. Value after reset depends on BIST execution condition. For details about BIST execution condition, see **Section 38.6.3.1, BIST Execution Condition.**

BIST is skipped:	FFFF FFFF _H
BIST is executed without MBIST:	0000 0000 _H
BIST is executed with MBIST:	BIST result

Table 38.426 Mapping of MBIST1FTAG0 Register

Bit Position	Memory Group Assignment
31 to 13	Cluster0 RAM
12 to 10	Instruction Cache RAM(PE1)
9 to 7	Instruction Cache RAM(PE0)
6, 5	Local RAM(PE0)
4, 3	Local RAM(PE1)
2	Ethernet RAM
1 to 0	FlexRay RAM

38.6.2.11 MBIST1FTAG1 — Memory BIST1 FTAG Signature Value Register 1

This register indicates the Memory BIST status of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Value after reset:		BIST result *1																										
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	MBIST1FTAG1[31:16]											
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	MBIST1FTAG1[15:0]											
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.427 MBIST1FTAG1 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1FTAG1[31:0]	The state of self-diagnostic MBIST of Bridge No.n 0: MBIST status is PASS. 1: MBIST status is FAIL.

Note 1. Value after reset depends on BIST execution condition. For details about BIST execution condition, see **Section 38.6.3.1, BIST Execution Condition.**

BIST is skipped:	FFFF FFFF _H
BIST is executed without MBIST:	0000 0000 _H
BIST is executed with MBIST:	BIST result

Table 38.428 Mapping of MBIST1FTAG1 Register

Bit Position	Memory Group Assignment
31 to 27	ICU-M cache RAM
26 to 23	Reserved
22 to 15	Cluster1 RAM
14 to 3	Cluster0 RAM
2 to 1	Reserved
0	Cluster0 RAM

38.6.2.12 MBIST1FTAG2 — Memory BIST1 FTAG Signature Value Register 2

This register indicates the Memory BIST status of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Value after reset: BIST result *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1FTAG2[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1FTAG2[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.429 MBIST1FTAG2 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1FTAG2[31:0]	The state of self-diagnostic MBIST of Bridge No.n 0: MBIST status is PASS. 1: MBIST status is FAIL.

Note 1. Value after reset depends on BIST execution condition. For details about BIST execution condition, see **Section 38.6.3.1, BIST Execution Condition.**

BIST is skipped:	FFFF FFFF _H
BIST is executed without MBIST:	0000 0000 _H
BIST is executed with MBIST:	BIST result

Table 38.430 Mapping of MBIST1FTAG2 Register

Bit Position	Memory Group Assignment
31 to 28	Reserved
27 to 26	sDMAC1 RAM
25 to 24	sDMAC0 RAM
23	DTS RAM
22 to 21	RS-CANFD RAM
20 to 19	DFE1 RAM
18 to 17	DFE0 RAM
16 to 5	GTM RAM
4 to 1	ICU-M Local RAM
0	Reserved

38.6.2.13 BSEQ0ST — BIST Sequencer Status Register

This register indicates the state of the BIST sequencer. This register is automatically updated by the BIST result after BIST execution.

Value after reset: BIST result *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BISTEN D	CMPE R
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.431 BSEQ0ST Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When reading, the value after reset is read. When writing, write the value after reset.
1	BISTEND	0: The BIST sequence did not finish within the time limit. 1: The BIST sequence finished within the time limit.
0	CMPEERR	0: BIST normal end 1: BIST abnormal end

Note 1. Value after reset depends on BIST execution condition. For details about BIST execution condition, see **Section 38.6.3.1, BIST Execution Condition.**

BIST is skipped(except for Debug mode[TRST = 0]): 0000 0001_H

BIST is skipped/not executed(Debug mode[TRST = 1]): 0000 0002_H

BIST is executed: BIST result

38.6.2.14 BSEQ0STB — BIST Sequencer Inverted Status Register

This register indicates the inverted state of the BIST sequencer. This register is automatically updated by the BIST result after BIST execution.

Value after reset: BIST result *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BISTEN DB	CMPE RB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.432 BSEQ0STB Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When reading, the value after reset is read. When writing, write the value after reset.
1	BISTENDB	0: The BIST sequence finished within the time limit. 1: The BIST sequence did not finish within the time limit.
0	CMPERRB	0: BIST abnormal end 1: BIST normal end

Note 1. Value after reset depends on BIST execution condition. For details about BIST execution condition, see **Section 38.6.3.1, BIST Execution Condition.**

BIST is skipped(except for Debug mode[TRST = 0]): 0000 0002_H

BIST is skipped/not executed(Debug mode[TRST = 1]): 0000 0001_H

BIST is executed: BIST result

38.6.2.15 BISTST — BIST Result Register

This register indicates the result of the BIST. This register is automatically updated by the BIST result after BIST execution.

Value after reset: BIST result *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MBIST1ST	LBIST2ST	LBIST1ST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.433 BISTST Register Contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved When reading, the value after reset is read. When writing, write the value after reset.
2	MBIST1ST	Memory BIST result 0: MBIST1 passed successfully 1: MBIST1 has detected an error
1	LBIST2ST	Logic BIST (Reset toggle check) result 0: LBIST2 passed successfully 1: LBIST2 has detected an error
0	LBIST1ST	Logic BIST (Data toggle check) result 0: LBIST1 passed successfully 1: LBIST1 has detected an error

Note 1. Value after reset depends on BIST execution condition. For details about BIST execution condition, see **Section 38.6.3.1, BIST Execution Condition.**

BIST is skipped: 0000 000F_H

BIST is executed:

MBIST1ST value:

without MBIST: 0

with MBIST: BIST result

LBIST1ST, LBIST2ST value:

without LBIST: 0

with LBIST: BIST result

38.6.2.16 BSEQ0SEL — BIST Scenario Select Register

This register is used to select the execution scenario of BIST at the next System Reset 2 occurrence.

Value after reset: 0000 0030_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PARAM SEL	—	—	—	—	—	—	—	—	—	HWTESTSET [1:0]	—	—	HWLBISTSEL [1:0]	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Table 38.434 BSEQ0SEL Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved When reading, the value after reset is read. When writing, write the value after reset.
15	PARAMSEL	Selects BIST parameters (HWLBISTSEL[1:0], HWTESTSET[1:0]) of BIST which will be executed during next System Reset 2 triggered by software reset. 0: Parameters specified by Flash option byte*1. (default) 1: Parameters specified by BSEQ0SEL registers.
14 to 6	—	Reserved When reading, the value after reset is read. When writing, write the value after reset.
5 to 4	HWTESTSET[1:0]	BIST selection 00 _B : Prohibited 01 _B : LBIST ONLY 10 _B : MBIST ONLY 11 _B : LBIST AND MBIST For BIST, four operation settings can be selected by combining HWTESTSET[1:0] and HWLBISTSEL[1:0]. Refer to Table 38.435, BIST scenario use case.
3 to 2	—	Reserved When reading, the value after reset is read. When writing, write the value after reset.
1, 0	HWLBISTSEL[1:0]	BIST scenario selection 00 _B : BIST scenario 1 01 _B : BIST scenario 2 10 _B : BIST scenario 3 11 _B : BIST scenario 4 For BIST, four operation settings can be selected by combining HWTESTSET[1:0] and HWLBISTSEL[1:0]. Refer to Table 38.435, BIST scenario use case.

Note 1. For details, see **Section 41, Flash Memory.**

Table 38.435 BIST scenario use case

HwLBISTSEL [1:0]* ¹	HwTESTSET [1:0]* ¹	Function	Run time* ²	Target ASIL* ³
00	10	MBIST Execution	5 ms (MBIST 5ms)	—
01	11	LBIST and MBIST Execution	10 ms (LBIST 5 ms, MBIST 5 ms)	ASIL B
10	01	LBIST Execution	15 ms (LBIST 15 ms)	ASIL D
11	11	LBIST and MBIST Execution	20 ms (LBIST 15 ms, MBIST 5 ms)	ASIL D

Note 1. Settings other than above are prohibited.

Note 2. Run time is calculated by the typical frequency of the Internal OSC. For details, see *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

Note 3. ISO26262 compliance for LFM by LBIST.

38.6.2.17 BSEQ0CTL — BIST Skip Control Register

This register is used to control the BIST execution at the next System Reset 2 occurrence.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HWBI STSEL	HWBIS TEXE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.436 BSEQ0CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When reading, the value after reset is read. When writing, write the value after reset.
1	HWBISTSEL	Selects flash option byte value or register value for the BIST execution setting. 0: BIST is executed according to Flash Option Byte* ¹ value. 1: BIST is executed according to HWBISTEXE value.
0	HWBISTEXE	This bit selects if BIST is skipped or not at the next system reset 2. 0: BIST is skipped at the next system reset 2 1: BIST is executed at the next system reset 2

Note 1. For details, see **Section 41, Flash Memory**.

38.6.3 Functions

38.6.3.1 BIST Execution Condition

The BIST is executed during reset sequence of System Reset 1 and System Reset 2. The following tables explain the conditions of BIST execution.

Table 38.437 BIST Execution during System Reset 1

Conditions			BIST Execution during Reset Sequence
Reset Category	TRST pin	HWBIST (Flash option*1)	
System Reset 1	High	1	BIST not executed. *2
		0	BIST is skipped. *3
	Low	1	BIST is executed. *4
		0	BIST is skipped. *3

Note 1. For details, See **Section 41, Flash Memory**.

Note 2. Reset is released after the expiration of the selected BIST scenario time.

Note 3. Reset is released without waiting for the selected BIST scenario time.

Note 4. Reset is released after BIST has finished. BIST is executed with preset parameters. For details, See **Section 41, Flash Memory**.

Table 38.438 BIST Execution during System Reset 2 (ECM, SWDTA, Software Reset)

Reset Category	TRST pin	HWBISTSEL (Register)	
System Reset 2	High	0	HWBIST (Flash option*1) Value 1: BIST is not executed.*2 0: BIST is skipped.*3
		1	BSEQ0CTL register HWBISTEXE Value 1: BIST is not executed.*2 0: BIST is skipped.*3
	Low	0	HWBIST (Flash option*1) Value 1: BIST is executed.*4 0: BIST is skipped.*3
		1	BSEQ0CTL register HWBISTEXE Value 1: BIST is executed. 0: BIST is skipped.*3

Note 1. For details, See **Section 41, Flash Memory**.

Note 2. Reset is released after the expiration of the selected BIST scenario time.

Note 3. Reset is released without waiting for the selected BIST scenario time.

Note 4. Reset is released after BIST has finished. BIST is executed with preset parameters. For details, See **Section 41, Flash Memory**.

38.6.3.2 BIST Result Confirmation

After the BIST execution, read and compare the following register values to confirm whether the BIST execution completed normally without failure.

- BSEQ0ST = 0000 0002_H
- BSEQ0STB = 0000 0001_H
- LBISTSIG1 = LBISTREF1
- LBISTSIG2 = LBISTREF2
- MBISTSIG1 = MBISTREF1
- BISTST = 0000 0000_H

38.6.4 Operation

By setting System Reset 2 before shutdown, BIST can be performed during the reset sequence in accordance with the setting of the BSEQ0CTL register. When running BIST, the scenario can be selected in accordance with the setting of the BSEQ0SEL register. For LBIST and MBIST, either a power-up BIST using System Reset 1 (set by using flash option byte 3) or a shutdown BIST using System Reset 2 (set by the BSEQ0SEL register) can be selected.

Table 38.439 BIST Scenario Select Condition

RESET	Source	BIST Scenario Select
System Reset 1	External Reset, Standby Reset, VMON reset	Flash option byte*1
System Reset 2	ECM reset, SWDTA reset, Software reset	BSEQ0SEL register (PARAMSEL = 1)

Note 1. For details, See **Section 41, Flash Memory**.

38.7 MISG

38.7.1 Overview

This MCU incorporates multi-input signature generators (MISG) for self-diagnosis by the CPUs.

The table below shows the overview of the MISG specifications.

Table 38.440 Specification Overview

Item	Description
Signature generation units	This MCU incorporates the signature generation unit for each CPU core. MISG extracts necessary information from CPU trace and inputs it to the signature generation units in write monitoring mode.
Generating polynomials	Two types of polynomials are available to generate a 32-bit signature. <ul style="list-style-type: none"> MISR1: $G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ MISR2: $G(x) = x^{32} + x^{22} + x^2 + x + 1$
Signature generation	Signature generation can be enabled or disabled. <ul style="list-style-type: none"> Signature generation in MISR1 can be enabled or disabled. Signature generation in MISR2 can be enabled or disabled. <p>The following two conditions can be selected as conditions for signature generation:</p> <ul style="list-style-type: none"> Writing to the register A signature is generated by writing to the MISR calculation data register (MISRCDR_*). Monitoring write access A signature is generated if writing to the address area specified for monitoring occurs when writing by using the CPU that is being monitored. The address area is specified by the monitoring area base address register and the monitoring area address mask register.
Automatic signature comparison	Two signature generation units are selected for comparison of signatures. Each signature generation unit has a data counter, and comparison proceeds when the values of the data counters in the MISGs selected as the target for comparison match. Moreover, MISG stops the generation of signature and the count of the data counters when the values of the data counters in the MISGs match the values of MISR compare data counter register (MISRCMPDCNT), and comparison also proceeds after both the counters that are the target for comparison match stop. The data counter counts the number of write accesses to the MISRCDR_* register or the address area being monitored.
Error notification	When signatures are compared and do not match, the ECM is notified of an error. Enabling or disabling of error notification to the ECM can be selected. An interrupt request is not issued directly to the INTC.

Note 1. "*" in MISRCDR_*, MISR1_*, and MISR2_* indicates PEm. (m = 0, 1)

38.7.2 Block Diagram

38.7.2.1 MISG

Figure 38.2 shows a block diagram of the MISG. The MISG consists of the signature generation unit and the signature comparison unit.

Using write monitoring mode as the signature generation condition (**Section 38.7.3.1(2), Write Monitoring Mode**) is only possible between the corresponding signature generation unit and the CPU. Therefore, each signature generation unit is given a name corresponding to the CPU number (MISG_PE0, MISG_PE1). MISG_PEm can monitor write access by CPU_m. ($m = 0, 1$)

When register write mode is the signature generation condition, there is no correspondence between a CPU and the signature generation unit. Any CPU can generate a signature in any signature generation unit.

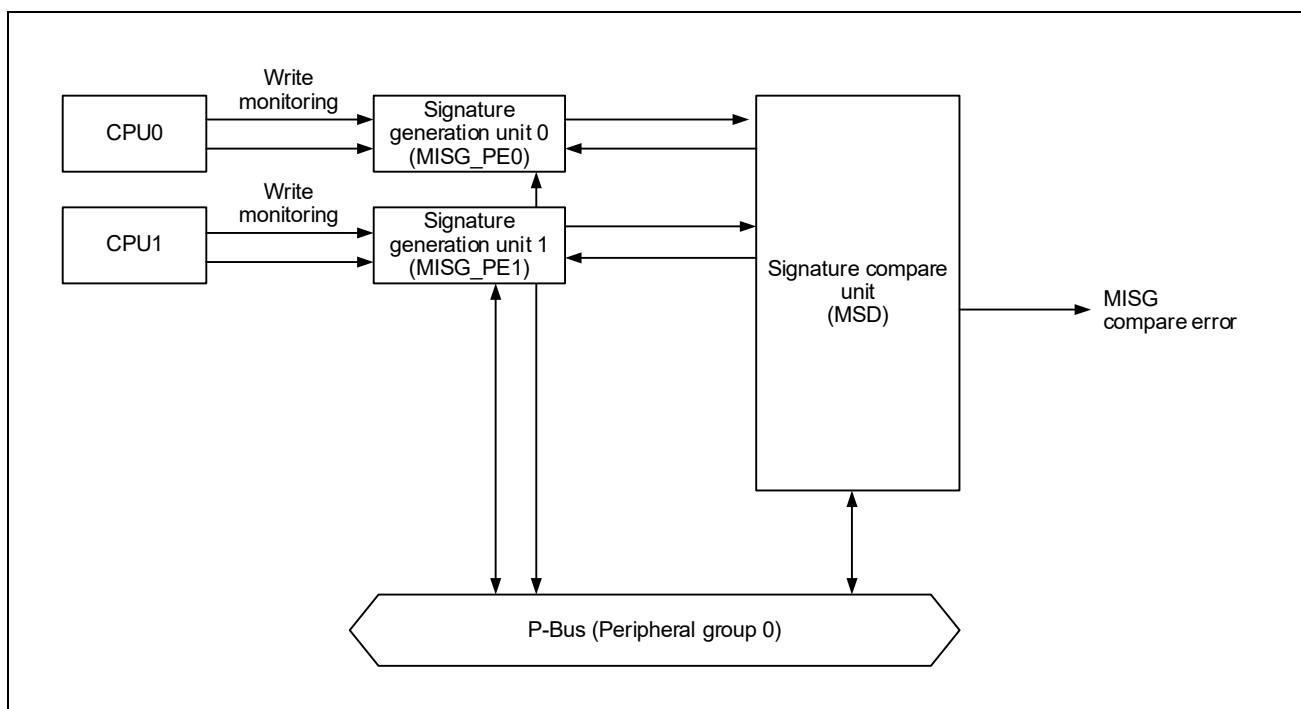


Figure 38.2 MISG Block Diagram (Multi Core)

38.7.2.2 Signature Generation

Figure 38.3 shows the flow of data in signature generation. MISR1 and MISR2 consist of two 32-bit signature generation units (MISR1_*, MISR2_*).

MISR2_* can generate a signature from 32-bits of data being written by the CPU being monitored, or from data being written to MISRCDR_*. Signature generation units stop the signature generation based on a signal from signature compare unit triggered by automatic signature comparison.

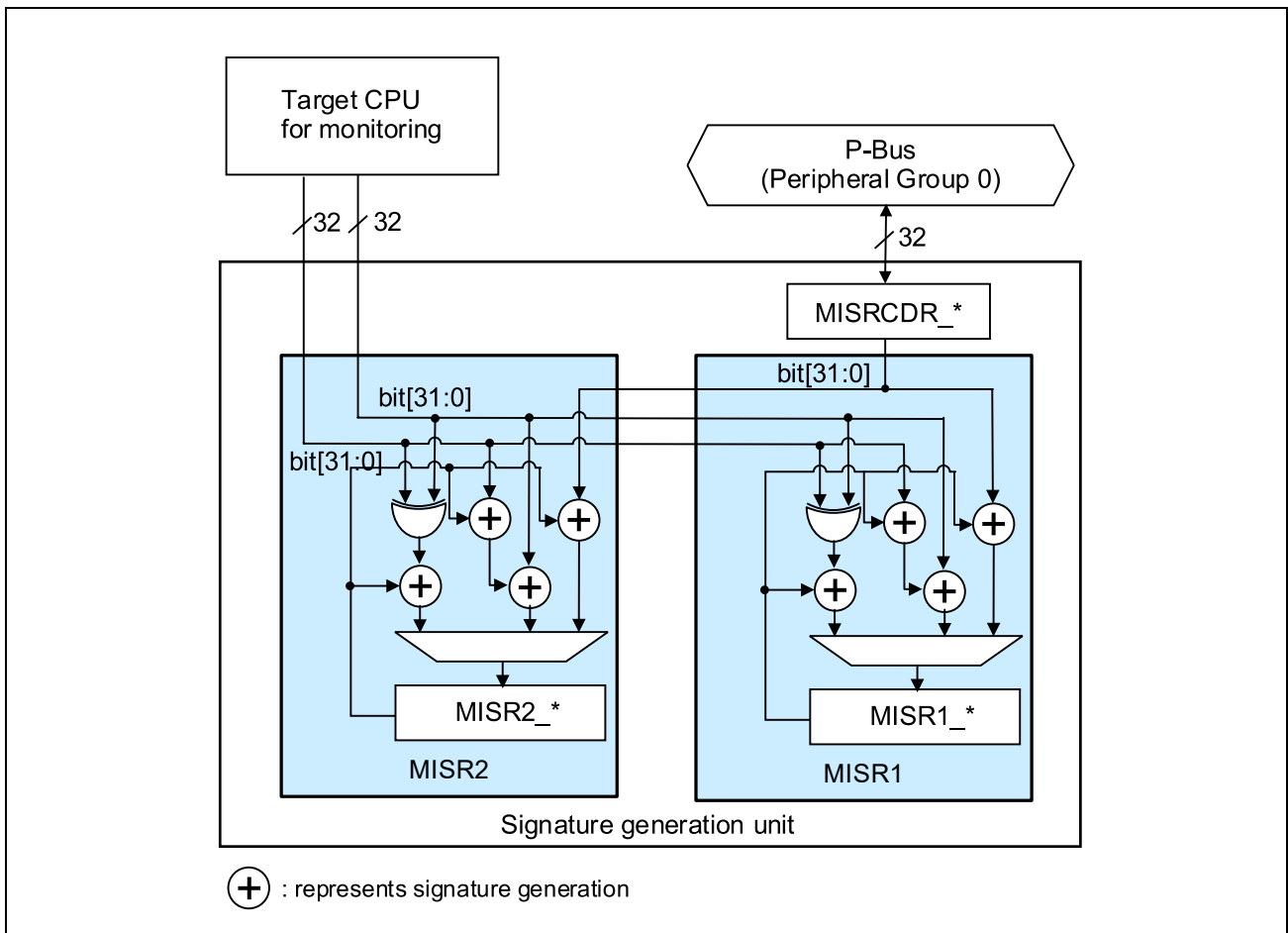


Figure 38.3 Signature Generation Units

Figure 38.4 shows the block diagram of signature generation in MISR1_* and the polynomials.

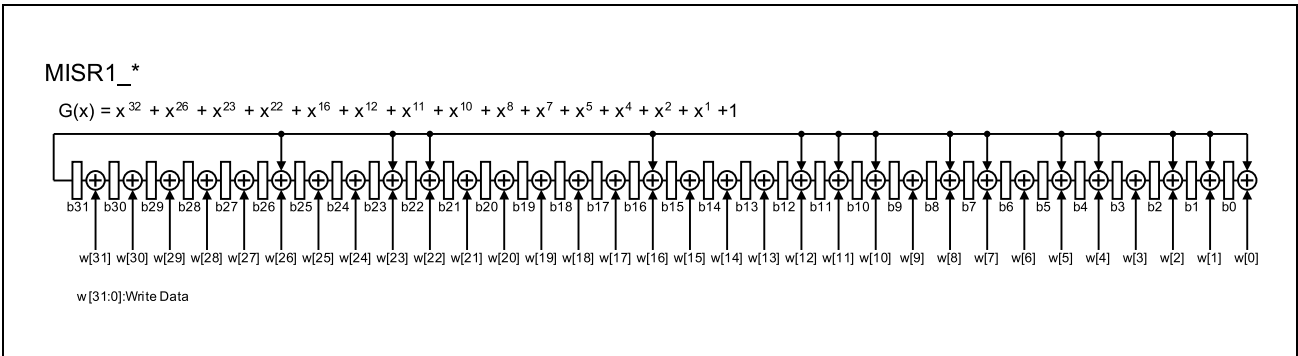


Figure 38.4 Block Diagram of Signature Generation in MISR1_*

Figure 38.5 shows the block diagram of signature generation in MISR2_* and the polynomials.

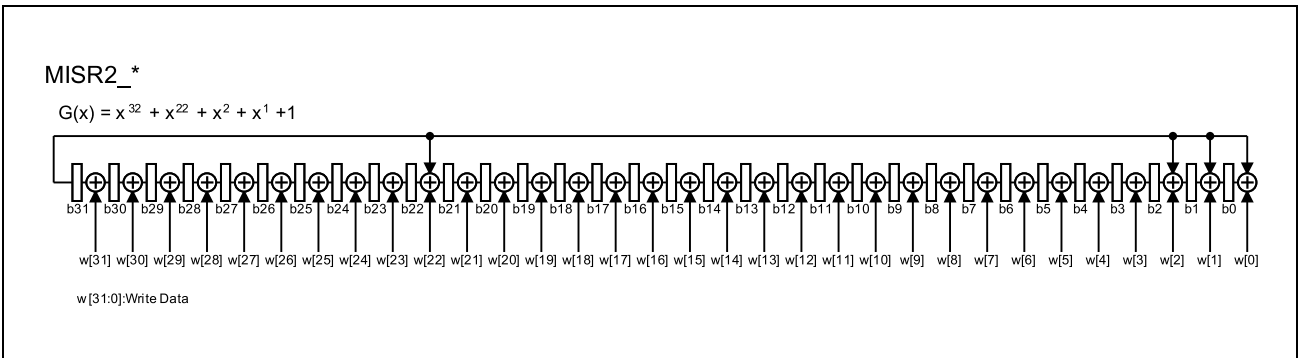


Figure 38.5 Block Diagram of Signature Generation in MISR2_*

38.7.2.3 Signature Compare Unit

The signature compare unit consists of the signature generation selector part, counter compare part and signature compare part. The signature generation selector part can select the input signals to the signature compare part and the counter compare part from all implemented signature generation units. Furthermore, the signature generation selector part inputs the stop signal for signature generation and the control signal for count up generated by the counter compare part.

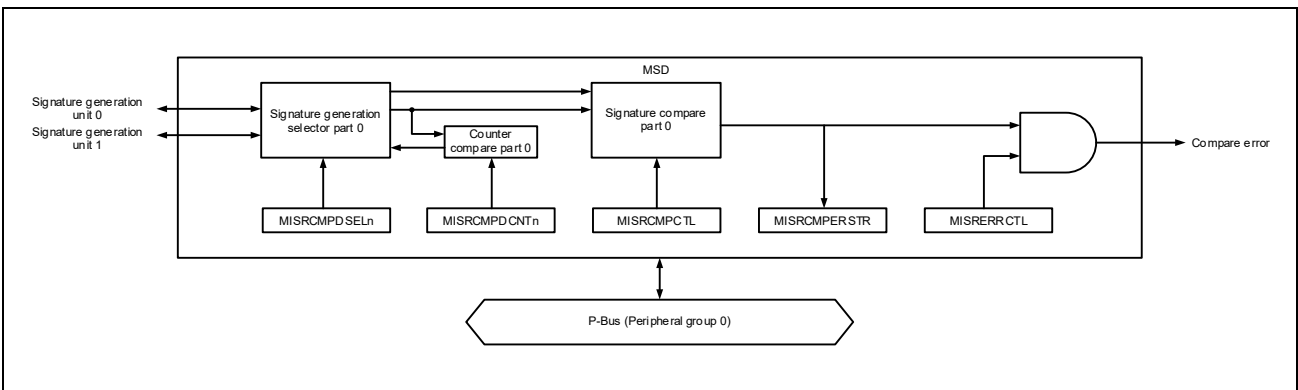


Figure 38.6 Block Diagram of MSD

38.7.3 Functional Specifications

38.7.3.1 Conditions for Signature Generation

The conditions for signature generation in MISR1 and MISR2 can be selected by the setting the MISR control register (MISRCR_*).

Note 1. “*” in MISRCR_* indicates PEm. (m = 0, 1)

Signature generation conditions for MISRi (i = 1, 2)

Table 38.441 Signature Generation Conditions for MISRi

MISRCR. MISRiEN	MISRCR. MISEiCND	Signature Generation Conditions
0	—	MISRi does not generate a signature.
1	0	Register write mode Writing to the MISRCDR_* register leads to MISRi generating a signature.
1	1	Write monitoring mode MISRi generates a signature after the corresponding CPU writes to an address specified for monitoring.

(1) Register Write Mode

Writing to the MISR calculation data register (MISRCDR_*) while MISR1 is in register write mode leads to the generation of a 32-bit signature in MISR1 from the value held in the multi-input signature register 1 (MISR1_*) and the data written to MISRCDR_*, and the result is retained in MISR1_*. Similarly, a 32-bit signature is generated in MISR2 from the value held in the multi-input signature register 2 (MISR2_*) and the data written to MISRCDR_*, and the result is retained in MISR2_*.

Writing to MISRCDR_* can proceed in 8-, 16-, or 32-bit units, and bits to which a value is not written are treated as 0. For example, writing to the 16 lower-order bits of MISRCDR_* leads to the generation of a signature with the 16 higher-order bits of write data treated as 0. Similarly, writing to the 16 higher-order bits of MISRCDR_* leads to the generation of a signature with the 16 lower-order bits of write data treated as 0.

MISR1 and MISR2 do not identify the bus master that writes to the MISRCDR_*. Writing by any bus master, whether a CPU, DMAC, or debugging master, produces a signature.

Note 1. “*” in MISRCDR_*, MISR1_*, and MISR2_* indicates PEm. (m = 0, 1)

(2) Write Monitoring Mode

When the CPU writes to an address area specified for monitoring while MISR1 is in write-monitoring mode, MISR1 generates compressed 32-bit signatures from the value in MISR1_* and 32-bit data being written by the CPU, and generates signatures from the 32-bit compressed data and the value that is again held in MISR1_*. The generated signatures are stored in MISR1_*. Similarly, when the CPU writes to an address area specified for monitoring while MISR2 is in write-monitoring mode, MISR2 generates compressed 32-bit signatures from the value in MISR2_* and 32-bit data being written by the CPU, and generates signatures from the 32-bit compressed data and the value that is again held in MISR2_*. The generated signatures are stored in MISR2_*.

The address area specified for monitoring for signature generation is set by the MISR monitoring base address register (MISRBASEADR) and the MISR monitoring address mask register (MISRADRMASK). If the CPU write address is within the address area specified for monitoring, a signature is generated.

Write monitoring mode monitors write access by the corresponding CPU in 8-, 16-, 32-, or 64-bit units. When writing proceeds in 8-, or 16-bit units, remaining bits to which a value is not written are treated as 0, and 32-bit data is always input to MISR1 or MISR2. Write data is allocated to the lower-order side regardless of the destination address. For example, when 16 bits are written to an address of the form $4N + 2$, the 16 bits are allocated to the lower-order side and a signature is generated with the 16 higher-order bits treated as 0.

In case of 64-bit write access, the data is split into two 32-bit data, that is, lower word and upper word, and then they are used to generate a signature in lower to upper order.

The CPU write access destinations that can be monitored by MISR1 and MISR2 are as follows:

Local RAM, Cluster RAM, Peripheral module connected to System bus, I-Bus or P-Bus

Furthermore, CPU instructions which MISR1 and MISR2 can monitor are:

ST.B, ST.H, ST.W, ST.DW

SST.B, SST_H, SST_W

PREPARE, PUSHP (Following restriction exists.)

The restrictions described below apply to the given store operations.

- (1) Instructions that cause the CPU to write data in response to a slave request: Store operations such as CLR1, NOT1, SET1, CAXI, and STC.W are not subject to monitoring.
- (2) Instructions for saving units of data larger than 64 bits on the stack (PREPARE and PUSHSP): signature values between the masters may not match when the instruction is suspended because of an interrupt. Therefore, the area to be used with the instructions must not be subject to monitoring or steps must be taken to make sure that the operation is not suspended due to an interrupt.
- (3) STV.DW, STV.QW, STV.W, STVZ.H4 (FXU store operation) and CACHE, PREFI are not subject to monitoring.
- (4) Register bank function: signature values can be uncertain when context switch occurs. Therefore, the area to be used by the evacuations needs to be out of monitoring.

Write monitoring mode can only be implemented between the specified signature generation unit and the corresponding CPU.

In this product, the following monitoring operations are possible:

- Monitoring of CPU_m write access by signature generation unit m (m = 0, 1)

Note 1. "*" in MISR1_* and MISR2_* indicates PEm. (m = 0, 1)

38.7.3.2 Automatic Signature Comparison

The signature compare part is enabled by MISR1CMPENn bit and MISR2CMPENn bit of the comparator control register (MISRCMPCTL) and it compares signatures selected from two signature generation units by the MISR compare data select register (MISRCMPDSEL) corresponding to the signature compare part. Each signature generation unit contains the data counter and the comparison is done when the count values of the data counters contained in the signature generation unit which are selected as the target for comparison match each other. Moreover, MISG stops the generation of signature and the count of the data counters when the values of the data counters in the MISGs match the values of MISR compare data counter register (MISRCMPDCNT) and comparison is also done after both counters that are the target for comparison match stop.

38.7.3.3 Data Counter

The MISR data counter register (MISRDCNT_*) counts the number of write accesses to MISRCDR_* or to the address range specified by BASEADR and ADRMSK. When the CNTSTA bit = 1 and the CNTTRG bit = 0 in the data counter control register (MISRDCNTCTL), MISRDCNT_* counts the number of write accesses to MISRCDR_*. When the CNTSTA bit = 1 and the CNTTRG bit = 1 in MISRDCNTCTL, MISRDCNT_* counts the number of write accesses by the corresponding CPU to the address range specified by MISRBASEADR and MISRADRMSK.

MISRDCNT_* can count the number of times either MISR1 or MISR2 or both MISR1 and MISR2 generate signatures by setting the trigger for counting up by MISRDCNT_* as the signature generation condition in MISR1 or MISR2. Note, however, that if the signature generation conditions for MISR1 and MISR2 and the trigger for counting up by the data counter are not consistent, the value of MISRDCNT_* and the signature generation count will not match.

The MISR data counter register (MISRDCNT_*) stops the count base on a signal from the signature compare part triggered by the automatic signature compare function.

CAUTIONS

1. When the MISR1EN and MISR2EN bits in the MISRCR register are both 0, counting by MISRDCNTCTL is not incremented even if writing to MISRCDR and the address range specified by MISRBASEADR and MISRADRMSK proceeds.
2. A double word access is treated as two single word accesses and MISRDCNT_* register counts two in write monitoring mode. Signature generation unit (MISG) uses only lower word and stops just in the stop count specified by MISRCMPDCNT register in case a double word access occurs when MISRDCNT_* = MISRCMPDCNT – 1.

Note 1. “*” in MISRDCNT_* and MISRCDR_* indicates PE_m. (m = 0, 1)

38.7.3.4 Error Notification

When the CMPERREN bit in the error notification control register (MISRERRCTL) is set to 1, the ECM is notified of a MISR compare error (MISR_ERR) when signatures are compared and do not match. At the same time, the error flag in the compare error status register is set. Moreover, MISR compare error is being notified during compare mismatch of signatures. See **Section 38.7.5.3, Recommended Clear Flow of Error Flag** regarding error flag clear.

There is no generation of interrupt request to INTC.

38.7.4 Register Specifications

38.7.4.1 Register Map

The table below lists the registers of the signature generation units.

Registers with symbols ending in “_PEm” are those of signature generation unit m (MISG_PEm). (m = 0, 1)

Note that the endings of the register symbols (“_PEm”) are omitted if signature generation unit m does not require identification.

$$\text{MISG_PEm_base} = \text{FFFB } 0000_{\text{H}} + m \times 40_{\text{H}} \quad (m = 0, 1)$$

$$\text{MSD_base} = \text{FFFB } 0200_{\text{H}}$$

Table 38.442 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
MISG_PEm (m = 0, 1) (Peripheral Group0)	Multi Input Signature Register 1	MISR1	<MISG_PEm_base> + 00 _H	32	
	Multi Input Signature Register 2	MISR2	<MISG_PEm_base> + 08 _H	32	
	Calculation Data Register	MISRCDR	<MISG_PEm_base> + 10 _H	8, 16, 32	
	Control Register	MISRCR	<MISG_PEm_base> + 18 _H	8	
	Monitoring Base Address Register	MISRBASEADR	<MISG_PEm_base> + 1C _H	32	
	Monitoring Address Mask Register	MISRADRMSK	<MISG_PEm_base> + 20 _H	32	
	Data Counter Control Register	MISRDCNTCTL	<MISG_PEm_base> + 24 _H	8	
	Data Counter Register	MISRDCNT	<MISG_PEm_base> + 28 _H	16	
MSD (Peripheral Group0)	Comparator Control Register	MISRCMPCTL	<MSD_base> + 00 _H	8	
	Compare Error Status Register	MISRCMPERSTR	<MSD_base> + 04 _H	8	
	Compare Error Status Clear Register	MISRCMPERRSTC	<MSD_base> + 08 _H	8	
	Error Notification Control Register	MISRERRCTL	<MSD_base> + 0C _H	8	
	Compare Data Select Register	MISRCMPDSEL	<MSD_base> + 10 _H	8	
	Compare Data Counter Register	MISRCMPDCNT	<MSD_base> + 20 _H	16	

38.7.4.2 MISR1 — Multi Input Signature Register 1

The Multi-input signature register 1 is a 32-bit readable/writable register.

Once the signature generation condition is met, this register generates a new signature each time the condition is met and retains the generated value. For the conditions for signature generation, see **Section 38.7.3.1, Conditions for Signature Generation**.

Signatures are generated by using the following polynomial:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MISR1[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MISR1[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.443 MISR1 Register Contents

Bit Position	Bit Name	Function
31 to 0	MISR1[31:0]	Multi Input signature register 1

38.7.4.3 MISR2 — Multi Input Signature Register 2

Multi-input signature register 2 is a 32-bit readable/writable register.

Once the signature generation condition is met, this register generates a new signature each time the condition is met and retains the generated value. For the conditions for signature generation, see **Section 38.7.3.1, Conditions for Signature Generation**.

Signatures are generated by using the following polynomial:

$$G(x) = x^{32} + x^{22} + x^2 + x + 1$$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MISR2[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MISR2[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.444 MISR2 Register Contents

Bit Position	Bit Name	Function
31 to 0	MISR2[31:0]	Multi Input signature register 2

38.7.4.4 MISRCDR — Calculation Data Register

The MISR calculation data register is a 32-bit write-only register.

When signature generation in register write mode is selected, a signature is generated in MISR1 or MISR2 by writing to this register. Data written to this register is the data input to the multi-input signature register 1 (MISR1_*) or the multi-input signature register 2 (MISR2_*). For the conditions for signature generation, see **Section 38.7.3.1, Conditions for Signature Generation**.

This register can be written in 8-, 16-, or 32-bit units. When writing proceeds in 8- or 16-bit units, remaining bits to which a value is not written are treated as 0, and 32-bit data is always input to MISR1_* or MISR2_*.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MISRCDR[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MISRCDR[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 38.445 MISRCDR Register Contents

Bit Position	Bit Name	Function
31 to 0	MISRCDR[31:0]	Calculation data to MISR1 or MISR2 New signatures are generated by the CPU writing to this register. The signature generating condition is as follows: MISRCR.MISR1EN = 1 and MISRCR.MISR1CND = 0 or MISRCR.MISR2EN = 1 and MISRCR.MISR2CND = 0

38.7.4.5 MISRCR — Control Register

MISR control register is an 8-bit readable/writable register.

The MISR1EN and MISR2EN bits are used to enable signature generation in MISR1 and MISR2. When the MISR1EN or MISR2EN bit is 1, MISR1 or MISR2 generates a signature and retains the generated value. When the MISR1EN or MISR2EN bit is 0, MISR1 or MISR2 does not generate signatures and the values of these registers are not updated.

When the MISR1EN or MISR2EN bit is 1, the MISR1CND or MISR2CND bit selects the signal generation condition for MISR1 or MISR2. Setting the MISR1CND or MISR2CND bit to 0 selects signature generation by MISR1 or MISR2 in register write mode, so writing to MISRCDR_* leads to signature generation. Setting the MISR1CND or MISR2CND bit to 1 selects signature generation by MISR1 or MISR2 in write monitoring mode. Writing by the CPU to the address range specified by the MISRBASEADR and MISRADRMSK registers leads to signature generation. For the conditions for signature generation, see **Section 38.7.3.1, Conditions for Signature Generation**.

MISR1EN bit and MISR2EN bit are cleared by the signal from the signature compare part of the signature compare unit in automatic signature comparison. The clear timing is the next edge of the bus clock after the counter value set by the MISR data counter register (MISRDCNT_*) and the MISR compare data counter register (MISRCMPDCNT). After the clear, it is possible to set again the MISR1EN or MISR2EN in a state that the MISRDCNT_* matches MISRCMPDCNT however neither the signature nor the counter value is updated.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	MISR2CND	MISR1CND	MISR2EN	MISR1EN
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.446 MISRCR Register Contents

Bit Position	Bit Name	Function
7 to 4	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
3	MISR2CND	Controls the signature generation condition when MISR2EN is "1" 0: Writing to MISRCDR 1: Writing to the address range which is specified by BASEADR and ADRMSK
2	MISR1CND	Controls the signature generation condition when MISR1EN is "1" 0: Writing to MISRCDR 1: Writing to the address range specified by BASEADR and ADRMSK
1	MISR2EN	MISR2 enable bit 0: Do not generate the MISR2 signature 1: Generate the MISR2 signature by writing to MISRCDR and by writing to the address range specified BASEADR and ADRMSK
0	MISR1EN	MISR1 enable bit 0: Do not generate the MISR1 signature 1: Generate the MISR1 signature by writing to MISRCDR and by writing to the address range specified BASEADR and ADRMSK

38.7.4.6 MISRBASEADR — Monitoring Base Address Register

MISRBASEADR specifies the CPU write access area to be monitored by MISG when signature generation in write monitoring mode is selected. In combination with the setting of the monitoring area mask address register, this register specifies the address range of the monitoring area. For the conditions for signature generation, see **Section 38.7.3.1, Conditions for Signature Generation**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BASEADR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BASEADR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.447 MISRBASEADR Register Contents

Bit Position	Bit Name	Function
31 to 0	BASEADR[31:0]	Base address of the monitoring address range

For the mechanism to judge access to the monitoring area, see **Section 38.7.4.7, MISRADRMSK — Monitoring Address Mask Register**.

38.7.4.7 MISRADRMSK — Monitoring Address Mask Register

MISRADRMSK specifies the CPU write access area to be monitored by MISG when signature generation in write monitoring mode is selected. In combination with the setting of the monitoring area base address register, this register specifies the address range of the monitoring area. For the conditions for signature generation, see **Section 38.7.3.1, Conditions for Signature Generation**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADRMSK[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADRMSK[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.448 MISRADRMSK Register Contents

Bit Position	Bit Name	Function
31 to 0	ADRMSK[31:0]	Mask address of the monitoring address range

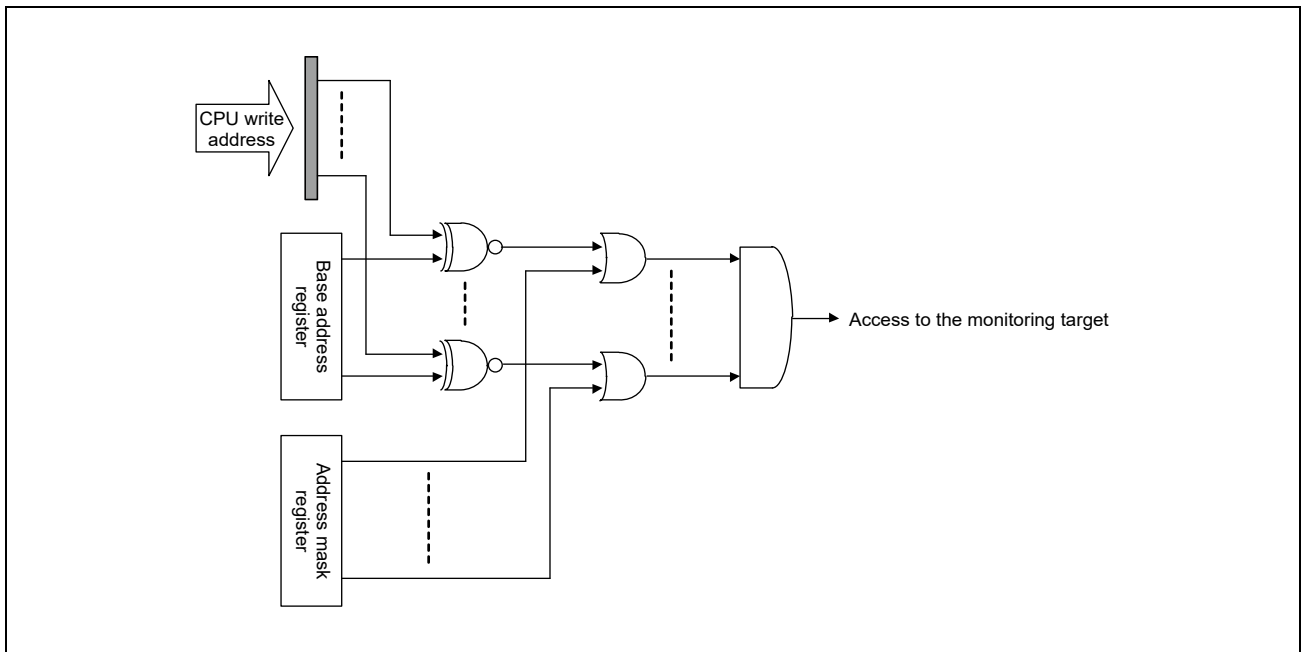


Figure 38.7 How to Detect Access to the Monitoring Area

38.7.4.8 MISRDCNTCTL — Data Counter Control Register

The MISR data counter control register is an 8-bit readable/writable register.

This register controls operation of the MISR data counter register. If the event selected by the CNTTRG bit occurs while the CNTSTA bit is 1, the data counter is incremented. For operation of the data counter, see **Section 38.7.3.3, Data Counter**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CNTTRG	CNTSTA
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 38.449 MISRDCNTCTL Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	CNTTRG	Selects the trigger to count up: 0: Writing to the MISRCRDR 1: Writing to the address range specified by BASEADR and the ADRMSK
0	CNTSTA	Enables the counter when CNTSTA = 1 and MISR1EN = 1 or MISR2EN = 1 0: Stop the data counter 1: Enable the data counter: the count up trigger is selected by CNTTRG

38.7.4.9 MISRDCNT — Data Counter Register

The data counter is a 16-bit readable/writable register.

If the data counter values of two signature generation units selected for comparison match, automatic comparison of the signatures proceeds.

When the CNTTRG bit in the data counter control register is 0, the data counter is incremented by write access to the MISR calculation data register. When the CNTTRG bit in the data control register is 1, the data counter is incremented by write access by the corresponding CPU to the address area specified by the MISRBASEADR and MISRADRMSK registers. For operation of the data counter, see **Section 38.7.3.3, Data Counter**.

Value after reset: 0000_H

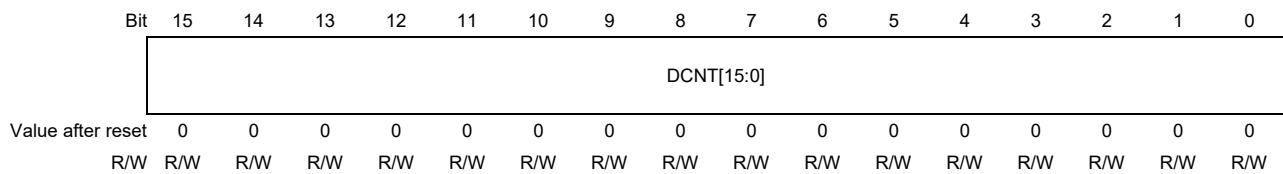


Table 38.450 MISRDCNT Register Contents

Bit Position	Bit Name	Function
15 to 0	DCNT[15:0]	The data counter register

38.7.4.10 MISRCMPCTL — Comparator Control Register

MISRCMPCTL is an 8-bit readable/writable register.

This register controls the comparator that compares signatures generated in the signature generation units. For automatic comparison of signatures, see **Section 38.7.3.2, Automatic Signature Comparison**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	MISR2 CMPEN0	—	—	—	MISR1 CMPEN0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R	R	R/W

Table 38.451 MISRCMPCTL Register Contents

Bit Position	Bit Name	Function
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	MISR2CMPEN0	Controls comparator for MISR2 signatures specified by MISRCMPDSEL register 0: Do not compare 1: Compare
3 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	MISR1CMPEN0	Controls comparator for MISR1 signatures specified by MISRCMPDSEL register 0: Do not compare 1: Compare

38.7.4.11 MISRCMPERSTR — Compare Error Status Register

The compare error status register is an 8-bit read-only register.

If a mismatch occurs in signature comparison enabled by the comparator control register, the corresponding error flag is set.

The error flag is cleared by writing 1 to the corresponding clear bit in the compare error status clear register. The flag is also cleared by a reset.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	MISR2ERR0	—	—	—	MISR1ERR0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 38.452 MISRCMPERSTR Register Contents

Bit Position	Bit Name	Function
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	MISR2ERR0	Indicates compare errors occurred in MISR2 signatures specified by MISRCMPDSEL register 0: Signatures matched 1: Signatures did not match
3 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	MISR1ERR0	Indicates compare errors occurred in MISR1 signatures specified by MISRCMPDSEL register 0: Signatures matched 1: Signatures did not match

38.7.4.12 MISRCMPERRSTC — Compare Error Status Clear Register

The compare error status clear register is an 8-bit write-only register.

When an error flag in the compare error status register is 1, writing 1 to corresponding clear bit clears the error flag. Read the MISR compare error status register and write 1 to the clear bit for the flag that is 1.

See **Section 38.7.5.3, Recommended Clear Flow of Error Flag** regarding error flag clear.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	MISR2CLR0	—	—	—	MISR1CLR0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	W	R	R	R	W

Table 38.453 MISRCMPERRSTC Register Contents

Bit Position	Bit Name	Function
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	MISR2CLR0	MISR2 signature compare error clear bit 0 1: Clear the MISR2ERR0 bit of MISRCMPERSTR register
3 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	MISR1CLR0	MISR1 signature compare error clear bit 0 1: Clear the MISR1ERR0 bit of MISRCMPERSTR register

38.7.4.13 MISRERRCTL — Error Notification Control Register

The error notification control register is an 8-bit readable/writable register.

This register enables or disables error notification when signatures are compared by the automatic signature comparison and do not match. For the automatic signature comparison and error notification, see **Section 38.7.3.2, Automatic Signature Comparison** and **Section 38.7.3.4, Error Notification**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CMPPERREN
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 38.454 MISRERRCTL Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	CMPPERREN	Comparison error notice permission bit 0: Do not report the error when the signatures do not match 1: Report the error when the signatures do not match

38.7.4.14 MISRCMPDSEL — Compare Data Select Register

The compare data select register is an 8-bit readable/writable register.

This register selects the input data of signatures to be compared by the automatic signature comparison. For the automatic signature comparison and error notification, see **Section 38.7.3.2, Automatic Signature Comparison** and **Section 38.7.3.4, Error Notification**.

Value after reset: 03_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	MISGPESEL[1:0]	
Value after reset	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R/W	R/W

Table 38.455 MISRCMPDSEL Register Contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1, 0	MISGPESEL[1:0]	Specify the MISG modules whose signatures are to be compared 11 _B : Select MISG modules for PE1 and PE0 Other: Setting prohibited

38.7.4.15 MISRCMPDCNT — Compare Data Counter Register

The compare data counter register is a 16-bit readable/writable register.

This register sets the count value that stops the data counter of the signature generation unit in automatic signature comparison. For the automatic signature comparison and error notification, see **Section 38.7.3.2, Automatic Signature Comparison** and **Section 38.7.3.4, Error Notification**.

Value after reset: FFFF_H

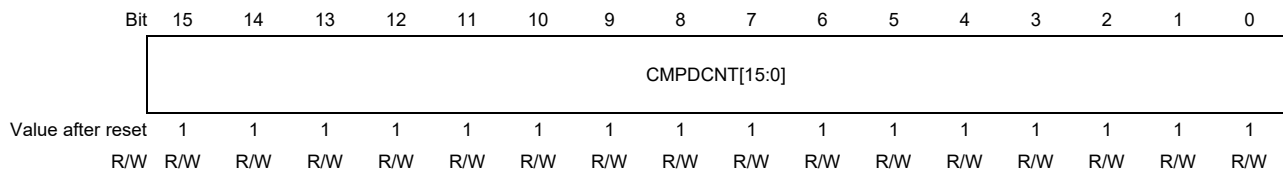


Table 38.456 MISRCMPDCNT Register Contents

Bit Position	Bit Name	Function
15 to 0	CMPDCNT[15:0]	Specify the counter value to stop generating signatures

38.7.5 Usage

38.7.5.1 Usage Example 1

MISG facilitates diagnosis of the self-diagnostic program. By using the MISG to compress intermediate transitions of the self-diagnostic program, the saving and comparison of all intermediate results becomes unnecessary. Results of self diagnosis including intermediate transitions can be judged by comparing the results of compression by the MISG with the expected results after the program ends.

This has the effect of reducing requirements for memory capacity and time for processing comparisons (substitution by CRCs is also possible).

Example of settings (when PEm is running the self-diagnostic program)

The descriptions of MISG registers below applies to the registers of signature generation unit 1 (MISG_PEm), which is for PEm.

- (1) Initialize the multi-input signature register 1 (MISR1) and multi-input signature register 2 (MISR2).
- (2) Set the MISR1CND or MISR2CND bit in the MISR control register (MISRCR) to 1 to select signature generation by MISR1 or MISR2 in write monitoring mode.
- (3) Use the MISR monitoring base address register (MISRBASEADR) and MISR monitoring address mask register (MISRADRMSK) to set the address area for monitoring.
- (4) Set the MISR1EN or MISR2EN bit in the MISR control register to 1 to enable signature generation in MISR1 or MISR2.
- (5) Read the MISR control register (MISRCR) and execute “syncp” instruction to confirm MISR enabled.
- (6) Run the self-diagnostic program on PEm.
- (7) Upon completion of execution of the self-diagnostic program, MISR1 or MISR2 data is compared with the expected value in flash memory.

38.7.5.2 Usage Example 2

Use multiple processors to run the same processing (including the self-diagnostic program) to confirm the correctness of results. Comparing results from different hardware raises reliability.

Example of settings (when PE0 and PE1 are running the same task)

The descriptions of MISG registers below applies to the MISG registers of PE0 and PE1.

- (1) Initialize the multi-input signature register 1 (MISR1), multi-input signature register 2 (MISR2), and data counter register (MISRDCNT).
- (2) Set the MISR1CND or MISR2CND bit in the MISR control register (MISRCR) to 0 to select signature generation by MISR1 or MISR2 in register write mode.
- (3) Set the signature generation unit of PE0 and PE1 as the input of the signature compare part by using the MISR compare data select register (MISRCMPDSEL).
- (4) Set the value of data counter for comparing the signatures by using the MISR compare data counter register (MISRCMPDCNT).
- (5) Set MISR1CMPEN0 or MISR2CMPEN0 in the MISR comparator control register (MISRCMPCTL) to 1 to enable signature comparison by the comparator.

- (6) Set the CMPERREN bit in the MISR error notification control register to 1 to enable error notification to the ECM.
- (7) Set the CNTTRG bit in the MISR data counter control register (MISRDCNTCTL) to 0 to set writing to MISRCDR_* as the trigger for counting up by the data counter. Set the CNTSTA bit to 1 to enable operation of the data counter.
- (8) Set the MISR1EN or MISR2EN bit in the MISR control register to 1 to enable signature generation in MISR1 or MISR2.
- (9) Read the MISR control registers (MISRCR in MISG_PE0 and MISG_PE1) and execute “syncp” instruction to confirm MISR enabled.
- (10) Run the self-diagnostic program on all CPUs.
- (11) The self-diagnostic program stores intermediate results while the program is running in MISRCDR_* of the signature generation units. The signature comparison part stops the signature generation and the counting of data counter when it matches the value of the MISR compare data counter register (MISRCMPDCNT). Signature comparison is then performed when the data counters of both PE0 and PE1 reach the value of the MISR compare data counter register (MISRCMPDCNT).
- (12) Check the comparison status register to see if there were any errors in comparison. See **Section 38.7.5.3, Recommended Clear Flow of Error Flag** regarding error flag clear.

38.7.5.3 Recommended Clear Flow of Error Flag

If the MISR compare error status clear register (MISRCMPERRSTC) clears the MISR compare error status register (MISRCMPERSTR) with signature mismatch and signature comparison enabled, the flag of MISR compare error is set again. Following shows the recommended clear flow of the error flag.

- (1) Set the MISRiCMPENn bit in the MISR comparator control register (MISRCMPCTL) to 0 to disable the comparison.
- (2) Clear the MISR compare error status register (MISRCMPERSTR) by the MISR compare error status clear register (MISRCMPERRSTC).
- (3) Check if the MISR compare error status register (MISRCMPERSTR) is cleared.
- (4) Initialize the multi-input signature register 1 (MISR1), multi-input signature register 2 (MISR2), and data counter register (MISRDCNT).

38.8 ECM

The ECM monitors various failure detection states in this microcontroller, and defines the operation to be carried out upon failure detection. For the details of the ECM, see **Section 39, Error Control Module (ECM)**.

38.9 Voltage Monitor

The Voltage Monitor detects over and under voltage of the voltage domains, E0VCC, VCC and VDD. For the details of the Voltage Monitor, see **Section 12, Power Supply Voltage Monitor**.

38.10 Clock Monitor

The Clock Monitor detects abnormal frequencies of internal clocks related to the safety of the device. For the details of the Clock Monitor, see **Section 14, Clock Monitor**.

38.11 Data CRC Function

The Data CRC Function generates CRC to verify the data streams protected by CRC. For the details of the Data CRC Function, see **Section 40, Data CRC Function B (DCRB)**.

Section 39 Error Control Module (ECM)

39.1 Features

39.1.1 Number of Units and Channels

This microcontroller has the following number of ECM unit.

Table 39.1 Number of Units

Product Name	RH850/E2x-FCC1	
	373 Pins	292 Pins
Number of Units	1	1
Name	ECM	ECM

Table 39.2 Number of Units

Product Name	RH850/E2M	
	373 Pins	292 Pins
Number of Units	1	1
Name	ECM	ECM

Table 39.3 Index

Index	Meaning
m	Throughout this section, the individual ECM Master and ECM Checker are identified by the index "m" (m = M, C).

39.1.2 Register Base Address

ECM base addresses are listed in the following table.

ECM register addresses are given as offsets from the base addresses in general.

Table 39.4 Register Base Address

Base Address Name	Base Address	Bus Group
<ECMM_base>	FFCB 0000 _H	Peripheral Group 6
<ECMC_base>	FFCB 1000 _H	Peripheral Group 6
<ECM_base>	FFCB 2000 _H	Peripheral Group 6

39.1.3 Clock Supply

Clock supply by and to ECM is listed in the following table.

Table 39.5 Clock Supply

Unit Name	Unit Clock Name	Clock Supply Name
ECM	PCLK	CLK_LSB
	cntclk* ¹	CLK_WDTICUM

Note 1. cntclk is used for delay timer and clear mask timer logics

39.1.4 Interrupt Requests

ECM interrupt requests are listed in the following table. The interrupt request signal is driven to the high level with a pulse width of one PCLK cycle when the error source status flag of an error source for which interrupt generation is enabled is set.

Table 39.6 Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Description	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
INTECMMI	INTECMMI	ECM maskable interrupt (EI level)	EIINT8	—	—
INTECMDCLSMI	INTECMDCLSMI	DCLS error interrupt (EI level)	EIINT9	—	—
FEINT0	INTECMNMI	ECM FE level interrupt (FE level)	FEINT0	—	—

39.1.5 Reset Sources

ECMM, ECMC and ECM reset sources are listed in the following table. ECMM, ECMC and ECM are initialized by these reset sources.

Table 39.7 Reset Sources

Unit Name	Register Name	Reset Condition							
		Power Up Reset	System Reset1			System Reset2	Application Reset	Module Reset	JTAG Reset
			Standby Reset	External Reset	VMON Reset				
ECM Master Registers									
ECMM	ECM master error set trigger register	✓	✓	✓	✓	✓	✓	—	—
ECMM	ECM master error clear trigger register	✓	✓	✓	✓	✓	✓	—	—
ECMM	ECM master error source status register n ^{*1}	✓	✓	(—)	X	—	—	—	—
ECM Checker Registers									
ECMC	ECM checker error set trigger register	✓	✓	✓	✓	✓	✓	—	—
ECMC	ECM checker error clear trigger register	✓	✓	✓	✓	✓	✓	—	—
ECMC	ECM checker error source status register n ^{*1}	✓	✓	(—)	X	—	—	—	—
ECM Common Registers									
ECM	ECM error pulse configuration register	✓	✓	✓	✓	✓	✓	—	—
ECM	ECM maskable interrupt configuration register n ^{*1}	✓	✓	✓	✓	✓	✓	—	—
ECM	ECM FE level interrupt configuration register n ^{*1}	✓	✓	✓	✓	✓	✓	—	—
ECM	ECM internal reset configuration register n ^{*1}	✓	✓	✓	✓	✓	✓	—	—
ECM	ECM error mask register n ^{*1}	✓	✓	✓	✓	✓	✓	—	—
ECM	ECM error source status clear register n ^{*1}	✓	✓	✓	✓	✓	✓	—	—
ECM	ECM key-code protection register	✓	✓	✓	✓	✓	✓	—	—
ECM Common Registers									
ECM	ECM pseudo error trigger register n ^{*1}	✓	✓	✓	✓	✓	✓	—	—
ECM	ECM delay timer control register	✓	✓	✓	✓	✓	✓	—	—
ECM	ECM delay timer register	✓	✓	✓	✓	✓	✓	—	—
ECM	ECM delay timer compare register	✓	✓	✓	✓	✓	✓	—	—
ECM	ECM maskable interrupt delay timer configuration register n ^{*1}	✓	✓	✓	✓	✓	✓	—	—
ECM	ECM FE level interrupt delay timer configuration register n ^{*1}	✓	✓	✓	✓	✓	✓	—	—
ECM	ECM error output clear invalidation configuration register	✓	✓	✓	✓	✓	—	—	—
ECM	ECM pseudo error mask register	✓	✓	✓	✓	✓	✓	—	—

Note: ✓: Register is initialized by reset, —: Register is not initialized by reset, X: Undefined value

(—): When the terminal RESET is asserted during reset sequence including BIST execution by external / internal reset, it will become X.

Note 1. n = 0 to 9

39.1.6 External Input and Output Signals

External Input/output signals of ECM are listed below.

Table 39.8 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
<u>ERROROUT_M</u>	Error output master signal	<u>ERROROUT_M</u>
<u>ERROROUT_C</u>	Error output checker signal	<u>ERROROUT_C</u>
<u>ERRORIN</u>	Error input signal	<u>ERRORIN</u>

39.2 Overview

39.2.1 Specification Overview

ECM (Error Control Module) collects error signals coming from different error sources and monitoring circuits. It also outputs error signals from the error pins ($\overline{\text{ERROROUT_M}}$, $\overline{\text{ERROROUT_C}}$) and generates interrupts and Error Control Module Reset signals. **Table 39.9** shows the specification overview of ECM.

Table 39.9 Specification Overview

Item	Description
Safety processing	<p>ECM can handle the following processing in response to error signal inputs from individual modules.</p> <ul style="list-style-type: none"> • Error flag set • EI level interrupt generation EI level interrupt generation can be controlled (enabled/disabled) for individual errors. • DCLS error interrupt generation DCLS error interrupt generation (EI level) can be controlled (enabled/disabled) for individual errors. Count the DCLS error and error message based on the DCLS error happen times are reported. 2bit status register for counting the DCLS error, the max count is 3. • FE level interrupt generation FE level interrupt generation can be controlled (enabled/disabled) for individual errors. • Internal reset generation System reset 2 or Application reset generation can be controlled (enabled/disabled) for individual errors. • Error pin output Pin output mask can be controlled (enabled/disabled) for individual errors. Output can be toggled in response to a timer input or made at a fixed level.
Error status	<p>ECM incorporates error source status registers, which can be used to confirm the error status from the error flag.</p> <p>The error flags are only cleared by a power up reset or Standby Reset (System reset 1). In case of External reset (System reset1), System reset2, Application reset, Module reset and JTAG reset, the error flags are kept and the reset generation source can be confirmed by reading the status register after reset.</p>
Debug, self-diagnosis	<ul style="list-style-type: none"> • Pseudo errors can be generated for debug and self-diagnosis. The operation during injection of pseudo errors is identical to that for the occurrence of real errors. All configurations for the masking of the error pin output, interrupt, or Error Control Module Reset apply in the same way. • ECM incorporates a loop-back function of the error pin output that is used to diagnose the path to the error output pin. The status of the error output pin is reflected to an internal register and can be confirmed by reading the register.
Timeout function	<p>ECM incorporates a function that generates an error signal output or Error Control Module Reset when the count value of the delay timer matches with the delay timer compare register because the delay timer was not stopped during the interrupt processing after being started simultaneously with the occurrence of an interrupt request.</p>
Port safe state	<p>$\overline{\text{ERROROUTZ}}$ connects to port safe state and ECM can control the state of general purpose I/O to a condition according to user configuration. Configuration state is Hi-Z, Low and High output. For details of function, see Section 2, Pin Function.</p>
Register protection	<p>A write-protection with a key code is implemented to protect registers from illegal write access. For details of function, see Section 39.3.11, ECMKCPROT — ECM Key Code Protection Register.</p>
$\overline{\text{ERROROUT}}$ clear masking	<p>ECM incorporates a function that can mask software clearance of $\overline{\text{ERROROUT}}$ until the counter which is started from error occurrence reaches the value specified in the Error Output Clear Invalidation Configuration register. If another error occurs during time counting, then the time count is reset and restarted from 0.</p>
Others	<p>ECM is duplexed. ECM incorporates the error output pin.</p>

39.2.2 Block Diagram

ECM is redundantly implemented using ECM Master and ECM Checker. See **Figure 39.1** Connection of two ECM and **Figure 39.2** Connection between ECM Master, ECM Checker and peripherals.

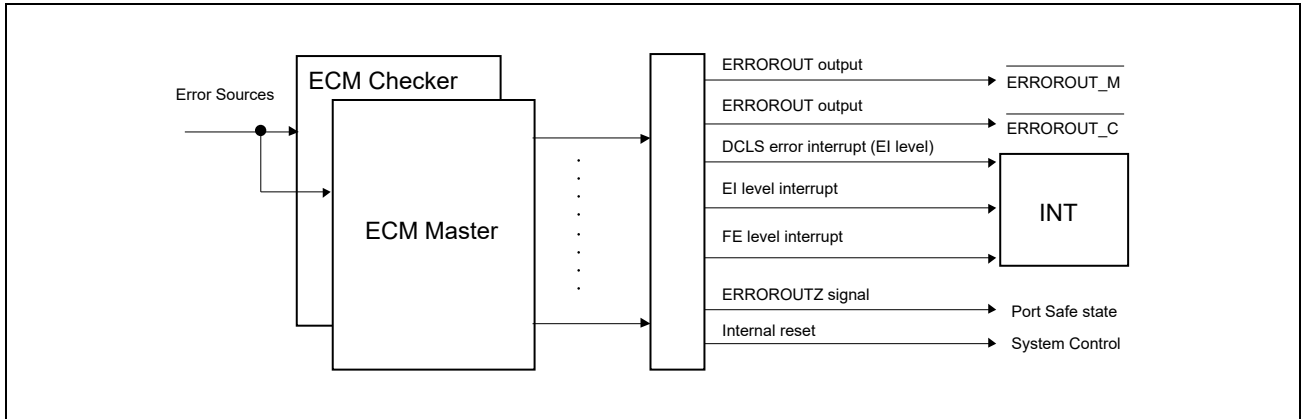


Figure 39.1 Connection of ECM

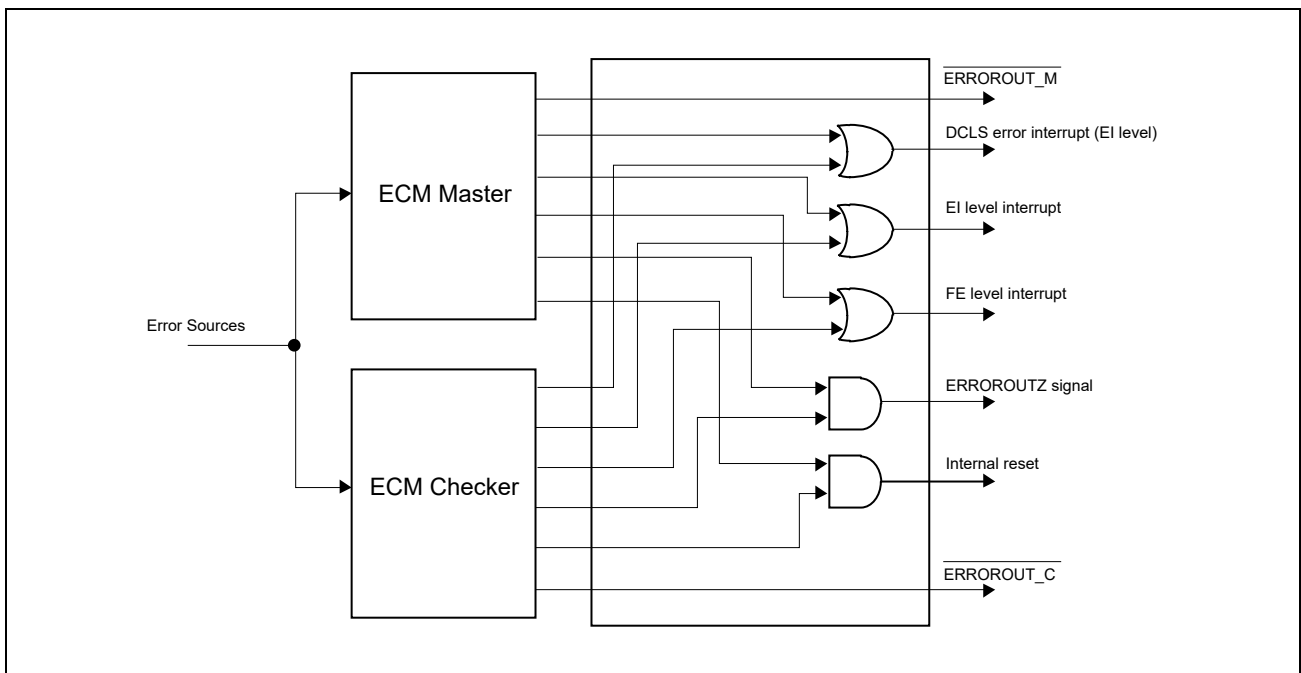


Figure 39.2 Connection between ECM Master, ECM Checker and Peripherals

CAUTION

Pay attention to the difference in output voltage between the $\overline{\text{ERROROUT_M}}$ and $\overline{\text{ERROROUT_C}}$ pins because different power supply systems are used for those pins.

$\overline{\text{ERROROUT_M}}$ pin: SYSVCC (5 V / 3.3 V)

$\overline{\text{ERROROUT_C}}$ pin: E0VCC (5 V) for P01_7, P15_7, P22_5 and P32_3; E1VCC (5 V / 3.3 V) for P11_2; E2VCC (5 V / 3.3 V) for P12_8

39.2.3 Error Input

Table 39.10 shows the error inputs to ECM of E2x-FCC1.

Table 39.11 shows the error inputs to ECM of E2M.

“—”: not covered, “✓”: covered, gray hatching: its number is reserve bit.

Table 39.10 List of Error Inputs of E2x-FCC1 (1/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State
0	Dual Core Lock-step	DCLS compare error (PE0)	✓	✓	✓	✓	✓	✓	✓	✓
1		DCLS compare error (PE1)	✓	✓	✓	✓	✓	✓	✓	✓
2		Reserved	—	—	—	—	—	—	—	—
3		Reserved	—	—	—	—	—	—	—	—
4		Reserved	—	—	—	—	—	—	—	—
5		Reserved	—	—	—	—	—	—	—	—
6		Reserved	—	—	—	—	—	—	—	—
7		Reserved	—	—	—	—	—	—	—	—
8	Watchdog timer	Watchdog timer ch0 error (PE0)	✓	✓	✓	✓*1	✓	✓	—	✓
9		Watchdog timer ch1 error (PE1)	✓	✓	✓	✓	✓	✓	—	✓
10		Reserved	—	—	—	—	—	—	—	—
11		Reserved	—	—	—	—	—	—	—	—
12		Reserved	—	—	—	—	—	—	—	—
13		Reserved	—	—	—	—	—	—	—	—
14		Reserved	—	—	—	—	—	—	—	—
15		Reserved	—	—	—	—	—	—	—	—
16	Local RAM (own core)	Local RAM (PE0) • Address feedback compare error	✓	✓	✓	✓	✓	✓	—	✓
17		Local RAM (PE1) • Address feedback compare error	✓	✓	✓	✓	✓	✓	—	✓
18		Reserved	—	—	—	—	—	—	—	—
19		Reserved	—	—	—	—	—	—	—	—
20		Reserved	—	—	—	—	—	—	—	—
21		Reserved	—	—	—	—	—	—	—	—
22		Reserved	—	—	—	—	—	—	—	—
23		Reserved	—	—	—	—	—	—	—	—
24		Local RAM ECC (PE0) • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓
25		Local RAM ECC (PE1) • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓
26		Reserved	—	—	—	—	—	—	—	—
27		Reserved	—	—	—	—	—	—	—	—
28		Reserved	—	—	—	—	—	—	—	—
29		Reserved	—	—	—	—	—	—	—	—
30		Reserved	—	—	—	—	—	—	—	—

Table 39.10 List of Error Inputs of E2x-FCC1 (2/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State
31	Local RAM (own core)	Reserved	—	—	—	—	—	—	—	—
32		Local RAM ECC (PE0) • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓
33		Local RAM ECC (PE1) • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓
34		Reserved	—	—	—	—	—	—	—	—
35		Reserved	—	—	—	—	—	—	—	—
36		Reserved	—	—	—	—	—	—	—	—
37		Reserved	—	—	—	—	—	—	—	—
38		Reserved	—	—	—	—	—	—	—	—
39		Reserved	—	—	—	—	—	—	—	—
40		Local RAM ECC (PE0) • Error address overflow	✓	✓	✓	✓	✓	✓	—	✓
41		Local RAM ECC (PE1) • Error address overflow	✓	✓	✓	✓	✓	✓	—	✓
42		Reserved	—	—	—	—	—	—	—	—
43		Reserved	—	—	—	—	—	—	—	—
44		Reserved	—	—	—	—	—	—	—	—
45		Reserved	—	—	—	—	—	—	—	—
46		Reserved	—	—	—	—	—	—	—	—
47		Reserved	—	—	—	—	—	—	—	—
48	Instruction Cache RAM	Instruction Cache RAM (PE0) • Address feedback compare error	✓	✓	✓	✓	✓	✓	—	✓
49		Instruction Cache RAM (PE1) • Address feedback compare error	✓	✓	✓	✓	✓	✓	—	✓
50		Reserved	—	—	—	—	—	—	—	—
51		Reserved	—	—	—	—	—	—	—	—
52		Reserved	—	—	—	—	—	—	—	—
53		Reserved	—	—	—	—	—	—	—	—
54		Reserved	—	—	—	—	—	—	—	—
55		Reserved	—	—	—	—	—	—	—	—
56		Instruction Cache RAM EDC (PE0) • EDC 2bit error • EDC 1bit error	✓	✓	✓	✓	✓	✓	—	✓
57		Instruction Cache RAM EDC (PE1) • EDC 2bit error • EDC 1bit error	✓	✓	✓	✓	✓	✓	—	✓
58		Reserved	—	—	—	—	—	—	—	—
59		Reserved	—	—	—	—	—	—	—	—
60		Reserved	—	—	—	—	—	—	—	—
61		Reserved	—	—	—	—	—	—	—	—
62		Reserved	—	—	—	—	—	—	—	—
63		Reserved	—	—	—	—	—	—	—	—

Table 39.10 List of Error Inputs of E2x-FCC1 (3/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State
64	Instruction Cache RAM	Reserved	—	—	—	—	—	—	—	—
65		Reserved	—	—	—	—	—	—	—	—
66		Reserved	—	—	—	—	—	—	—	—
67		Reserved	—	—	—	—	—	—	—	—
68		Reserved	—	—	—	—	—	—	—	—
69		Reserved	—	—	—	—	—	—	—	—
70		Reserved	—	—	—	—	—	—	—	—
71		Reserved	—	—	—	—	—	—	—	—
72	Reserved		—	—	—	—	—	—	—	
73	Reserved		—	—	—	—	—	—	—	
74	Reserved		—	—	—	—	—	—	—	
75	Reserved		—	—	—	—	—	—	—	
76	Reserved		—	—	—	—	—	—	—	
77	Reserved		—	—	—	—	—	—	—	
78	Reserved		—	—	—	—	—	—	—	
79	Reserved		—	—	—	—	—	—	—	
80	PE guard function (PEG)	PEG error (PE0)	✓	✓	✓	✓	✓	✓	—	✓
81		PEG error (PE1)	✓	✓	✓	✓	✓	✓	—	✓
82		Reserved	—	—	—	—	—	—	—	—
83		Reserved	—	—	—	—	—	—	—	—
84		Reserved	—	—	—	—	—	—	—	—
85		Reserved	—	—	—	—	—	—	—	—
86		Reserved	—	—	—	—	—	—	—	—
87		Reserved	—	—	—	—	—	—	—	—
88	Clock Monitor	Clock monitor error (CLMA5) (PE0)	✓	✓	✓	✓	✓	✓	—	✓
89		Clock monitor error (CLMA6) (PE1)	✓	✓	✓	✓	✓	✓	—	✓
90		Reserved	—	—	—	—	—	—	—	—
91		Reserved	—	—	—	—	—	—	—	—
92		Reserved	—	—	—	—	—	—	—	—
93		Reserved	—	—	—	—	—	—	—	—
94		Reserved	—	—	—	—	—	—	—	—
95		Reserved	—	—	—	—	—	—	—	—
96	OSTM	OSTM1 Interrupt	✓	✓	✓	✓	✓	✓	—	✓
97		OSTM2 Interrupt	✓	✓	✓	✓	✓	✓	—	✓
98		Reserved	—	—	—	—	—	—	—	—
99		Reserved	—	—	—	—	—	—	—	—
100		Reserved	—	—	—	—	—	—	—	—
101		Reserved	—	—	—	—	—	—	—	—
102		Reserved	—	—	—	—	—	—	—	—

Table 39.10 List of Error Inputs of E2x-FCC1 (4/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State
103	Reserved		—	—	—	—	—	—	—	—
104	Reserved		—	—	—	—	—	—	—	—
105	Reserved		—	—	—	—	—	—	—	—
106	Reserved		—	—	—	—	—	—	—	—
107	Reserved		—	—	—	—	—	—	—	—
108	Reserved		—	—	—	—	—	—	—	—
109	Reserved		—	—	—	—	—	—	—	—
110	Reserved		—	—	—	—	—	—	—	—
111	Reserved		—	—	—	—	—	—	—	—
112	Mode Error	Unintended Debug Enable detection (PE0)	✓	✓	✓	✓	✓	✓	—	✓
113		Unintended Debug Enable detection (PE1)	✓	✓	✓	✓	✓	✓	—	✓
114		Reserved	—	—	—	—	—	—	—	—
115		Reserved	—	—	—	—	—	—	—	—
116		Reserved	—	—	—	—	—	—	—	—
117		Reserved	—	—	—	—	—	—	—	—
118		Reserved	—	—	—	—	—	—	—	—
119		Reserved	—	—	—	—	—	—	—	—
120	PEG error	PEG error (PE0) Detected in a read request from PE0 to the others LRAM	✓	✓	✓	✓	✓	✓	—	✓
121		PEG error (PE1) Detected in a read request from PE1 to the others LRAM	✓	✓	✓	✓	✓	✓	—	✓
122		Reserved	—	—	—	—	—	—	—	—
123		Reserved	—	—	—	—	—	—	—	—
124		Reserved	—	—	—	—	—	—	—	—
125		Reserved	—	—	—	—	—	—	—	—
126		Reserved	—	—	—	—	—	—	—	—
127		Reserved	—	—	—	—	—	—	—	—
128	Reserved		—	—	—	—	—	—	—	—
129	Reserved		—	—	—	—	—	—	—	—
130	Reserved		—	—	—	—	—	—	—	—
131	Reserved		—	—	—	—	—	—	—	—
132	Reserved		—	—	—	—	—	—	—	—
133	Reserved		—	—	—	—	—	—	—	—
134	Reserved		—	—	—	—	—	—	—	—
135	Reserved		—	—	—	—	—	—	—	—
136	Reserved		—	—	—	—	—	—	—	—
137	Reserved		—	—	—	—	—	—	—	—
138	Reserved		—	—	—	—	—	—	—	—
139	Reserved		—	—	—	—	—	—	—	—
140	Reserved		—	—	—	—	—	—	—	—

Table 39.10 List of Error Inputs of E2x-FCC1 (5/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State	
141	Reserved		—	—	—	—	—	—	—	—	
142	Reserved		—	—	—	—	—	—	—	—	
143	Reserved		—	—	—	—	—	—	—	—	
144	Reserved		—	—	—	—	—	—	—	—	
145	Reserved		—	—	—	—	—	—	—	—	
146	Reserved		—	—	—	—	—	—	—	—	
147	Reserved		—	—	—	—	—	—	—	—	
148	Reserved		—	—	—	—	—	—	—	—	
149	Reserved		—	—	—	—	—	—	—	—	
150	Reserved		—	—	—	—	—	—	—	—	
151	Reserved		—	—	—	—	—	—	—	—	
152	Cluster RAM	Cluster RAM • Address feedback compare error	✓	✓	✓	✓	✓	✓	—	✓	
153		Cluster RAM ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
154		Cluster RAM ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
155		Cluster RAM ECC • Error address overflow	✓	✓	✓	✓	✓	✓	—	✓	
156		Reserved		—	—	—	—	—	—	—	—
157		Reserved		—	—	—	—	—	—	—	—
158		Reserved		—	—	—	—	—	—	—	—
159		Reserved		—	—	—	—	—	—	—	—
160	Local RAM (other core)	Reserved	—	—	—	—	—	—	—	—	
161		LRAM (error by other core access) • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
162		LRAM (error by other core access) • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
163		LRAM (error by other core access) • Error address overflow	✓	✓	✓	✓	✓	✓	—	✓	
164	sDMA	sDMAC0 RAM • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
165		sDMAC0 RAM • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
166		sDMAC1 RAM • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
167		sDMAC1 RAM • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	

Table 39.10 List of Error Inputs of E2x-FCC1 (6/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State	
168	Peripheral RAM	Peripheral (DTS) RAM ECC • ECC 2bit error • Address feedback compare error	✓	✓	✓	✓	✓	✓	—	✓	
169		Peripheral (DTS) RAM ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
170		Peripheral (DTS) RAM ECC • Error address overflow	✓	✓	✓	✓	✓	✓	—	✓	
171		Peripheral(except DTS) RAM ECC • Error address overflow	✓	✓	✓	✓	✓	✓	—	✓	
172		Peripheral(FlexRay) RAM ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
173		Peripheral(FlexRay) RAM ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
174		Peripheral(CAN) RAM ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
175		Peripheral(CAN) RAM ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
176		Peripheral(DFE) RAM ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
177		Peripheral(DFE) RAM ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
178		Peripheral(GTM) RAM ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
179		Peripheral(GTM) RAM ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
180		Peripheral(Ethernet) RAM ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
181		Peripheral(Ethernet) RAM ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
182		Reserved	—	—	—	—	—	—	—	—	—
183		Reserved	—	—	—	—	—	—	—	—	—
184		Reserved	—	—	—	—	—	—	—	—	—
185		Reserved	—	—	—	—	—	—	—	—	—
186		Reserved	—	—	—	—	—	—	—	—	—
187		Reserved	—	—	—	—	—	—	—	—	—
188	Reserved	—	—	—	—	—	—	—	—	—	
189	Reserved	—	—	—	—	—	—	—	—	—	
190	Reserved	—	—	—	—	—	—	—	—	—	
191	Reserved	—	—	—	—	—	—	—	—	—	

Table 39.10 List of Error Inputs of E2x-FCC1 (7/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State
192	Peripheral RAM	Reserved	—	—	—	—	—	—	—	—
193		Reserved	—	—	—	—	—	—	—	—
194		Reserved	—	—	—	—	—	—	—	—
195		Reserved	—	—	—	—	—	—	—	—
196		Reserved	—	—	—	—	—	—	—	—
197		Reserved	—	—	—	—	—	—	—	—
198		Reserved	—	—	—	—	—	—	—	—
199		Reserved	—	—	—	—	—	—	—	—
200		Code Flash	Code Flash • Address parity error	✓	✓	✓	✓	✓	✓	—
201	Code Flash ECC • ECC 2bit error		✓	✓	✓	✓	✓	✓	—	✓
202	Code Flash ECC • ECC 1bit error		✓	✓	✓	✓	✓	✓	—	✓
203	Code Flash ECC • Error address overflow		✓	✓	✓	✓	✓	✓	—	✓
204	Reserved		—	—	—	—	—	—	—	—
205	Reserved		—	—	—	—	—	—	—	—
206	Reserved		—	—	—	—	—	—	—	—
207	Reserved		—	—	—	—	—	—	—	—
208	Data Flash		Data Flash ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—
209		Data Flash ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓
210		Data Flash ECC • Error address overflow	✓	✓	✓	✓	✓	✓	—	✓
211		Reserved	—	—	—	—	—	—	—	—
212		Reserved	—	—	—	—	—	—	—	—
213		Reserved	—	—	—	—	—	—	—	—
214		Reserved	—	—	—	—	—	—	—	—
215		Reserved	—	—	—	—	—	—	—	—
216		Bus ECC	Data Bus ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—
217	Data Bus ECC • ECC 1bit error		✓	✓	✓	✓	✓	✓	—	✓
218	Address Bus ECC • EDC 2bit error • EDC 1bit error		✓	✓	✓	✓	✓	✓	—	✓
219	Reserved		—	—	—	—	—	—	—	—
220	Reserved		—	—	—	—	—	—	—	—
221	Reserved		—	—	—	—	—	—	—	—
222	Reserved		—	—	—	—	—	—	—	—
223	Reserved		—	—	—	—	—	—	—	—

Table 39.10 List of Error Inputs of E2x-FCC1 (8/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State
224	Cluster RAM Guard (CRG)	CRAM Guard error	✓	✓	✓	✓	✓	✓	—	✓
225	P-Bus Guard (PBG)	P-Bus Guard error (including INTC2, DTS and sDMAC Guard error)	✓	✓	✓	✓	✓	✓	—	✓
226	H-Bus Guard (HBG)	H-Bus Guard error	✓	✓	✓	✓	✓	✓	—	✓
227	I-Bus Guard (IBG)	I-Bus Guard error (IPIR MEV Barrier TPTM)	✓	✓	✓	✓	✓	✓	—	✓
228	Reserved		—	—	—	—	—	—	—	—
229	Reserved		—	—	—	—	—	—	—	—
230	Reserved		—	—	—	—	—	—	—	—
231	Reserved		—	—	—	—	—	—	—	—
232	Clock Monitor	Clock monitor error (CLMA0)	✓	✓	✓	✓	✓	✓	—	✓
233		Clock monitor error (CLMA1)	✓	✓	✓	✓	✓	✓	—	✓
234		Clock monitor error (CLMA2)	✓	✓	✓	✓	✓	✓	—	✓
235		Clock monitor error (CLMA3)	✓	✓	✓	✓	✓	✓	—	✓
236		Clock monitor error (CLMA4)	✓	✓	✓	✓	✓	✓	—	✓
237	Reserved		—	—	—	—	—	—	—	—
238	Reserved		—	—	—	—	—	—	—	—
239	Reserved		—	—	—	—	—	—	—	—
240	DSADC ADC Cyclic ADC	AD parity error	✓	✓	✓	✓	✓	✓	—	✓
241	MISG	MISG compare error	✓	✓	✓	✓	✓	✓	—	✓
242	DTS	DTS compare error	✓	✓	✓	✓	✓	✓	—	✓
243	External Error Input	ERRORIN*2	✓	✓	✓	✓	✓	✓	—	✓
244	Flash	Flash access error	✓	✓	✓	✓	✓	✓	—	✓
245		FACI reset transfer error	✓	—	—	—	✓	—	—	—
246		BIST parameter transfer error	✓	—	—	—	✓	—	—	—
247		Reserved		—	—	—	—	—	—	—

Table 39.10 List of Error Inputs of E2x-FCC1 (9/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State
248	Reserved		—	—	—	—	—	—	—	—
249	Reserved		—	—	—	—	—	—	—	—
250	Reserved		—	—	—	—	—	—	—	—
251	Reserved		—	—	—	—	—	—	—	—
252	Reserved		—	—	—	—	—	—	—	—
253	Reserved		—	—	—	—	—	—	—	—
254	Reserved		—	—	—	—	—	—	—	—
255	Reserved		—	—	—	—	—	—	—	—
256	Reserved		—	—	—	—	—	—	—	—
257	Reserved		—	—	—	—	—	—	—	—
258	Reserved		—	—	—	—	—	—	—	—
259	Reserved		—	—	—	—	—	—	—	—
260	Reserved		—	—	—	—	—	—	—	—
261	Reserved		—	—	—	—	—	—	—	—
262	Reserved		—	—	—	—	—	—	—	—
263	Reserved		—	—	—	—	—	—	—	—
264	Reserved		—	—	—	—	—	—	—	—
265	Reserved		—	—	—	—	—	—	—	—
266	Reserved		—	—	—	—	—	—	—	—
267	Reserved		—	—	—	—	—	—	—	—
268	Reserved		—	—	—	—	—	—	—	—
269	Reserved		—	—	—	—	—	—	—	—
270	Reserved		—	—	—	—	—	—	—	—
271	Reserved		—	—	—	—	—	—	—	—
272	Reserved		—	—	—	—	—	—	—	—
273	Reserved		—	—	—	—	—	—	—	—
274	Reserved		—	—	—	—	—	—	—	—
275	Reserved		—	—	—	—	—	—	—	—
276	Reserved		—	—	—	—	—	—	—	—
277	Reserved		—	—	—	—	—	—	—	—
278	Reserved		—	—	—	—	—	—	—	—
279	Reserved		—	—	—	—	—	—	—	—
280	Data Path	DMA Comp error	✓	✓	✓	✓	✓	✓	—	✓
281	Redundancy	BUS Bridge Comp error	✓	✓	✓	✓	✓	✓	—	✓

Table 39.10 List of Error Inputs of E2x-FCC1 (10/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State	
282	BUS Routing checker*3	Inter-processor element Bus	✓	✓	✓	✓	✓	✓	—	✓	
283		Inter-cluster Bus (I-Bus)	✓	✓	✓	✓	✓	✓	—	✓	
284		Peripheral Bus (P-Bus)	✓	✓	✓	✓	✓	✓	—	✓	
285		CRAM Bus	✓	✓	✓	✓	✓	✓	—	✓	
286		System Bus	✓	✓	✓	✓	✓	✓	—	✓	
287		Global FLASH Bus	✓	✓	✓	✓	✓	✓	—	✓	
288		Local FLASH Bus	✓	✓	✓	✓	✓	✓	—	✓	
289		Reserved	—	—	—	—	—	—	—	—	—
290		Reserved	—	—	—	—	—	—	—	—	—
291		Reserved	—	—	—	—	—	—	—	—	—
292	Voltage Monitor	EVCC Secondary HDET	✓	✓	✓	✓	✓	✓	—	✓	
293		EVCC Secondary LDET	✓	✓	✓	✓	✓	✓	—	✓	
294		VCC Secondary HDET	✓	✓	✓	✓	✓	✓	—	✓	
295		VCC Secondary LDET	✓	✓	✓	✓	✓	✓	—	✓	
296		VDD Secondary HDET	✓	✓	✓	✓	✓	✓	—	✓	
297		VDD Secondary LDET	✓	✓	✓	✓	✓	✓	—	✓	
298		Reserved	—	—	—	—	—	—	—	—	—
299	Mode Error	Mode error • Unintended activation of Production Test Mode	✓	✓	✓	✓	✓	✓	—	✓	
300		Mode error • Unintended activation of Normal Operation Mode	✓	✓	✓	✓	✓	✓	—	✓	
301		Mode error • Unintended deactivation of Normal Operation Mode	✓	✓	✓	✓	✓	✓	—	✓	
302		Mode error • Unintended activation of Serial Programming Mode	✓	✓	✓	✓	✓	✓	—	✓	
303		Mode error • Unintended activation of User Boot Mode	✓	✓	✓	✓	✓	✓	—	✓	
304		Mode error • Unintended deactivation of User Boot Mode	✓	✓	✓	✓	✓	✓	—	✓	
305		Mode error • Mode latch error	✓	✓	✓	✓	✓	✓	—	✓	
306	Reserved	—	—	—	—	—	—	—	—	—	
307	Reserved	—	—	—	—	—	—	—	—	—	
308	ECM	ECM compare error	✓	✓	✓	✓	✓	✓	—	✓	

Note 1. The internal reset generation is enabled in the initial state.

Note 2. Please enable DNF for noise elimination if ERRORIN function is used. There is a possibility that ECMmESSTRn for ERRORIN is set. Therefore please set ECMEMKn for ERRORIN before setting DNF and clear ECMESSTCn for ERRORIN after setting DNF. Then please clear ECMEMKn if needed. About DNF setting flow, refer to **Section 2.6.1.2, Peripheral Function DNF**.

Note 3. BUS routing checker can detect an unintended routing on internal bus.

Table 39.11 List of Error Inputs of E2M (1/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State
0	Dual Core Lock-step	DCLS compare error (PE0)	✓	✓	✓	✓	✓	✓	✓	✓
1		DCLS compare error (PE1)	✓	✓	✓	✓	✓	✓	✓	✓
2		Reserved	—	—	—	—	—	—	—	—
3		Reserved	—	—	—	—	—	—	—	—
4		Reserved	—	—	—	—	—	—	—	—
5		Reserved	—	—	—	—	—	—	—	—
6		Reserved	—	—	—	—	—	—	—	—
7		Reserved	—	—	—	—	—	—	—	—
8	Watchdog timer	Watchdog timer ch0 error (PE0)	✓	✓	✓	✓*1	✓	✓	—	✓
9		Watchdog timer ch1 error (PE1)	✓	✓	✓	✓	✓	✓	—	✓
10		Reserved	—	—	—	—	—	—	—	—
11		Reserved	—	—	—	—	—	—	—	—
12		Reserved	—	—	—	—	—	—	—	—
13		Reserved	—	—	—	—	—	—	—	—
14		Reserved	—	—	—	—	—	—	—	—
15		Reserved	—	—	—	—	—	—	—	—
16	Local RAM (own core)	Local RAM (PE0) • Address feedback compare error	✓	✓	✓	✓	✓	✓	—	✓
17		Local RAM (PE1) • Address feedback compare error	✓	✓	✓	✓	✓	✓	—	✓
18		Reserved	—	—	—	—	—	—	—	—
19		Reserved	—	—	—	—	—	—	—	—
20		Reserved	—	—	—	—	—	—	—	—
21		Reserved	—	—	—	—	—	—	—	—
22		Reserved	—	—	—	—	—	—	—	—
23		Reserved	—	—	—	—	—	—	—	—
24		Local RAM ECC (PE0) • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓
25		Local RAM ECC (PE1) • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓
26		Reserved	—	—	—	—	—	—	—	—
27		Reserved	—	—	—	—	—	—	—	—
28		Reserved	—	—	—	—	—	—	—	—
29		Reserved	—	—	—	—	—	—	—	—
30		Reserved	—	—	—	—	—	—	—	—

Table 39.11 List of Error Inputs of E2M (2/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State
31	Local RAM (own core)	Reserved	—	—	—	—	—	—	—	—
32		Local RAM ECC (PE0) • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓
33		Local RAM ECC (PE1) • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓
34		Reserved	—	—	—	—	—	—	—	—
35		Reserved	—	—	—	—	—	—	—	—
36		Reserved	—	—	—	—	—	—	—	—
37		Reserved	—	—	—	—	—	—	—	—
38		Reserved	—	—	—	—	—	—	—	—
39		Reserved	—	—	—	—	—	—	—	—
40		Local RAM ECC (PE0) • Error address overflow	✓	✓	✓	✓	✓	✓	—	✓
41		Local RAM ECC (PE1) • Error address overflow	✓	✓	✓	✓	✓	✓	—	✓
42		Reserved	—	—	—	—	—	—	—	—
43		Reserved	—	—	—	—	—	—	—	—
44		Reserved	—	—	—	—	—	—	—	—
45		Reserved	—	—	—	—	—	—	—	—
46		Reserved	—	—	—	—	—	—	—	—
47		Reserved	—	—	—	—	—	—	—	—
48	Instruction Cache RAM	Instruction Cache RAM (PE0) • Address feedback compare error	✓	✓	✓	✓	✓	✓	—	✓
49		Instruction Cache RAM (PE1) • Address feedback compare error	✓	✓	✓	✓	✓	✓	—	✓
50		Reserved	—	—	—	—	—	—	—	—
51		Reserved	—	—	—	—	—	—	—	—
52		Reserved	—	—	—	—	—	—	—	—
53		Reserved	—	—	—	—	—	—	—	—
54		Reserved	—	—	—	—	—	—	—	—
55		Reserved	—	—	—	—	—	—	—	—
56		Instruction Cache RAM EDC (PE0) • EDC 2bit error • EDC 1bit error	✓	✓	✓	✓	✓	✓	—	✓
57		Instruction Cache RAM EDC (PE1) • EDC 2bit error • EDC 1bit error	✓	✓	✓	✓	✓	✓	—	✓
58		Reserved	—	—	—	—	—	—	—	—
59		Reserved	—	—	—	—	—	—	—	—
60		Reserved	—	—	—	—	—	—	—	—
61		Reserved	—	—	—	—	—	—	—	—
62		Reserved	—	—	—	—	—	—	—	—
63		Reserved	—	—	—	—	—	—	—	—

Table 39.11 List of Error Inputs of E2M (3/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State
64	Instruction Cache RAM	Reserved	—	—	—	—	—	—	—	—
65		Reserved	—	—	—	—	—	—	—	—
66		Reserved	—	—	—	—	—	—	—	—
67		Reserved	—	—	—	—	—	—	—	—
68		Reserved	—	—	—	—	—	—	—	—
69		Reserved	—	—	—	—	—	—	—	—
70		Reserved	—	—	—	—	—	—	—	—
71		Reserved	—	—	—	—	—	—	—	—
72	Reserved		—	—	—	—	—	—	—	
73	Reserved		—	—	—	—	—	—	—	
74	Reserved		—	—	—	—	—	—	—	
75	Reserved		—	—	—	—	—	—	—	
76	Reserved		—	—	—	—	—	—	—	
77	Reserved		—	—	—	—	—	—	—	
78	Reserved		—	—	—	—	—	—	—	
79	Reserved		—	—	—	—	—	—	—	
80	PE guard function (PEG)	PEG error (PE0)	✓	✓	✓	✓	✓	✓	—	✓
81		PEG error (PE1)	✓	✓	✓	✓	✓	✓	—	✓
82		Reserved	—	—	—	—	—	—	—	—
83		Reserved	—	—	—	—	—	—	—	—
84		Reserved	—	—	—	—	—	—	—	—
85		Reserved	—	—	—	—	—	—	—	—
86		Reserved	—	—	—	—	—	—	—	—
87		Reserved	—	—	—	—	—	—	—	—
88	Clock Monitor	Clock monitor error (CLMA5) (PE0)	✓	✓	✓	✓	✓	✓	—	✓
89		Clock monitor error (CLMA6) (PE1)	✓	✓	✓	✓	✓	✓	—	✓
90		Reserved	—	—	—	—	—	—	—	—
91		Reserved	—	—	—	—	—	—	—	—
92		Reserved	—	—	—	—	—	—	—	—
93		Reserved	—	—	—	—	—	—	—	—
94		Reserved	—	—	—	—	—	—	—	—
95		Reserved	—	—	—	—	—	—	—	—
96	OSTM	OSTM1 Interrupt	✓	✓	✓	✓	✓	✓	—	✓
97		OSTM2 Interrupt	✓	✓	✓	✓	✓	✓	—	✓
98		Reserved	—	—	—	—	—	—	—	—
99		Reserved	—	—	—	—	—	—	—	—
100		Reserved	—	—	—	—	—	—	—	—
101		Reserved	—	—	—	—	—	—	—	—
102		Reserved	—	—	—	—	—	—	—	—

Table 39.11 List of Error Inputs of E2M (4/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State
103	Reserved		—	—	—	—	—	—	—	—
104	Reserved		—	—	—	—	—	—	—	—
105	Reserved		—	—	—	—	—	—	—	—
106	Reserved		—	—	—	—	—	—	—	—
107	Reserved		—	—	—	—	—	—	—	—
108	Reserved		—	—	—	—	—	—	—	—
109	Reserved		—	—	—	—	—	—	—	—
110	Reserved		—	—	—	—	—	—	—	—
111	Reserved		—	—	—	—	—	—	—	—
112	Mode Error	Unintended Debug Enable detection (PE0)	✓	✓	✓	✓	✓	✓	—	✓
113		Unintended Debug Enable detection (PE1)	✓	✓	✓	✓	✓	✓	—	✓
114		Reserved	—	—	—	—	—	—	—	—
115		Reserved	—	—	—	—	—	—	—	—
116		Reserved	—	—	—	—	—	—	—	—
117		Reserved	—	—	—	—	—	—	—	—
118		Reserved	—	—	—	—	—	—	—	—
119		Reserved	—	—	—	—	—	—	—	—
120	PEG error	PEG error (PE0) Detected in a read request from PE0 to the others LRAM	✓	✓	✓	✓	✓	✓	—	✓
121		PEG error (PE1) Detected in a read request from PE1 to the others LRAM	✓	✓	✓	✓	✓	✓	—	✓
122		Reserved	—	—	—	—	—	—	—	—
123		Reserved	—	—	—	—	—	—	—	—
124		Reserved	—	—	—	—	—	—	—	—
125		Reserved	—	—	—	—	—	—	—	—
126		Reserved	—	—	—	—	—	—	—	—
127		Reserved	—	—	—	—	—	—	—	—
128	Reserved		—	—	—	—	—	—	—	—
129	Reserved		—	—	—	—	—	—	—	—
130	Reserved		—	—	—	—	—	—	—	—
131	Reserved		—	—	—	—	—	—	—	—
132	Reserved		—	—	—	—	—	—	—	—
133	Reserved		—	—	—	—	—	—	—	—
134	Reserved		—	—	—	—	—	—	—	—
135	Reserved		—	—	—	—	—	—	—	—
136	Reserved		—	—	—	—	—	—	—	—
137	Reserved		—	—	—	—	—	—	—	—
138	Reserved		—	—	—	—	—	—	—	—
139	Reserved		—	—	—	—	—	—	—	—
140	Reserved		—	—	—	—	—	—	—	—

Table 39.11 List of Error Inputs of E2M (5/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State	
141	Reserved		—	—	—	—	—	—	—	—	
142	Reserved		—	—	—	—	—	—	—	—	
143	Reserved		—	—	—	—	—	—	—	—	
144	Reserved		—	—	—	—	—	—	—	—	
145	Reserved		—	—	—	—	—	—	—	—	
146	Reserved		—	—	—	—	—	—	—	—	
147	Reserved		—	—	—	—	—	—	—	—	
148	Reserved		—	—	—	—	—	—	—	—	
149	Reserved		—	—	—	—	—	—	—	—	
150	Reserved		—	—	—	—	—	—	—	—	
151	Reserved		—	—	—	—	—	—	—	—	
152	Cluster RAM	Cluster RAM • Address feedback compare error	✓	✓	✓	✓	✓	✓	—	✓	
153		Cluster RAM ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
154		Cluster RAM ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
155		Cluster RAM ECC • Error address overflow	✓	✓	✓	✓	✓	✓	—	✓	
156		Reserved		—	—	—	—	—	—	—	—
157		Reserved		—	—	—	—	—	—	—	—
158		Reserved		—	—	—	—	—	—	—	—
159		Reserved		—	—	—	—	—	—	—	—
160	Local RAM (other core)	Reserved	—	—	—	—	—	—	—	—	
161		LRAM (error by other core access) • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
162		LRAM (error by other core access) • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
163		LRAM (error by other core access) • Error address overflow	✓	✓	✓	✓	✓	✓	—	✓	
164	sDMA	sDMAC0 RAM • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
165		sDMAC0 RAM • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
166		sDMAC1 RAM • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
167		sDMAC1 RAM • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	

Table 39.11 List of Error Inputs of E2M (6/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State	
168	Peripheral RAM	Peripheral (DTS) RAM ECC • ECC 2bit error • Address feedback compare error	✓	✓	✓	✓	✓	✓	—	✓	
169		Peripheral (DTS) RAM ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
170		Peripheral (DTS) RAM ECC • Error address overflow	✓	✓	✓	✓	✓	✓	—	✓	
171		Peripheral(except DTS) RAM ECC • Error address overflow	✓	✓	✓	✓	✓	✓	—	✓	
172		Peripheral(FlexRay) RAM ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
173		Peripheral(FlexRay) RAM ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
174		Peripheral(CAN) RAM ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
175		Peripheral(CAN) RAM ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
176		Peripheral(DFE) RAM ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
177		Peripheral(DFE) RAM ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
178		Peripheral(GTM) RAM ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
179		Peripheral(GTM) RAM ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
180		Peripheral(Ethernet) RAM ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓	
181		Peripheral(Ethernet) RAM ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓	
182		Reserved	—	—	—	—	—	—	—	—	—
183		Reserved	—	—	—	—	—	—	—	—	—
184		Reserved	—	—	—	—	—	—	—	—	—
185		Reserved	—	—	—	—	—	—	—	—	—
186		Reserved	—	—	—	—	—	—	—	—	—
187		Reserved	—	—	—	—	—	—	—	—	—
188	Reserved	—	—	—	—	—	—	—	—	—	
189	Reserved	—	—	—	—	—	—	—	—	—	
190	Reserved	—	—	—	—	—	—	—	—	—	
191	Reserved	—	—	—	—	—	—	—	—	—	

Table 39.11 List of Error Inputs of E2M (7/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State
192	Peripheral RAM	Reserved	—	—	—	—	—	—	—	—
193		Reserved	—	—	—	—	—	—	—	—
194		Reserved	—	—	—	—	—	—	—	—
195		Reserved	—	—	—	—	—	—	—	—
196		Reserved	—	—	—	—	—	—	—	—
197		Reserved	—	—	—	—	—	—	—	—
198		Reserved	—	—	—	—	—	—	—	—
199		Reserved	—	—	—	—	—	—	—	—
200	Code Flash	Code Flash • Address parity error	✓	✓	✓	✓	✓	✓	—	✓
201		Code Flash ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓
202		Code Flash ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓
203		Code Flash ECC • Error address overflow	✓	✓	✓	✓	✓	✓	—	✓
204		Reserved	—	—	—	—	—	—	—	—
205		Reserved	—	—	—	—	—	—	—	—
206		Reserved	—	—	—	—	—	—	—	—
207		Reserved	—	—	—	—	—	—	—	—
208	Data Flash	Data Flash ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓
209		Data Flash ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓
210		Data Flash ECC • Error address overflow	✓	✓	✓	✓	✓	✓	—	✓
211		Reserved	—	—	—	—	—	—	—	—
212		Reserved	—	—	—	—	—	—	—	—
213		Reserved	—	—	—	—	—	—	—	—
214		Reserved	—	—	—	—	—	—	—	—
215		Reserved	—	—	—	—	—	—	—	—
216	Bus ECC	Data Bus ECC • ECC 2bit error	✓	✓	✓	✓	✓	✓	—	✓
217		Data Bus ECC • ECC 1bit error	✓	✓	✓	✓	✓	✓	—	✓
218		Address Bus ECC • EDC 2bit error • EDC 1bit error	✓	✓	✓	✓	✓	✓	—	✓
219		Reserved	—	—	—	—	—	—	—	—
220		Reserved	—	—	—	—	—	—	—	—
221		Reserved	—	—	—	—	—	—	—	—
222		Reserved	—	—	—	—	—	—	—	—
223		Reserved	—	—	—	—	—	—	—	—

Table 39.11 List of Error Inputs of E2M (8/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State
224	Cluster RAM Guard (CRG)	CRAM Guard error	✓	✓	✓	✓	✓	✓	—	✓
225	P-Bus Guard (PBG)	P-Bus Guard error (including INTC2, DTS and sDMAC Guard error)	✓	✓	✓	✓	✓	✓	—	✓
226	H-Bus Guard (HBG)	H-Bus Guard error	✓	✓	✓	✓	✓	✓	—	✓
227	I-Bus Guard (IBG)	I-Bus Guard error (IPIR MEV Barrier TPTM)	✓	✓	✓	✓	✓	✓	—	✓
228	Reserved		—	—	—	—	—	—	—	—
229	Reserved		—	—	—	—	—	—	—	—
230	Reserved		—	—	—	—	—	—	—	—
231	Reserved		—	—	—	—	—	—	—	—
232	Clock Monitor	Clock monitor error (CLMA0)	✓	✓	✓	✓	✓	✓	—	✓
233		Clock monitor error (CLMA1)	✓	✓	✓	✓	✓	✓	—	✓
234		Clock monitor error (CLMA2)	✓	✓	✓	✓	✓	✓	—	✓
235		Clock monitor error (CLMA3)	✓	✓	✓	✓	✓	✓	—	✓
236		Clock monitor error (CLMA4)	✓	✓	✓	✓	✓	✓	—	✓
237	Reserved		—	—	—	—	—	—	—	—
238	Reserved		—	—	—	—	—	—	—	—
239	Reserved		—	—	—	—	—	—	—	—
240	DSADC ADC Cyclic ADC	AD parity error	✓	✓	✓	✓	✓	✓	—	✓
241	MISG	MISG compare error	✓	✓	✓	✓	✓	✓	—	✓
242	DTS	DTS compare error	✓	✓	✓	✓	✓	✓	—	✓
243	External Error Input	ERRORIN*2	✓	✓	✓	✓	✓	✓	—	✓
244	Flash	Flash access error	✓	✓	✓	✓	✓	✓	—	✓
245		FACI reset transfer error	✓	—	—	—	✓	—	—	—
246		BIST parameter transfer error	✓	—	—	—	✓	—	—	—
247		Reserved		—	—	—	—	—	—	—

Table 39.11 List of Error Inputs of E2M (9/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State
248	Reserved		—	—	—	—	—	—	—	—
249	Reserved		—	—	—	—	—	—	—	—
250	Reserved		—	—	—	—	—	—	—	—
251	Reserved		—	—	—	—	—	—	—	—
252	Reserved		—	—	—	—	—	—	—	—
253	Reserved		—	—	—	—	—	—	—	—
254	Reserved		—	—	—	—	—	—	—	—
255	Reserved		—	—	—	—	—	—	—	—
256	Reserved		—	—	—	—	—	—	—	—
257	Reserved		—	—	—	—	—	—	—	—
258	Reserved		—	—	—	—	—	—	—	—
259	Reserved		—	—	—	—	—	—	—	—
260	Reserved		—	—	—	—	—	—	—	—
261	Reserved		—	—	—	—	—	—	—	—
262	Reserved		—	—	—	—	—	—	—	—
263	Reserved		—	—	—	—	—	—	—	—
264	Reserved		—	—	—	—	—	—	—	—
265	Reserved		—	—	—	—	—	—	—	—
266	Reserved		—	—	—	—	—	—	—	—
267	Reserved		—	—	—	—	—	—	—	—
268	Reserved		—	—	—	—	—	—	—	—
269	Reserved		—	—	—	—	—	—	—	—
270	Reserved		—	—	—	—	—	—	—	—
271	Reserved		—	—	—	—	—	—	—	—
272	Reserved		—	—	—	—	—	—	—	—
273	Reserved		—	—	—	—	—	—	—	—
274	Reserved		—	—	—	—	—	—	—	—
275	Reserved		—	—	—	—	—	—	—	—
276	Reserved		—	—	—	—	—	—	—	—
277	Reserved		—	—	—	—	—	—	—	—
278	Reserved		—	—	—	—	—	—	—	—
279	Reserved		—	—	—	—	—	—	—	—
280	Data Path Redundancy	DMA Comp error	✓	✓	✓	✓	✓	✓	—	✓
281		BUS Bridge Comp error	✓	✓	✓	✓	✓	✓	—	✓

Table 39.11 List of Error Inputs of E2M (10/10)

No.	Module	Error Sources	Error Flag Set	Maskable Interrupt	FE level Interrupt	Internal Reset	ERROROUT Output	Delay Timer Start	DCLS Error Interrupt	Port Safe State	
282	BUS Routing checker*3	Inter-processor element Bus	✓	✓	✓	✓	✓	✓	—	✓	
283		Inter-cluster Bus (I-Bus)	✓	✓	✓	✓	✓	✓	—	✓	
284		Peripheral Bus (P-Bus)	✓	✓	✓	✓	✓	✓	—	✓	
285		CRAM Bus	✓	✓	✓	✓	✓	✓	—	✓	
286		System Bus	✓	✓	✓	✓	✓	✓	—	✓	
287		Global FLASH Bus	✓	✓	✓	✓	✓	✓	—	✓	
288		Local FLASH Bus	✓	✓	✓	✓	✓	✓	—	✓	
289		Reserved	—	—	—	—	—	—	—	—	—
290		Reserved	—	—	—	—	—	—	—	—	—
291		Reserved	—	—	—	—	—	—	—	—	—
292	Voltage Monitor	EVCC Secondary HDET	✓	✓	✓	✓	✓	✓	—	✓	
293		EVCC Secondary LDET	✓	✓	✓	✓	✓	✓	—	✓	
294		VCC Secondary HDET	✓	✓	✓	✓	✓	✓	—	✓	
295		VCC Secondary LDET	✓	✓	✓	✓	✓	✓	—	✓	
296		VDD Secondary HDET	✓	✓	✓	✓	✓	✓	—	✓	
297		VDD Secondary LDET	✓	✓	✓	✓	✓	✓	—	✓	
298		Reserved	—	—	—	—	—	—	—	—	—
299	Mode Error	Mode error • Unintended activation of Production Test Mode	✓	✓	✓	✓	✓	✓	—	✓	
300		Mode error • Unintended activation of Normal Operation Mode	✓	✓	✓	✓	✓	✓	—	✓	
301		Mode error • Unintended deactivation of Normal Operation Mode	✓	✓	✓	✓	✓	✓	—	✓	
302		Mode error • Unintended activation of Serial Programming Mode	✓	✓	✓	✓	✓	✓	—	✓	
303		Mode error • Unintended activation of User Boot Mode	✓	✓	✓	✓	✓	✓	—	✓	
304		Mode error • Unintended deactivation of User Boot Mode	✓	✓	✓	✓	✓	✓	—	✓	
305		Mode error • Mode latch error	✓	✓	✓	✓	✓	✓	—	✓	
306	Reserved	—	—	—	—	—	—	—	—	—	
307	Reserved	—	—	—	—	—	—	—	—	—	
308	ECM	ECM compare error	✓	✓	✓	✓	✓	✓	—	✓	

Note 1. The internal reset generation is enabled in the initial state.

Note 2. Please enable DNF for noise elimination if ERRORIN function is used. There is a possibility that ECMmESSTRn for ERRORIN is set. Therefore please set ECMEMKn for ERRORIN before setting DNF and clear ECMESSTCn for ERRORIN after setting DNF. Then please clear ECMEMKn if needed. About DNF setting flow, refer to **Section 2.6.1.2, Peripheral Function DNF**.

Note 3. BUS routing checker can detect an unintended routing on internal bus.

39.2.4 Operations for Error Output

After reset release, the $\overline{\text{ERROROUT_M}}$ pin outputs the low (error) level. Follow the procedure described in **Section 39.3.3, ECMmECLR — ECM Master/Checker Error Clear Trigger Register (m = M/C)**, to clear the error before using ECM. As the $\overline{\text{ERROROUT_C}}$ pin is multiplexed with a general-purpose port and other functions, select the $\overline{\text{ERROROUT_C}}$ function before use. For settings, see **Section 2, Pin Function**.

The error output can be configured for two different modes of operation, non-dynamic or dynamic.

The error output is in synchronization with the occurrence of error source conditions and the error level is output as the pin state regardless of the dynamic mode pulse cycle.

ERROROUTZ connects to port safe state and ECM can control the state of general purpose I/O to a condition according to user configuration. For details of function, see **Section 2, Pin Function**. ERROROUTZ can be used as only non-dynamic mode.

Table 39.12 Operation for Error Output

Error Status ECMmSSE	Operating Mode		Error Output		
	ECMEPCFG.ECMSL0 Bit	Operating Mode	Error Output Level*1	ERROROUTZ*1	Error Status
0	0	Non-dynamic	H	H	No error
	1	Dynamic	Toggles (according to timer input)	H	No error
1	0	Non-dynamic	L	L	Error
	1	Dynamic	L	L	Error

Note 1. The level of the error output ($\overline{\text{ERROROUT_M}}$ and $\overline{\text{ERROROUT_C}}$) and ERROROUTZ can be masked by ECMEMKn registers value.

39.2.5 ERROROUT_M and ERROROUT_C Behavior at Reset

Below table explains the behavior of the error output logic and the ERROROUT_M and ERROROUT_C pins at reset. Also the level of the ERROROUT_M and ERROROUT_C signals during and after reset is explained.

Table 39.13 ERROROUT_M Behavior at Reset

Category	Reset Condition					
	Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
<u>ERROROUT_M</u> pin level during reset	Low level	Low level	Level according to error status before reset*1	Level according to error status before reset*1	Level according to error status before reset	Level according to error status before reset
<u>ERROROUT_M</u> pin level after reset	Low level	Low level	Level according to error status before reset*1	Level according to error status before reset*1	Level according to error status before reset	Level according to error status before reset

Note 1. The level of the ERROROUT_M can be changed by clearing ECMEMKn registers at reset. To keep the level, it is necessary to clear the ECMMESSTRn bit which corresponds to ECMEMKn bit of setting which masked error signal output before reset.

Table 39.14 ERROROUT_C Behavior at Reset

Category	Reset Condition					
	Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
<u>ERROROUT_C</u> pin level during reset	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Level according to error status before reset	Level according to error status before reset
<u>ERROROUT_C</u> pin level after reset	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Level according to error status before reset	Level according to error status before reset

39.2.5.1 Dynamic Mode Enable

1. Initialize the related timer OSTM0.
2. Set the error output to high level by setting the ECMmECT (m = M/C) bit in the ECM master/checker error clear trigger register to 1.
3. Set the ECMEPCFG.ECMSL0 bit to 1 for dynamic mode.
4. Start the timer OSTM0.

39.2.5.2 Dynamic Mode Disable

1. Set the error output to low level by setting the ECMmEST bit (m = M/C) in the ECM master/checker error set trigger register to 1.
2. Stop the timer OSTM0.
3. Clear the ECMSL0 bit in the ECM error pulse configuration register to 0 to specify non-dynamic mode.

39.2.6 Loop-Back Function

ECM incorporates a loop-back function that is used to check the path to the error output pin. The output level of the error output pin can be checked with the ECMmSSE311 bit (m = M/C) in the ECM master/checker error source status register 9.

39.2.7 Pseudo Error Generation

ECM incorporates a function that can generate pseudo errors for test or debug purposes. The operation of the ECM during injection of pseudo errors is identical to that for the occurrence of real errors. All configurations for error masks, interrupt, internal reset, or delay timer apply in the same way.

39.2.8 Error Status

The error status is indicated by ECM master/checker error source status registers. The error status is only cleared by writing to ECMESSTCn, power up reset or standby reset. In case of reset except for power up reset and standby reset, the error status is kept and the error that triggered the reset can be confirmed by reading the ECM master/checker error source status registers after reset release.

39.2.9 Register Protection

Write protected registers are protected from inadvertent write access due to erroneous program execution, etc.

By releasing the protection of ECMKCROT, it can be written. For details, see **Section 39.3.11, ECMKCROT — ECM Key Code Protection Register**.

39.2.10 Timeout Function for Interrupt Processing

The delay timer incorporated in ECM can be started simultaneously with the occurrence of an interrupt request. ECM incorporates a function that generates an error signal output or Error Control Module Reset when the count value of the delay timer matches with the value of the delay timer compare register because the delay timer was not stopped during the interrupt processing. The timer counting is not stopped when a break occurs.

The counting of the delay timer always starts from 0. Configure the duration until an Error Control Module Reset or error output is generated with the settings of the delay timer compare register.

In response to the occurrence of a new error source condition with the setting to start the delay timer while the delay timer is operating, the counter value of the delay timer which is currently operating is not reset. Instead, counting by the timer continues.

39.2.11 Masking of “Error Clear Trigger Registers”

The active error output status must be cleared by software via the Error clear trigger registers (ECMMECLR/ECMCECLR). A minimum activation time of the error output is achieved by the Error output clear invalidation counter. This counter is (re)started each time a new error event is triggered at the ECM. It counts up from 0000_H to FFFF_H. Error output clear by software is not possible unless this counter reaches the compare value configured in the ECMEOCCFG register. If Error output clear invalidation counter is still running, Error output clear is masked and Error output clear request by software is ignored.

39.2.12 DCLS Error Interrupt (EI Level) and EI Level Interrupt

This module can generate specific interrupts of DCLS error for each CPU. When DCLS error is occurred in the state of setting the ECMMICFG0, DCLS error interrupt and EI level interrupt are generated. The following figure shows the generation logic of DCLS error interrupt and EI level interrupt from ECM. For each PEn (n = 0, 1), EI level interrupt of CPU_n is generated by EI level interrupt from ECM master, ECM checker and other DCLS error interrupt except self's DCLS error interrupt being OR-ed.

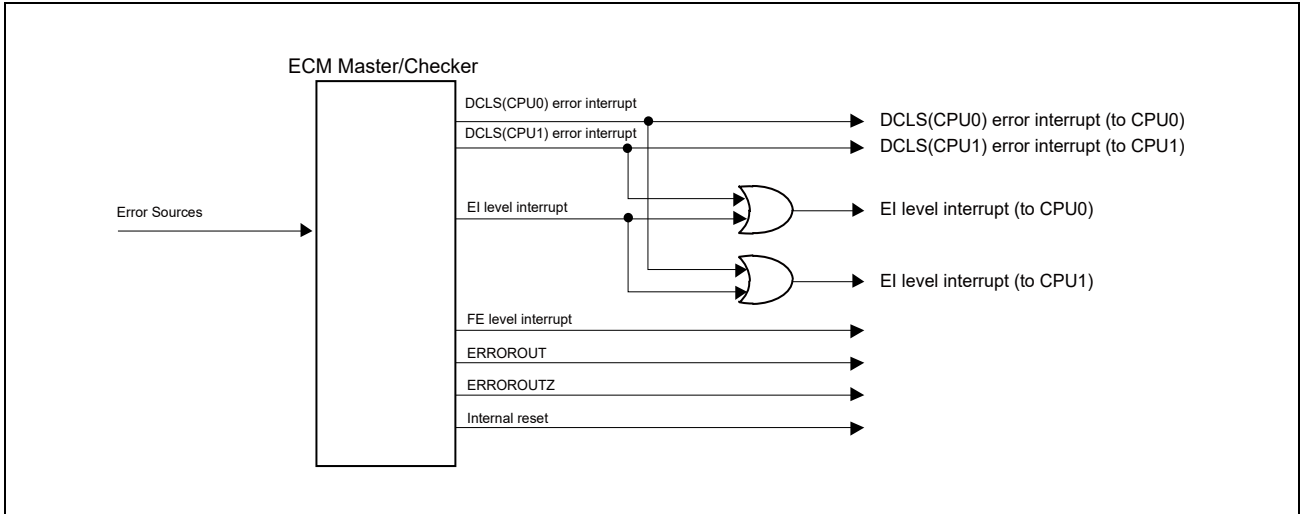


Figure 39.3 DCLS Error Interrupt (EI Level) and EI Level Interrupt

39.3 Register Specification

39.3.1 List of Registers

ECM consists of three address areas: ECM master, ECM checker, ECM common.

The following shows the register map of the ECM master and checker registers.

Table 39.15 Address List of ECM Master and Checker Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
ECM Master Registers <ECMM_base: FFCB 0000 _H >					
ECMM	ECM Master Error Set Trigger Register	ECMMESET	<ECMM_base> + 00 _H	8	ECMKCPROT
ECMM	ECM Master Error Clear Trigger Register	ECMMECLR	<ECMM_base> + 04 _H	8	ECMKCPROT
ECMM	ECM Master Error Source Status Register n ^{*1}	ECMMESSTRn ^{*1}	<ECMM_base> + 08 _H to 2C _H	32	—
ECM Checker Registers <ECMC_base: FFCB 1000 _H >					
ECMC	ECM Checker Error Set Trigger Register	ECMCESET	<ECMC_base> + 00 _H	8	ECMKCPROT
ECMC	ECM Checker Error Clear Trigger Register	ECMCECLR	<ECMC_base> + 04 _H	8	ECMKCPROT
ECMC	ECM Checker Error Source Status Register n ^{*1}	ECMCESSTRn ^{*1}	<ECMC_base> + 08 _H to 2C _H	32	—

Note 1. n = 0 to 9

The following shows the register map of the ECM common registers.

Table 39.16 Address List of ECM Common Registers

Module Name	Register Name	Symbol Name	Address	Access	Access Protection
ECM Common Registers <ECM_base: FFCB 2000 _H >					
ECM	ECM Error Pulse Configuration Register	ECMEPCFG	<ECM_base> + 00 _H	8	ECMKCPROT
ECM	ECM Maskable Interrupt Configuration Register n*1	ECMMICFGn*1	<ECM_base> + 04 _H ~28 _H	32	ECMKCPROT
ECM	ECM FE Level Interrupt Configuration Register n*1	ECNMICFGn*1	<ECM_base> + 2C _H ~50 _H	32	ECMKCPROT
ECM	ECM Internal Reset Configuration Register n*1	ECMIRCFGn*1	<ECM_base> + 54 _H ~78 _H	32	ECMKCPROT
ECM	ECM Error Mask Register n*1	ECMEMKn*1	<ECM_base> + 7C _H ~A0 _H	32	ECMKCPROT
ECM	ECM Error Source Status Clear Trigger Register n*1	ECMESSTCn*1	<ECM_base> + A4 _H ~C8 _H	32	ECMKCPROT
ECM	ECM Key Code Protection Register	ECMKCPROT	<ECM_base> + CC _H	32	—
ECM	ECM Pseudo Error Trigger Register n*1	ECMPEn*1	<ECM_base> + D0 _H to F4 _H	32	ECMKCPROT
ECM	ECM Delay Timer Control Register	ECMDTMCTL	<ECM_base> + F8 _H	8	ECMKCPROT
ECM	ECM Delay Timer Register	ECMDTMR	<ECM_base> + FC _H	16	—
ECM	ECM Delay Timer Compare Register	ECMDTMCMP	<ECM_base> + 100 _H	32	ECMKCPROT
ECM	ECM Maskable Interrupt Delay Timer Configuration Register n*1	ECMMIDTMCFGn*1	<ECM_base> + 104 _H to 128 _H	32	ECMKCPROT
ECM	ECM FE Level Interrupt Delay Timer Configuration Register n*1	ECNMIDTMCFGn*1	<ECM_base> + 12C _H to 150 _H	32	ECMKCPROT
ECM	ECM Error Output Clear Invalidation Configuration Register	ECMEOCCFG	<ECM_base> + 154 _H	32	ECMKCPROT
ECM	ECM Pseudo Error Mask Register	ECMPPEM	<ECM_base> + 158 _H	32	—

Note 1. n = 0 to 9

The register areas of the ECM registers are common to the redundancy area to be implemented. Writes to the common register areas are conducted simultaneously. The common area for ECM master is read by reading access to the common area. The ECM master register and the ECM checker register represent the address areas which can be written separately.

39.3.2 ECMmESET — ECM Master/Checker Error Set Trigger Register (m = M/C)

The ECM Master/Checker Error Set Trigger Register is for setting the error signal from the error pin to the low level. When the ECMmEST bit is set to 1, the error pin outputs the low level. The output cannot be masked. This registers is also the trigger for the port safe state function. When Error signal output is set, each port state will be changed to a condition according to user's setting. Writing to this register is protected by ECMKCPROT. Refer to **Section 39.3.11, ECMKCPROT — ECM Key Code Protection Register** for the details of key code protection. This register is always read as 00_H.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ECMmEST
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 39.17 ECMmESET Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ECMmEST	Error set trigger bit 0: Writing 0 is invalid 1: Sets the output level from the error pin to the active (low) level.

CAUTIONS

Setting or clearing the error output from the $\overline{\text{ERROROUT_M}}$ and $\overline{\text{ERROROUT_C}}$ pin via the ECMmESET or ECMmECLR register will set the ECMmSSE308 bit of the ECMmESSTR9 register (ECM compare error). Therefore, the ECMmESET register has to be set following the sequence below.

1. Set the MSKM bit and MSKC bit of the ECMPPEM register to "masked".
2. Set the ECMmEST bit in the ECMmESET register.
3. Wait until $\overline{\text{ERROROUT_M}}$ and $\overline{\text{ERROROUT_C}}$ become low by checking that the ECMmSSE311 bit of the ECMmESSTR9 register is "0".
4. Set the MSKM bit and MSKC bit of the ECMPPEM register to "not masked".

39.3.3 ECMmECLR — ECM Master/Checker Error Clear Trigger Register (m = M/C)

The ECM Master/Checker Error Clear Trigger Register is for setting the error signal from the error pin to the high level (toggle). When the ECMmECT bit is set to 1, the error pin outputs the high level (toggle) as long as there are no other sources that set the error pin to the low level. This registers is also the trigger for the port safe state function. When Error signal output is cleared, each port state will be changed to a condition according to user's setting. Writing to this register is protected by ECMKCPROT. Refer to **Section 39.3.11, ECMKCPROT — ECM Key Code Protection Register** for the details of key code protection. This register is always read as 00_H.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ECMmECT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 39.18 ECMmECLR Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ECMmECT	Error clear trigger bit 0: Writing 0 is invalid 1: Sets the output level from the error pin to the inactive (high) level.

CAUTIONS

Clearing of the error pin is only possible if all errors, not masked by ECMEMK registers, are cleared beforehand.

Setting or clearing the error output via the ECMmECLR register will generate an error. Therefore, the following has to be set in advance. The sequence below is executed by either CPU.

1. Set the MSKM bit and MSKC bit of the ECMPEM register to "masked".
2. Set the ECMmECT bit in the ECMmECLR register.
3. Wait until $\overline{\text{ERROROUT_M}}$ and $\overline{\text{ERROROUT_C}}$ become high by reading ECMmSSE311 bit of ECMmESSTR9 register 5 times. After that, check that the ECMmSSE311 bit of the ECMmESSTR9 register is "1". If the ECMmSSE311 bit of the ECMmESSTR9 register is not "1", a new error may have occurred.
4. Set the MSKM bit and MSKC bit of the ECMPEM register to "not masked".

Note: This procedure is used when the ECMEOCCFG register is not set.

NOTE

If the Error Pin Low Time counter is still running, the Error Output clear function is masked and Error Output clear requests are ignored. Error Output clear function is executed, after the Error Pin Low Time Counter has expired and the Error Output clear register is written.

The configuration flow of port safe state mode must be set after error signal is cleared by the ECMmECLR register. For the configuration flow of port safe state mode setting, see **Section 2.6.2.3, Configuration Flow**.

39.3.4 ECMmESSTR0 to ECMmESSTR9 — ECM Master/Checker Error Source Status Register 0 to 9 (m = M/C)

The ECM Master/Checker Error Source Status Registers 0 to 9 are read-only registers.

This register represents the status of individual internal error sources, and is unrelated to the setting of the error mask.

ECMmSSEs are provided for the each status of individual internal error sources.

ECMmSSE001-ECMmSSE000 are status bits for counting DCLS error from PE1-PE0. To each DCLS error from each CPU, 2bit expanded status register is prepared for counting the DCLS error. If DCLS error occurs when these bits are “11_B”, these bits keep the values.

Value after reset: 0000 0000_H

ECMmESSTR0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMm SSE 023	ECMm SSE 022	ECMm SSE 021	ECMm SSE 020	ECMm SSE 019	ECMm SSE 018	ECMm SSE 017	ECMm SSE 016	ECMm SSE 015	ECMm SSE 014	ECMm SSE 013	ECMm SSE 012	ECMm SSE 011	ECMm SSE 010	ECMm SSE 009	ECMm SSE 008
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMm SSE 007[1:0]		ECMm SSE 006[1:0]		ECMm SSE 005[1:0]		ECMm SSE 004[1:0]		ECMm SSE 003[1:0]		ECMm SSE 002[1:0]		ECMm SSE 001[1:0]		ECMm SSE 000[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 39.19 ECMmESSTR0 Register Contents

Bit Position	Bit Name	Function
31 to 16	ECMmSSE023 to ECMmSSE008	Error source status bit ECMmSSE023 to ECMmSSE008 correspond to error sources 23 to 8. 0: Error not occurred 1: Error occurred
15 to 0	ECMmSSE007[1:0] to ECMmSSE000[1:0]	Error source status bit ECMmSSE007 to ECMmSSE000 count error sources 7 to 0. 00: Error doesn't happen 01: Error happened once 10: Error happened twice 11: Error happened 3 times or more

ECMmESSTRn (n = 1 to 8, x = (n-1) × 32)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMm SSE [x+55]	ECMm SSE [x+54]	ECMm SSE [x+53]	ECMm SSE [x+52]	ECMm SSE [x+51]	ECMm SSE [x+50]	ECMm SSE [x+49]	ECMm SSE [x+48]	ECMm SSE [x+47]	ECMm SSE [x+46]	ECMm SSE [x+45]	ECMm SSE [x+44]	ECMm SSE [x+43]	ECMm SSE [x+42]	ECMm SSE [x+41]	ECMm SSE [x+40]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMm SSE [x+39]	ECMm SSE [x+38]	ECMm SSE [x+37]	ECMm SSE [x+36]	ECMm SSE [x+35]	ECMm SSE [x+34]	ECMm SSE [x+33]	ECMm SSE [x+32]	ECMm SSE [x+31]	ECMm SSE [x+30]	ECMm SSE [x+29]	ECMm SSE [x+28]	ECMm SSE [x+27]	ECMm SSE [x+26]	ECMm SSE [x+25]	ECMm SSE [x+24]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 39.20 ECMmESSTRn Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMmSSE[x+55] to ECMmSSE[x+24]	Error source status bit ECMmSSE[x+55] to ECMmSSE[x+24] correspond to error sources [x+55] to [x+24]. 0: Error not occurred 1: Error occurred

ECMmESSTR9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMm SSE 311	ECMm SSE 310	ECMm SSE 309	ECMm SSE 308	ECMm SSE 307	ECMm SSE 306	ECMm SSE 305	ECMm SSE 304	ECMm SSE 303	ECMm SSE 302	ECMm SSE 301	ECMm SSE 300	ECMm SSE 299	ECMm SSE 298	ECMm SSE 297	ECMm SSE 296
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMm SSE 295	ECMm SSE 294	ECMm SSE 293	ECMm SSE 292	ECMm SSE 291	ECMm SSE 290	ECMm SSE 289	ECMm SSE 288	ECMm SSE 287	ECMm SSE 286	ECMm SSE 285	ECMm SSE 284	ECMm SSE 283	ECMm SSE 282	ECMm SSE 281	ECMm SSE 280
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 39.21 ECMmESSTR9 Register Contents

Bit Position	Bit Name	Function
31	ECMmSSE311	The status of the $\overline{\text{ERROROUT}}$ pin 0: $\overline{\text{ERROROUT}}$ is low level 1: $\overline{\text{ERROROUT}}$ is high level In ECMMESSTR9 register, $\overline{\text{ERROROUT}}$ pin is $\overline{\text{ERROROUT_M}}$ pin. In ECMCESSTR9 register, $\overline{\text{ERROROUT}}$ pin is $\overline{\text{ERROROUT_C}}$ pin.
30	ECMmSSE310	The status of the ECMmEST writing 0: No Error 1: Error is set by ECMmEST
29	ECMmSSE309	The status of the delay timer overflow 0: No overflow 1: Overflow
28 to 0	ECMmSSE308 to ECMmSSE280	Error source status bit ECMmSSE308 to ECMmSSE280 count error sources 308 to 280. 0: Error not occurred 1: Error occurred

NOTE

Reserved bit

The read value of ECMmSSE bit listed as reserved in **Table 39.10**, **Table 39.11** is undefined. Please mask these values of reserve bit when user confirms ECMmESSTRn.

39.3.5 ECMEPCFG — ECM Error Pulse Configuration Register

The ECM Error Pulse Configuration Register is a read/write register. Writing to this register is protected by ECMKCPROT. Refer to **Section 39.3.11, ECMKCPROT — ECM Key Code Protection Register**, for the details of key code protection.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ECMSL0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 39.22 ECMEPCFG Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	ECMSL0	Error pin operation configuration bit Error output operation setting for the error pin 0: Non-dynamic mode 1: Dynamic mode

In Dynamic mode the timer output OSTM0 determines the output wave of the $\overline{\text{ERROROUT_M}}$ and $\overline{\text{ERROROUT_C}}$ pin in case of no error.

CAUTION

After setting the dynamic mode, it is recommended not to change to non-dynamic mode again, because this can cause a glitch in the error output.

39.3.6 ECMMICFG0 to ECMMICFG9 — ECM Maskable Interrupt Configuration Register 0 to 9

The ECM Maskable Interrupt Configuration Registers 0 to 9 are used to set the generation of the INTECMMI interrupts (EI level interrupts) and the INTECMDCLSMI interrupts (EI level interrupts) when DCLS error occurs. The generation of EI level interrupts in response to errors can be enabled. Writing to these registers are protected by ECMKCPROT. Refer to **Section 39.3.11, ECMKCPROT — ECM Key Code Protection Register**, for the details of key code protection.

Value after reset: 0000 0000_H

ECMMICFG0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMM IE023	ECMM IE022	ECMM IE021	ECMM IE020	ECMM IE019	ECMM IE018	ECMM IE017	ECMM IE016	ECMM IE015	ECMM IE014	ECMM IE013	ECMM IE012	ECMM IE011	ECMM IE010	ECMM IE009	ECMM IE008
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMM IE007[1:0]	ECMM IE006[1:0]	ECMM IE005[1:0]	ECMM IE004[1:0]	ECMM IE003[1:0]	ECMM IE002[1:0]	ECMM IE001[1:0]	ECMM IE000[1:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.23 ECMMICFG0 Register Contents

Bit Position	Bit Name	Function
31 to 16	ECMMIE023 to ECMMIE008	ECM maskable interrupt generation control bit ECMMIE023 to ECMMIE008 correspond to error sources 23 to 8. 0: Interrupt generation disabled 1: Interrupt generation enabled
15 to 0	ECMMIE007[1:0] to ECMMIE000[1:0]	ECM maskable interrupt generation control bit ECMMIE007 to ECMMIE000 correspond error sources 7 to 0. 00 : Interrupt generation disabled 01 : Interrupt generation enabled when one error counted Interrupt generation enabled when two errors counted Interrupt generation enabled when three errors counted 10*1: Interrupt generation disabled when one error counted Interrupt generation enabled when two errors counted Interrupt generation enabled when three errors counted 11*1: Interrupt generation disabled when one error counted Interrupt generation disabled when two errors counted Interrupt generation enabled when three errors counted

Note 1. Interrupt is not generated when four or more errors counted.

ECMMICFGn (n = 1 to 8, x = (n-1) × 32)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM MIE [x+55]	ECM MIE [x+54]	ECM MIE [x+53]	ECM MIE [x+52]	ECM MIE [x+51]	ECM MIE [x+50]	ECM MIE [x+49]	ECM MIE [x+48]	ECM MIE [x+47]	ECM MIE [x+46]	ECM MIE [x+45]	ECM MIE [x+44]	ECM MIE [x+43]	ECM MIE [x+42]	ECM MIE [x+41]	ECM MIE [x+40]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM MIE [x+39]	ECM MIE [x+38]	ECM MIE [x+37]	ECM MIE [x+36]	ECM MIE [x+35]	ECM MIE [x+34]	ECM MIE [x+33]	ECM MIE [x+32]	ECM MIE [x+31]	ECM MIE [x+30]	ECM MIE [x+29]	ECM MIE [x+28]	ECM MIE [x+27]	ECM MIE [x+26]	ECM MIE [x+25]	ECM MIE [x+24]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.24 ECMMICFGn Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMMIE[x+55] to ECMMIE[x+24]	ECM maskable interrupt generation control bit ECMMIE[x+55] to ECMMIE[x+24] correspond to error sources [x+55] to [x+24]. 0: Interrupt generation disabled 1: Interrupt generation enabled

ECMMICFG9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	ECMM IE308	ECMM IE307	—	ECMM IE305	ECMM IE304	ECMM IE303	ECMM IE302	ECMM IE301	ECMM IE300	ECMM IE299	ECMM IE298	ECMM IE297	ECMM IE296
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMM IE295	ECMM IE294	ECMM IE293	ECMM IE292	ECMM IE291	ECMM IE290	ECMM IE289	ECMM IE288	ECMM IE287	ECMM IE286	ECMM IE285	ECMM IE284	ECMM IE283	ECMM IE282	ECMM IE281	ECMM IE280
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.25 ECMMICFG9 Register Contents

Bit Position	Bit Name	Function
31 to 29	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
28 to 27	ECMMIE308 to ECMMIE307	ECM maskable interrupt generation control bit ECMMIE308 to ECMMIE307 correspond to error sources 308 to 307. 0: Interrupt generation disabled 1: Interrupt generation enabled
26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25 to 0	ECMMIE305 to ECMMIE280	ECM maskable interrupt generation control bit ECMMIE305 to ECMMIE280 correspond to error sources 305 to 280. 0: Interrupt generation disabled 1: Interrupt generation enabled

NOTE

Reserved bit

The value of ECMMIE bit listed as reserved for the given error input numbers in **Table 39.10**, **Table 39.11**. When read, the value after reset is returned. When writing, write the value after reset.

39.3.7 ECMNMICFG0 to ECMNMICFG9 — ECM FE Level Interrupt Configuration Register 0 to 9

The ECM FE Level Interrupt Configuration Registers 0 to 9 are used to set the generation of INTECMNMI interrupts (FE level interrupt). Writing to these registers are protected by ECMKCROT. Refer to **Section 39.3.11, ECMKCROT — ECM Key Code Protection Register**, for the details of key code protection.

Value after reset: 0000 0000_H

ECMNMICFG0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM NMI E023	ECM NMI E022	ECM NMI E021	ECM NMI E020	ECM NMI E019	ECM NMI E018	ECM NMI E017	ECM NMI E016	ECM NMI E015	ECM NMI E014	ECM NMI E013	ECM NMI E012	ECM NMI E011	ECM NMI E010	ECM NMI E009	ECM NMI E008
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM NMI E007[1:0]	ECM NMI E006[1:0]	ECM NMI E005[1:0]	ECM NMI E004[1:0]	ECM NMI E003[1:0]	ECM NMI E002[1:0]	ECM NMI E001[1:0]	ECM NMI E000[1:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.26 ECMNMICFG0 Register Contents

Bit Position	Bit Name	Function
31 to 16	ECMNMIE023 to ECMNMIE008	ECM FE level interrupt generation control bit ECMNMIE023 to ECMNMIE008 correspond to error sources 23 to 8. 0: Interrupt generation disabled 1: Interrupt generation enabled
15 to 0	ECMNMIE007[1:0] to ECMNMIE000[1:0]	ECM FE level interrupt generation control bit ECMNMIE007 to ECMNMIE000 correspond error source 7 to 0. 00 : Interrupt generation disabled 01 : Interrupt generation enabled when one error counted Interrupt generation enabled when two errors counted Interrupt generation enabled when three errors counted 10*1: Interrupt generation disabled when one error counted Interrupt generation enabled when two errors counted Interrupt generation enabled when three errors counted 11*1: Interrupt generation disabled when one error counted Interrupt generation disabled when two errors counted Interrupt generation enabled when three errors counted

Note 1. Interrupt is not generated when four or more errors counted.

ECMNMICFGn (n = 1 to 8, x = (n-1) × 32)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM NMIE [x+55]	ECM NMIE [x+54]	ECM NMIE [x+53]	ECM NMIE [x+52]	ECM NMIE [x+51]	ECM NMIE [x+50]	ECM NMIE [x+49]	ECM NMIE [x+48]	ECM NMIE [x+47]	ECM NMIE [x+46]	ECM NMIE [x+45]	ECM NMIE [x+44]	ECM NMIE [x+43]	ECM NMIE [x+42]	ECM NMIE [x+41]	ECM NMIE [x+40]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM NMIE [x+39]	ECM NMIE [x+38]	ECM NMIE [x+37]	ECM NMIE [x+36]	ECM NMIE [x+35]	ECM NMIE [x+34]	ECM NMIE [x+33]	ECM NMIE [x+32]	ECM NMIE [x+31]	ECM NMIE [x+30]	ECM NMIE [x+29]	ECM NMIE [x+28]	ECM NMIE [x+27]	ECM NMIE [x+26]	ECM NMIE [x+25]	ECM NMIE [x+24]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.27 ECMNMICFGn Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMNMIE[x+55] to ECMNMIE[x+24]	ECM FE level interrupt generation control bit ECMNMIE[x+55] to ECMNMIE[x+24] correspond to error sources [x+55] to [x+24]. 0: Interrupt generation disabled 1: Interrupt generation enabled

ECMNMICFG9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	ECM NMI E308	ECM NMI E307	—	ECM NMI E305	ECM NMI E304	ECM NMI E303	ECM NMI E302	ECM NMI E301	ECM NMI E300	ECM NMI E299	ECM NMI E298	ECM NMI E297	ECM NMI E296
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM NMI E295	ECM NMI E294	ECM NMI E293	ECM NMI E292	ECM NMI E291	ECM NMI E290	ECM NMI E289	ECM NMI E288	ECM NMI E287	ECM NMI E286	ECM NMI E285	ECM NMI E284	ECM NMI E283	ECM NMI E282	ECM NMI E281	ECM NMI E280
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.28 ECMNMICFG9 Register Contents

Bit Position	Bit Name	Function
31 to 29	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
28 to 27	ECMNMIE308 to ECMNMIE307	ECM FE level interrupt generation control bit ECMNMIE308 to ECMNMIE307 correspond to error sources 308 to 307. 0: Interrupt generation disabled 1: Interrupt generation enabled
26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25 to 0	ECMNMIE305 to ECMNMIE280	ECM FE level interrupt generation control bit ECMNMIE305 to ECMNMIE280 correspond to error sources 305 to 280. 0: Interrupt generation disabled 1: Interrupt generation enabled

NOTE

Reserved bit

The value of ECMNMIE bit listed as reserved for the given error input numbers in **Table 39.10**, **Table 39.11**. When read, the value after reset is returned. When writing, write the value after reset.

39.3.8 ECMIRCFG0 to ECMIRCFG9 — ECM Internal Reset Configuration Register 0 to 9

The ECM Internal Reset Configuration Registers 0 to 9 are used to set the generation of Error Control Module Reset in response to internal errors. Writing to this register is protected by ECMKCPROT. Refer to **Section 39.3.11, ECMKCPROT — ECM Key Code Protection Register**, for the details of key code protection.

Value after reset: 0001 0000_H (ECMIRCFG0), 0000 0000_H (ECMIRCFG1 to ECMIRCFG9)

ECMIRCFG0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM IRE023	ECM IRE022	ECM IRE021	ECM IRE020	ECM IRE019	ECM IRE018	ECM IRE017	ECM IRE016	ECM IRE015	ECM IRE014	ECM IRE013	ECM IRE012	ECM IRE011	ECM IRE010	ECM IRE009	ECM IRE008
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM IRE007[1:0]	ECM IRE006[1:0]	ECM IRE005[1:0]	ECM IRE004[1:0]	ECM IRE003[1:0]	ECM IRE002[1:0]	ECM IRE001[1:0]	ECM IRE000[1:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.29 ECMIRCFG0 Register Contents

Bit Position	Bit Name	Function
31 to 16	ECMIRE023 to ECMIRE008	ECM internal reset generation control bit ECMIRE023 to ECMIRE008 correspond to error sources 23 to 8. 0: ECM internal reset generation disabled 1: ECM internal reset generation enabled
15 to 0	ECMIRE007[1:0] to ECMIRE000[1:0]	ECM internal reset generation control bit ECMIRE007 to ECMIRE000 correspond error sources 7 to 0. 00 : ECM internal reset disabled 01 : ECM internal reset enabled when one error counted ECM internal reset enabled when two errors counted ECM internal reset enabled when three errors counted 10*: ECM internal reset disabled when one error counted ECM internal reset enabled when two errors counted ECM internal reset enabled when three errors counted 11*: ECM internal reset disabled when one error counted ECM internal reset disabled when two errors counted ECM internal reset enabled when three errors counted

Note 1. ECM internal reset is not generated when four or more errors counted.

ECMIRCFGn (n = 1 to 8, x = (n-1) × 32)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM IRE [x+55]	ECM IRE [x+54]	ECM IRE [x+53]	ECM IRE [x+52]	ECM IRE [x+51]	ECM IRE [x+50]	ECM IRE [x+49]	ECM IRE [x+48]	ECM IRE [x+47]	ECM IRE [x+46]	ECM IRE [x+45]	ECM IRE [x+44]	ECM IRE [x+43]	ECM IRE [x+42]	ECM IRE [x+41]	ECM IRE [x+40]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM IRE [x+39]	ECM IRE [x+38]	ECM IRE [x+37]	ECM IRE [x+36]	ECM IRE [x+35]	ECM IRE [x+34]	ECM IRE [x+33]	ECM IRE [x+32]	ECM IRE [x+31]	ECM IRE [x+30]	ECM IRE [x+29]	ECM IRE [x+28]	ECM IRE [x+27]	ECM IRE [x+26]	ECM IRE [x+25]	ECM IRE [x+24]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.30 ECMIRCFGn Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMIRE[x+55] to ECMIRE[x+24]	ECM internal reset generation control bit ECMIRE[x+55] to ECMIRE[x+24] correspond to error sources [x+55] to [x+24]. 0: ECM internal reset generation disabled 1: ECM internal reset generation enabled

ECMIRCFG9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	ECM IRE309	ECM IRE308	ECM IRE307	—	ECM IRE305	ECM IRE304	ECM IRE303	ECM IRE302	ECM IRE301	ECM IRE300	ECM IRE299	ECM IRE298	ECM IRE297	ECM IRE296
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM IRE295	ECM IRE294	ECM IRE293	ECM IRE292	ECM IRE291	ECM IRE290	ECM IRE289	ECM IRE288	ECM IRE287	ECM IRE286	ECM IRE285	ECM IRE284	ECM IRE283	ECM IRE282	ECM IRE281	ECM IRE280
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.31 ECMIRCFG9 Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
29	ECMIRE309	ECM internal reset generation control bit ECMIRE309 corresponds to delay timer overflow. 0: ECM internal reset generation disabled 1: ECM internal reset generation enabled
28 to 27	ECMIRE308 to ECMIRE307	ECM internal reset generation control bit ECMIRE308 to ECMIRE307 correspond to error sources 308 to 307. 0: ECM internal reset generation disabled 1: ECM internal reset generation enabled
26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25 to 0	ECMIRE305 to ECMIRE280	ECM internal reset generation control bit ECMIRE305 to ECMIRE280 correspond to error sources 305 to 280. 0: ECM internal reset generation disabled 1: ECM internal reset generation enabled

NOTE**Reserved bit**

The value of ECMIRE bit listed as reserved for the given error input numbers in **Table 39.10**, **Table 39.11**. When read, the value after reset is returned. When writing, write the value after reset.

39.3.9 ECMEMK0 to ECMEMK9 — ECM Error Mask Register 0 to 9

The ECM Error Mask Registers 0 to 9 are used to mask the individual error sources of the error pin output. This registers is also the trigger for the port safe state function. When Error signal output is not masked, each port state will be changed to a condition according to user's setting. Writing to this register is protected by ECMKCPROT. Refer to **Section 39.3.11, ECMKCPROT — ECM Key Code Protection Register**, for the details of key code protection.

Value after reset: 0000 0000_H

ECMEMK0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM EMK 023	ECM EMK 022	ECM EMK 021	ECM EMK 020	ECM EMK 019	ECM EMK 018	ECM EMK 017	ECM EMK 016	ECM EMK 015	ECM EMK 014	ECM EMK 013	ECM EMK 012	ECM EMK 011	ECM EMK 010	ECM EMK 009	ECM EMK 008
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM EMK 007[1:0]	ECM EMK 006[1:0]	ECM EMK 005[1:0]	ECM EMK 004[1:0]	ECM EMK 003[1:0]	ECM EMK 002[1:0]	ECM EMK 001[1:0]	ECM EMK 000[1:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.32 ECMEMK0 Register Contents

Bit Position	Bit Name	Function
31 to 16	ECMEMK023 to ECMEMK008	ECM error output signal mask control bit ECMEMK023 to ECMEMK008 correspond to error sources 23 to 8. 0: Error signal output is not masked 1: Error signal output is masked
15 to 0	ECMEMK007[1:0] to ECMEMK000[1:0]	ECM error output signal mask control bit ECMEMK007 to ECMEMK000 correspond error sources 7 to 0 00: Error output signal is not masked. 01: Error output signal is masked when one error counted Error output signal is not masked when two errors counted Error output signal is not masked when three errors counted 10: Error output signal is masked when one error counted Error output signal is masked when two errors counted Error output signal is not masked when three errors counted 11: Error output signal is masked

ECMEMKn (n = 1 to 8, x = (n-1) × 32)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM EMK [x+55]	ECM EMK [x+54]	ECM EMK [x+53]	ECM EMK [x+52]	ECM EMK [x+51]	ECM EMK [x+50]	ECM EMK [x+49]	ECM EMK [x+48]	ECM EMK [x+47]	ECM EMK [x+46]	ECM EMK [x+45]	ECM EMK [x+44]	ECM EMK [x+43]	ECM EMK [x+42]	ECM EMK [x+41]	ECM EMK [x+40]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM EMK [x+39]	ECM EMK [x+38]	ECM EMK [x+37]	ECM EMK [x+36]	ECM EMK [x+35]	ECM EMK [x+34]	ECM EMK [x+33]	ECM EMK [x+32]	ECM EMK [x+31]	ECM EMK [x+30]	ECM EMK [x+29]	ECM EMK [x+28]	ECM EMK [x+27]	ECM EMK [x+26]	ECM EMK [x+25]	ECM EMK [x+24]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.33 ECMEMKn Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMEMK[x+55] to ECMEMK[x+24]	ECM error output signal mask control bit ECMEMK[x+55] to ECMEMK[x+24] correspond to error sources [x+55] to [x+24]. 0: Error signal output is not masked 1: Error signal output is masked

ECMEMK9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	ECM EMK 309	ECM EMK 308	ECM EMK 307	—	ECM EMK 305	ECM EMK 304	ECM EMK 303	ECM EMK 302	ECM EMK 301	ECM EMK 300	ECM EMK 299	ECM EMK 298	ECM EMK 297	ECM EMK 296
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM EMK 295	ECM EMK 294	ECM EMK 293	ECM EMK 292	ECM EMK 291	ECM EMK 290	ECM EMK 289	ECM EMK 288	ECM EMK 287	ECM EMK 286	ECM EMK 285	ECM EMK 284	ECM EMK 283	ECM EMK 282	ECM EMK 281	ECM EMK 280
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.34 ECMEMK9 Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved When read, the value after reset is returned. When writing, write the value to “1”.
29	ECMEMK309	ECM error output signal mask control bit ECMEMK309 corresponds to delay timer overflow. 0: Error signal output is not masked 1: Error signal output is masked
28 to 27	ECMEMK308 to ECMEMK307	ECM error output signal mask control bit ECMEMK308 to ECMEMK307 correspond to error sources 308 to 307. 0: Error signal output is not masked 1: Error signal output is masked
26	—	Reserved When read, the value after reset is returned. When writing, write the value to “1”.
25 to 0	ECMEMK305 to ECMEMK280	ECM error output signal mask control bit ECMEMK305 to ECMEMK280 correspond to error sources 305 to 280. 0: Error signal output is not masked 1: Error signal output is masked

NOTES

1. Error mask specification
If an error flag is set but masked, clearing the mask will set the $\overline{\text{ERROROUT_M}}$, $\overline{\text{ERROROUT_C}}$ and $\overline{\text{ERROROUTZ}}$ to active low level. This happens independently from the time of the error occurrence. In other words, the flag is evaluated statically.
2. Reserved bit
Please set the value of ECMEMK bit listed as reserved in **Table 39.10**, **Table 39.11** to “1”.

39.3.10 ECMESSTC0 to ECMESSTC9 — ECM Error Source Status Clear Trigger Register 0 to 9

The ECM Error Source Status Clear Trigger Registers 0 to 9 are a write-only register and can be written in 32-bit units. These registers are used to clear the individual error source status of the ECM master/checker error source status registers 0 to 9. Both the error status of the ECM master and the ECM checker are cleared simultaneously.

Writing to this register is protected by ECMKCPROT. Refer to **Section 39.3.11, ECMKCPROT — ECM Key Code Protection Register.**

Value after reset: 0000 0000_H

ECMESSTC0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM CLSSE 023	ECM CLSSE 022	ECM CLSSE 021	ECM CLSSE 020	ECM CLSSE 019	ECM CLSSE 018	ECM CLSSE 017	ECM CLSSE 016	ECM CLSSE 015	ECM CLSSE 014	ECM CLSSE 013	ECM CLSSE 012	ECM CLSSE 011	ECM CLSSE 010	ECM CLSSE 009	ECM CLSSE 008
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM CLSSE 007[1:0]	ECM CLSSE 006[1:0]	ECM CLSSE 005[1:0]	ECM CLSSE 004[1:0]	ECM CLSSE 003[1:0]	ECM CLSSE 002[1:0]	ECM CLSSE 001[1:0]	ECM CLSSE 000[1:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 39.35 ECMESSTC0 Register Contents

Bit Position	Bit Name	Function
31 to 16	ECMCLSSE023 to ECMCLSSE008	ECM error status clear bit ECMCLSSE023 to ECMCLSSE008 correspond to ECMmSSE023 to ECMmSSE008. 0: Corresponding error status unchanged 1: Corresponding error status cleared
15 to 0	ECMCLSSE007[1:0] to ECMCLSSE000[1:0]	ECM error status clear bit ECMCLSSE007 to ECMCLSSE000 correspond to ECMmSSE007 to ECMmSSE000. 00: Corresponding error status unchanged 01: (Same with 11 _B setting) Error status cleared 10: (Same with 11 _B setting) Error status cleared 11: Corresponding error status cleared

ECMESSTCn (n = 1 to 8, x = (n-1) × 32)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM CLSSE [x+55]	ECM CLSSE [x+54]	ECM CLSSE [x+53]	ECM CLSSE [x+52]	ECM CLSSE [x+51]	ECM CLSSE [x+50]	ECM CLSSE [x+49]	ECM CLSSE [x+48]	ECM CLSSE [x+47]	ECM CLSSE [x+46]	ECM CLSSE [x+45]	ECM CLSSE [x+44]	ECM CLSSE [x+43]	ECM CLSSE [x+42]	ECM CLSSE [x+41]	ECM CLSSE [x+40]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM CLSSE [x+39]	ECM CLSSE [x+38]	ECM CLSSE [x+37]	ECM CLSSE [x+36]	ECM CLSSE [x+35]	ECM CLSSE [x+34]	ECM CLSSE [x+33]	ECM CLSSE [x+32]	ECM CLSSE [x+31]	ECM CLSSE [x+30]	ECM CLSSE [x+29]	ECM CLSSE [x+28]	ECM CLSSE [x+27]	ECM CLSSE [x+26]	ECM CLSSE [x+25]	ECM CLSSE [x+24]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 39.36 ECMESSTCn Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMCLSSE[x+55] to ECMCLSSE[x+24]	ECM error status clear bit ECMCLSSE[x+55] to ECMCLSSE[x+24] correspond to ECMmSSE[x+55] to ECMmSSE[x+24]. 0: Corresponding error status unchanged 1: Corresponding error status cleared

ECMESSTC9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ECM CLSSE 310	ECM CLSSE 309	ECM CLSSE 308	ECM CLSSE 307	ECM CLSSE 306	ECM CLSSE 305	ECM CLSSE 304	ECM CLSSE 303	ECM CLSSE 302	ECM CLSSE 301	ECM CLSSE 300	ECM CLSSE 299	ECM CLSSE 298	ECM CLSSE 297	ECM CLSSE 296
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM CLSSE 295	ECM CLSSE 294	ECM CLSSE 293	ECM CLSSE 292	ECM CLSSE 291	ECM CLSSE 290	ECM CLSSE 289	ECM CLSSE 288	ECM CLSSE 287	ECM CLSSE 286	ECM CLSSE 285	ECM CLSSE 284	ECM CLSSE 283	ECM CLSSE 282	ECM CLSSE 281	ECM CLSSE 280
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 39.37 ECMESSTC9 Register Contents

Bit Position	Bit Name	Function
31	—	Reserved When writing, write the value after reset.
30	ECMCLSSE310	ECM error status clear bit ECMCLSSE310 corresponds to ECMmSSE310. 0: Corresponding error status unchanged 1: Corresponding error status cleared
29	ECMCLSSE309	ECM error status clear bit ECMCLSSE309 corresponds to ECMmSSE309. 0: Corresponding error status unchanged 1: Corresponding error status cleared
28 to 0	ECMCLSSE308 to ECMCLSSE280	ECM error status clear bit ECMCLSSE308 to ECMCLSSE280 correspond to ECMmSSE308 to ECMmSSE280. 0: Corresponding error status unchanged 1: Corresponding error status cleared

NOTE

Reserved bit

The value of ECMCLSSE bit listed as reserved for the given error input numbers in **Table 39.10**, **Table 39.11**. When writing, write the value after reset.

39.3.11 ECMKCPROT — ECM Key Code Protection Register

The ECM Key Code Protection Register is used for protection against writing operation to the configuration registers.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 39.38 ECMKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500_H to this register to disable writing protected registers.
Write A5A5A501_H to this register to enable writing protected registers.

39.3.12 ECMPE0 to ECMPE9 — ECM Pseudo Error Trigger Register 0 to 9

The ECM Pseudo Error Trigger Registers 0 to 9 are write-only registers. These registers are used to generate a pseudo error for test purposes. The ECM operation in response to the generation of a pseudo error is identical to that of a real error source. Writing to this register is protected by ECMKCPROT. Refer to **Section 39.3.11, ECMKCPROT — ECM Key Code Protection Register**, for the details of key code protection.

Value after reset: 0000 0000_H

ECMPE0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM PE023	ECM PE022	ECM PE021	ECM PE020	ECM PE019	ECM PE018	ECM PE017	ECM PE016	ECM PE015	ECM PE014	ECM PE013	ECM PE012	ECM PE011	ECM PE010	ECM PE009	ECM PE008
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM PE007[1:0]	ECM PE006[1:0]	ECM PE005[1:0]	ECM PE004[1:0]	ECM PE003[1:0]	ECM PE002[1:0]	ECM PE001[1:0]	ECM PE000[1:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 39.39 ECMPE0 Register Contents

Bit Position	Bit Name	Function
31 to 16	ECMPE023 to ECMPE008	ECM pseudo error trigger bit ECMPE023 to ECMPE008 correspond to error sources 23 to 8. 0: Pseudo error is not generated. 1: Pseudo error is generated.
15 to 0	ECMPE007[1:0] to ECMPE000[1:0]	ECM pseudo error trigger bit ECMPE007 to ECMPE000 correspond to error sources 7 to 0. 00: Pseudo error is not generated 01: Pseudo error is generated as same as error counting once 10: Pseudo error is generated as same as error counting once 11: Pseudo error is generated as same as error counting once

ECMPEn (n = 1 to 8, (n-1) × 32)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM PE [x+55]	ECM PE [x+54]	ECM PE [x+53]	ECM PE [x+52]	ECM PE [x+51]	ECM PE [x+50]	ECM PE [x+49]	ECM PE [x+48]	ECM PE [x+47]	ECM PE [x+46]	ECM PE [x+45]	ECM PE [x+44]	ECM PE [x+43]	ECM PE [x+42]	ECM PE [x+41]	ECM PE [x+40]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM PE [x+39]	ECM PE [x+38]	ECM PE [x+37]	ECM PE [x+36]	ECM PE [x+35]	ECM PE [x+34]	ECM PE [x+33]	ECM PE [x+32]	ECM PE [x+31]	ECM PE [x+30]	ECM PE [x+29]	ECM PE [x+28]	ECM PE [x+27]	ECM PE [x+26]	ECM PE [x+25]	ECM PE [x+24]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 39.40 ECMPEn Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMPE[x+55] to ECMPE[x+24]	ECM pseudo error trigger bit ECMPE[x+55] to ECMPE[x+24] correspond to error sources [x+55] to [x+24]. 0: Pseudo error is not generated. 1: Pseudo error is generated.

ECMPE9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	ECM PE309	ECM PE308	—	—	ECM PE305	ECM PE304	ECM PE303	ECM PE302	ECM PE301	ECM PE300	ECM PE299	ECM PE298	ECM PE297	ECM PE296
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	W	W	R	R	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM PE295	ECM PE294	ECM PE293	ECM PE292	ECM PE291	ECM PE290	ECM PE289	ECM PE288	ECM PE287	ECM PE286	ECM PE285	ECM PE284	ECM PE283	ECM PE282	ECM PE281	ECM PE280
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 39.41 ECMPE9 Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved When writing, write the value after reset.
29	ECMPE309	ECM pseudo error trigger bit ECMPE309 corresponds to delay timer overflow. 0: Pseudo error is not generated. 1: Pseudo error is generated.
28	ECMPE308	ECM pseudo error trigger bit ECMPE308 corresponds to error source 308. 0: Pseudo error is not generated. 1: Pseudo error is generated.
27, 26	—	Reserved When writing, write the value after reset.
25 to 0	ECMPE305 to ECMPE280	ECM pseudo error trigger bit ECMPE305 to ECMPE280 correspond to error sources 305 to 280. 0: Pseudo error is not generated. 1: Pseudo error is generated.

NOTE**Reserved bit**

The value of ECMPE bit listed as reserved for the given error input numbers in **Table 39.10**, **Table 39.11**. When writing, write the value after reset.

39.3.13 ECMDTMCTL — ECM Delay Timer Control Register

The ECM Delay Timer Control Register is a read/write register. This register is used to control the delay timer. Writing to this register is protected by ECMKCPROT. Refer to **Section 39.3.11, ECMKCPROT — ECM Key Code Protection Register**, for the details of key code protection.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	DTMSTACNTCLK	—	—	DTMSTP	DTMSTA
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	R/W

Table 39.42 ECMDTMCTL Register Contents

Bit Position	Bit Name	Function
7 to 5	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
4	DTMSTACNTCLK	Delay timer start confirmation status. 0: Delay timer does not start 1: Delay timer starts.
3, 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	DTMSTP	Delay timer stop bit By writing “1” to this bit, delay timer is stopped (writing 0 is ignored). Simultaneously, DTMSTA bit will be 0. 0: Delay timer is completed or stop request is not in progress. 1: Stop request for delay timer is in progress.
0	DTMSTA	Delay timer start bit Specifies the operation of the delay timer at the occurrence of an error event. 0: Delay timer does not start 1: Delay timer starts

NOTES

- ECMDTMCTL register can be accessed via P-Bus but delay timer runs with not P-Bus clock but dedicated counter clock.
Therefore, time lag exists between writing of ECMDTMCTL and enabling of delay timer.
DTMSTACNTCLK can be used to confirm whether delay timer is enabled or not. Reconfirm that DTMSTA has been updated by checking DTMSTACNTCLK after writing to DTMSTA.
- ECMDTMCTL register can be written only when (DTMSTA, DTMSTACNTCLK) = (0, 0) or (1, 1).
Confirm the values of DTMSTA and DTMSTACNTCLK before writing to ECMDTMCTL.
- The delay timer needs 3 clocks of cntclk until it stops an overflow certainly.
Please consider to set the sufficient value to ECMDTMCMP[15:0].

39.3.14 ECMDTMR — ECM Delay Timer Register

The ECM Delay Timer Register is a read-only register. The ECM delay timer register is initialized by setting the ECMSTA bit of the ECM delay timer control register from 1 (timer in operation) to 0 (timer stopped).

Value after reset: 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMDTMR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 39.43 ECMDTMR Register Contents

Bit Position	Bit Name	Function
15 to 0	ECMDTMR[15:0]	Delay timer counter value.

39.3.15 ECMDTMCMP — ECM Delay Timer Compare Register

The ECM Delay Timer Compare Register is a read/write register. The ECMmESSTR9.ECMmSSE309 bit is set when the value of this register matches with the value of the ECM delay timer register. Writing data to this register has to be conducted while the delay timer is stopped. Writing to this register is protected by ECMKCPROT. Refer to **Section 39.3.11, ECMKCPROT — ECM Key Code Protection Register**, for the details of key code protection.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CMPW
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMDTMCMP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.44 ECMDTMCMP Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	CMPW	Indicates whether a write to the counter clock region of the ECMDTMCMP register setting is in progress 0: Not in progress 1: In progress. While the bit is set, further setting to the ECMDTMCMP is prohibited.
15 to 0	ECMDTMCMP[15:0]	Delay timer compare value

NOTES

1. ECMDTMCMP register is set via P-bus, however actual delay timer exists in a different clock domain. When ECMDTMCMP is configured, the value is copied across clock domains to reflect to the delay timer, which takes a certain time. CMPW indicates whether reflecting the new ECMDTMCMP value across clock domains is in progress.
2. While CMPW is “1”, further setting to ECMDTMCMP is ignored.
Please confirm CMPW = 0, before writing of ECMDTMCMP.

39.3.16 ECMMIDTMCFG0 to ECMMIDTMCFG9 — ECM Maskable Interrupt Delay Timer Configuration Register 0 to 9

The ECM Maskable Interrupt Delay Timer Configuration Registers 0 to 9 are used to enable/disable the delay timer start caused by EI level interrupts in response to errors. Writing to this register is protected by ECMKCROT. Refer to **Section 39.3.11, ECMKCROT — ECM Key Code Protection Register**, for the details of key code protection.

Value after reset: 0000 0000_H

ECMMIDTMCFG0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM MITE 023	ECM MITE 022	ECM MITE 021	ECM MITE 020	ECM MITE 019	ECM MITE 018	ECM MITE 017	ECM MITE 016	ECM MITE 015	ECM MITE 014	ECM MITE 013	ECM MITE 012	ECM MITE 011	ECM MITE 010	ECM MITE 009	ECM MITE 008
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM MITE 007[1:0]	ECM MITE 006[1:0]	ECM MITE 005[1:0]	ECM MITE 004[1:0]	ECM MITE 003[1:0]	ECM MITE 002[1:0]	ECM MITE 001[1:0]	ECM MITE 000[1:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.45 ECMMIDTMCFG0 Register Contents

Bit Position	Bit Name	Function
31 to 16	ECMMITE023 to ECMMITE008	ECM delay timer start control bit ECMMITE023 to ECMMITE008 correspond to maskable interrupts generated by error sources 23 to 8. 0: Delay timer start disabled 1: Delay timer start enabled
15 to 0	ECMMITE007[1:0] to ECMMITE000[1:0]*1	ECM delay timer start control bit ECMMITE007 to ECMMITE000 correspond to error sources 7 to 0. 00 : Delay timer start disabled 01 : Delay timer start enabled 10*2: Delay timer start disabled when one error counted Delay timer start enabled when two errors counted Delay timer start enabled when three errors counted 11*2: Delay timer start disabled when one error counted Delay timer start disabled when two errors counted Delay timer start enabled when three errors counted

Note 1. It is necessary that ECMMITE007[1:0] to ECMMITE000[1:0] bit corresponding to ECMMIE007[1:0] to ECMMIE000[1:0] bit is set to a value same as ECMMIE007[1:0] to ECMMIE000[1:0] when ECM delay timer is enabled.

Note 2. Delay timer is not started when four or more errors counted.

ECMMIDTMCFGn (n = 1 to 8, x = (n-1) × 32)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM MITE [x+55]	ECM MITE [x+54]	ECM MITE [x+53]	ECM MITE [x+52]	ECM MITE [x+51]	ECM MITE [x+50]	ECM MITE [x+49]	ECM MITE [x+48]	ECM MITE [x+47]	ECM MITE [x+46]	ECM MITE [x+45]	ECM MITE [x+44]	ECM MITE [x+43]	ECM MITE [x+42]	ECM MITE [x+41]	ECM MITE [x+40]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM MITE [x+39]	ECM MITE [x+38]	ECM MITE [x+37]	ECM MITE [x+36]	ECM MITE [x+35]	ECM MITE [x+34]	ECM MITE [x+33]	ECM MITE [x+32]	ECM MITE [x+31]	ECM MITE [x+30]	ECM MITE [x+29]	ECM MITE [x+28]	ECM MITE [x+27]	ECM MITE [x+26]	ECM MITE [x+25]	ECM MITE [x+24]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.46 ECMMIDTMCFGn Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMMITE[x+55] to ECMMITE[x+24]	ECM delay timer start control bit ECMMITE[x+55] to ECMMITE[x+24] correspond to maskable interrupts generated by error sources [x+55] to [x+24]. 0: Delay timer start disabled 1: Delay timer start enabled

ECMMIDTMCFG9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	ECM MITE 308	ECM MITE 307	—	ECM MITE 305	ECM MITE 304	ECM MITE 303	ECM MITE 302	ECM MITE 301	ECM MITE 300	ECM MITE 299	ECM MITE 298	ECM MITE 297	ECM MITE 296
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM MITE 295	ECM MITE 294	ECM MITE 293	ECM MITE 292	ECM MITE 291	ECM MITE 290	ECM MITE 289	ECM MITE 288	ECM MITE 287	ECM MITE 286	ECM MITE 285	ECM MITE 284	ECM MITE 283	ECM MITE 282	ECM MITE 281	ECM MITE 280
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.47 ECMMIDTMCFG9 Register Contents

Bit Position	Bit Name	Function
31 to 29	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
28 to 27	ECMMITE308 to ECMMITE307	ECM delay timer start control bit ECMMITE308 to ECMMITE307 correspond to maskable interrupts generated by error sources 308 to 307. 0: Delay timer start disabled 1: Delay timer start enabled
26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25 to 0	ECMMITE305 to ECMMITE280	ECM delay timer start control bit ECMMITE305 to ECMMITE280 correspond to maskable interrupts generated by error sources 305 to 280. 0: Delay timer start disabled 1: Delay timer start enabled

NOTE**Reserved bit**

The value of ECMMITE bit listed as reserved for the given error input numbers in **Table 39.10**, **Table 39.11**. When read, the value after reset is returned. When writing, write the value after reset.

39.3.17 ECMNMIDTMCFG0 to ECMNMIDTMCFG9 — ECM FE Level Interrupt Delay Timer Configuration Register 0 to 9

The ECM FE Level Interrupt Delay Timer Registers 0 to 9 are used to set enable/disable of the delay timer start caused by FE level interrupts in response to errors. Writing to this register is protected by ECMKCPROT. Refer to **Section 39.3.11, ECMKCPROT — ECM Key Code Protection Register**, for the details of key code protection.

Value after reset: 0000 0000_H

ECMNMIDTMCFG0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM NMITE 023	ECM NMITE 022	ECM NMITE 021	ECM NMITE 020	ECM NMITE 019	ECM NMITE 018	ECM NMITE 017	ECM NMITE 016	ECM NMITE 015	ECM NMITE 014	ECM NMITE 013	ECM NMITE 012	ECM NMITE 011	ECM NMITE 010	ECM NMITE 009	ECM NMITE 008
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM NMITE 007[1:0]	ECM NMITE 006[1:0]	ECM NMITE 005[1:0]	ECM NMITE 004[1:0]	ECM NMITE 003[1:0]	ECM NMITE 002[1:0]	ECM NMITE 001[1:0]	ECM NMITE 000[1:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.48 ECMNMIDTMCFG0 Register Contents

Bit Position	Bit Name	Function
31 to 16	ECMNMITE023 to ECMNMITE008	ECM delay timer start control bit ECMNMITE023 to ECMNMITE008 correspond to FE level interrupts generated by error sources 23 to 8. 0: Delay timer start disabled 1: Delay timer start enabled
15 to 0	ECMNMITE007[1:0] to ECMNMITE000[1:0] ^{*1}	ECM delay timer start control bit ECMNMITE007 to ECMNMITE000 correspond to error sources 7 to 0. 00: Delay timer start disabled 01: Delay timer start enabled 10 ^{*2} : Delay timer start disabled when one error counted Delay timer start enabled when two errors counted Delay timer start enabled when three errors counted 11 ^{*2} : Delay timer start disabled when one error counted Delay timer start disabled when two errors counted Delay timer start enabled when three errors counted

Note 1. The ECMNMITE007[1:0] to ECMNMITE000[1:0] bits corresponding to the ECMNMIE007[1:0] to ECMNMIE000[1:0] bits must be set to the same value as ECMNMIE007[1:0] to ECMNMIE000[1:0] when ECM delay timer is enabled.

Note 2. Delay timer is not started when four or more errors counted.

ECMNMIDTMCFGn (n = 1 to 8, x = (n-1) × 32)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM NMITE [x+55]	ECM NMITE [x+54]	ECM NMITE [x+53]	ECM NMITE [x+52]	ECM NMITE [x+51]	ECM NMITE [x+50]	ECM NMITE [x+49]	ECM NMITE [x+48]	ECM NMITE [x+47]	ECM NMITE [x+46]	ECM NMITE [x+45]	ECM NMITE [x+44]	ECM NMITE [x+43]	ECM NMITE [x+42]	ECM NMITE [x+41]	ECM NMITE [x+40]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM NMITE [x+39]	ECM NMITE [x+38]	ECM NMITE [x+37]	ECM NMITE [x+36]	ECM NMITE [x+35]	ECM NMITE [x+34]	ECM NMITE [x+33]	ECM NMITE [x+32]	ECM NMITE [x+31]	ECM NMITE [x+30]	ECM NMITE [x+29]	ECM NMITE [x+28]	ECM NMITE [x+27]	ECM NMITE [x+26]	ECM NMITE [x+25]	ECM NMITE [x+24]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.49 ECMNMIDTMCFGn Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMNMITE[x+55] to ECMNMITE[x+24]	ECM delay timer start control bit ECMNMITE[x+55] to ECMNMITE[x+24] correspond to FE level interrupts generated by error sources [x+55] to [x+24]. 0: Delay timer start disabled 1: Delay timer start enabled

ECMNIDTMCFG9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	ECM NMITE 308	ECM NMITE 307	—	ECM NMITE 305	ECM NMITE 304	ECM NMITE 303	ECM NMITE 302	ECM NMITE 301	ECM NMITE 300	ECM NMITE 299	ECM NMITE 298	ECM NMITE 297	ECM NMITE 296
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM NMITE 295	ECM NMITE 294	ECM NMITE 293	ECM NMITE 292	ECM NMITE 291	ECM NMITE 290	ECM NMITE 289	ECM NMITE 288	ECM NMITE 287	ECM NMITE 286	ECM NMITE 285	ECM NMITE 284	ECM NMITE 283	ECM NMITE 282	ECM NMITE 281	ECM NMITE 280
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.50 ECMNIDTMCFG9 Register Contents

Bit Position	Bit Name	Function
31 to 29	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
28 to 27	ECMNMITE308 to ECMNMITE307	ECM delay timer start control bit ECMNMITE308 to ECMNMITE307 correspond to FE level interrupts generated by error sources 308 to 307. 0: Delay timer start disabled 1: Delay timer start enabled
26	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
25 to 0	ECMNMITE305 to ECMNMITE280	ECM delay timer start control bit ECMNMITE305 to ECMNMITE280 correspond to FE level interrupts generated by error sources 305 to 280. 0: Delay timer start disabled 1: Delay timer start enabled

NOTE**Reserved bit**

The value of ECMNMITE bit listed as reserved for the given error input numbers in **Table 39.10**, **Table 39.11**. When read, the value after reset is returned. When writing, write the value after reset.

39.3.18 ECMEOCCFG — ECM Error Output Clear Invalidation Configuration Register

This register is a read/write register and can be written in 32-bit units.

After counter for Error Output clear invalidation exceed the value which is configured to this register, it is possible to clear non-safe status of error output by SW.

Configure to this register only if error output status is safe.

Writing to this register is protected by ECMKCPROT. Refer to **Section 39.3.11, ECMKCPROT — ECM Key Code Protection Register**, for the details of key code protection.

Value after reset: 0000 0000_H.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EOCI EN	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CMPW
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMEOUTCLRT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.51 ECMEOCCFG Register Contents

Bit Position	Bit Name	Function
31	EOCIEN	ERROROUT Clear Invalidation Function Enabled 0: Disabled 1: Enabled
30 to 17	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
16	CMPW	Indicates on execution of ECMEOUTCLRT[15:0] register setting to counter clock domain 0: Not in progress 1: In progress. While the bit is set, further setting to the ECMEOUTCLRT is prohibited.
15 to 0	ECMEOUTCLRT [15:0]	The number of clock cycles after which it is possible to clear error output by SW.

NOTES

- ECMEOCCFG register is set via P-bus, however actual “output clear invalidation counter” exists in a different clock domain. When ECMEOCCFG is configured, the value is copied across clock domains to reflect to the counter, which takes a certain time. CMPW indicates whether the reflection across clock domains is in progress.
- While CMPW is “1”, writing of ECMEOUTCLRT[15:0] is ignored. Please confirm CMPW = 0 before writing of ECMEOUTCLRT[15:0].

39.3.19 ECMPEM — ECM Pseudo Error Mask Register

This register can mask the pseudo error of “ECM compare error” to support self-diagnosis of the binding components for `ERROROUT_M` and `ERROROUT_C` pins.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSKM	MSKC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 39.52 ECMPEM Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1	MSKM	0: Pseudo error of “ECM compare error” for ECM master is NOT masked. 1: Pseudo error of “ECM compare error” for ECM master is masked.
0	MSKC	0: Pseudo error of “ECM compare error” for ECM checker is NOT masked. 1: Pseudo error of “ECM compare error” for ECM checker is masked.

Section 40 Data CRC Function B (DCRB)

This section contains a generic description of Data CRC Function B (DCRB). The first part of this section describes the features specific to this product, such as the number of units and register base addresses. The remainder of the section describes the functions and registers of DCRB.

40.1 Features

40.1.1 Number of Units

This microcontroller has the following number of DCRB units.

Table 40.1 Number of Units

Product Name	373 Pins	292 Pins
Number of Units	3	3
Name	DCRBn (n = 0 to 2)	DCRBn (n = 0 to 2)

Note: "n" in this section indicates the unit number of the individual DCRB units.

There are 373 pins in E2xFCC1 and E2M.

40.1.2 Register Base Addresses

DCRBn base addresses are listed in the following table. DCRBn register addresses are given as offsets from the base addresses in general.

Table 40.2 Register Base Addresses

Base Address Name	Base Address	Peripheral Group
<DCRB0_base>	FFF7 0000 _H	6
<DCRB1_base>	FFF7 1000 _H	6
<DCRB2_base>	FFF7 2000 _H	6

40.1.3 Clock Supply

The clock supply by and to DCRBn is given in the following table.

Table 40.3 Clock Supply

Unit Name	Unit Clock Name	Clock Supply Name
DCRBn	PCLK	CLK_LSB

40.1.4 Interrupt Requests

This module has no interrupt requests.

40.1.5 Reset Sources

The DCRBn reset sources are shown below. DCRBn is initialized by the following reset sources.

Table 40.4 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
DCRBn	All registers	✓	✓	✓	✓	—	—

40.1.6 External Input/Output Signals

This module has no external input/output signals.

40.2 Overview

40.2.1 Functional Overview

Data CRC Function B can be used to verify or generate CRC protected data streams of arbitrary length and different bit widths.

- Supported CRC polynomials
 - 32-bit Ethernet CRC
04C1 1DB7_H: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
 - 16-bit CCITT CRC
1021_H: $X^{16} + X^{12} + X^5 + 1$
 - 8-bit SAE J1850 CRC
1D_H: $X^8 + X^4 + X^3 + X^2 + 1$
 - 8-bit 0x2F CRC
2F_H: $X^8 + X^5 + X^3 + X^2 + X + 1$
- CRC generation to an arbitrary data block length
- After initialization of the DCRB data register, every write access to the DCRB input register generates a new CRC according to the chosen polynomial and the result is stored in the DCRB data register.

40.2.2 Block Diagram

The following figure shows the block diagram of Data CRC Function B.

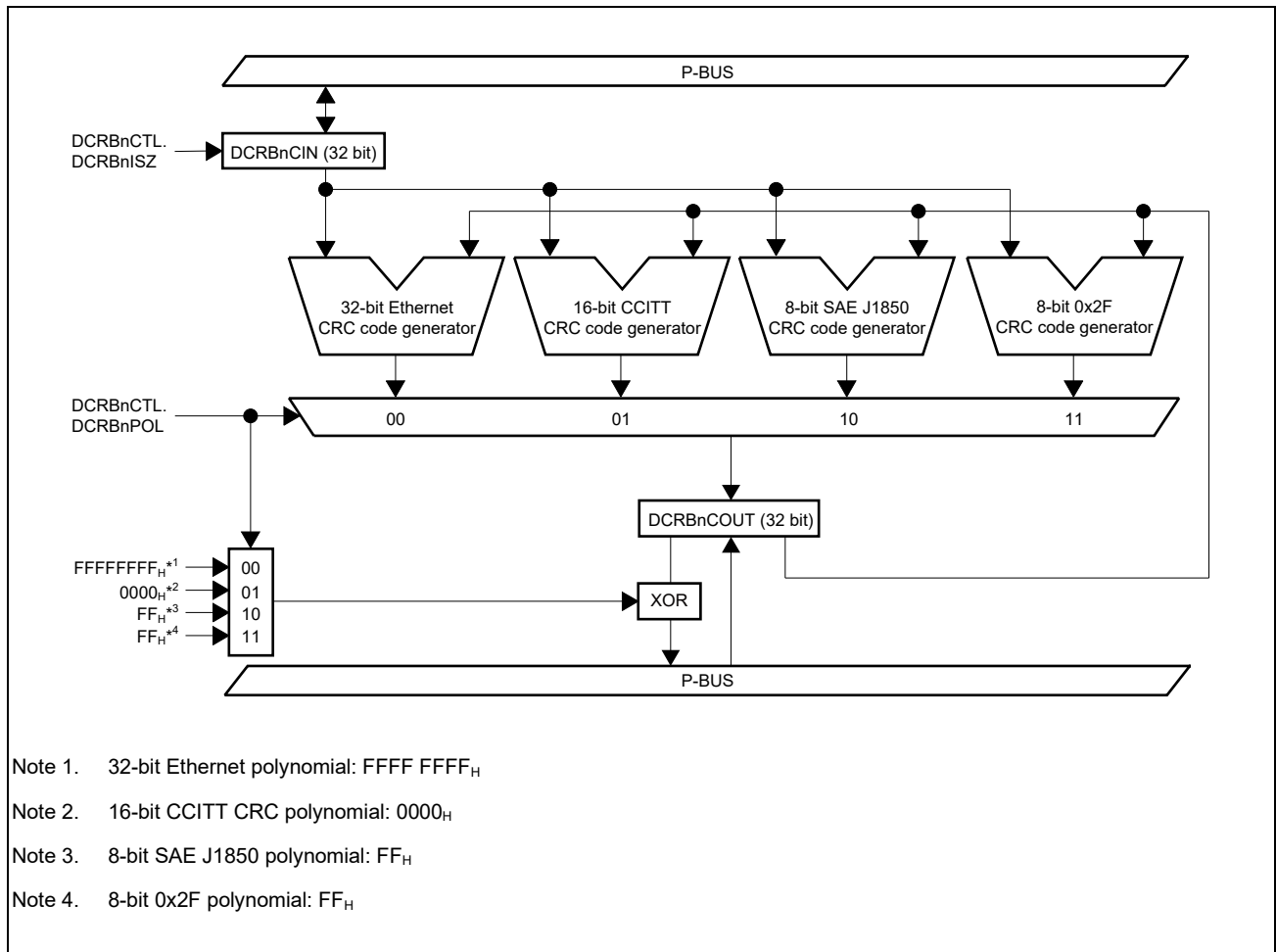
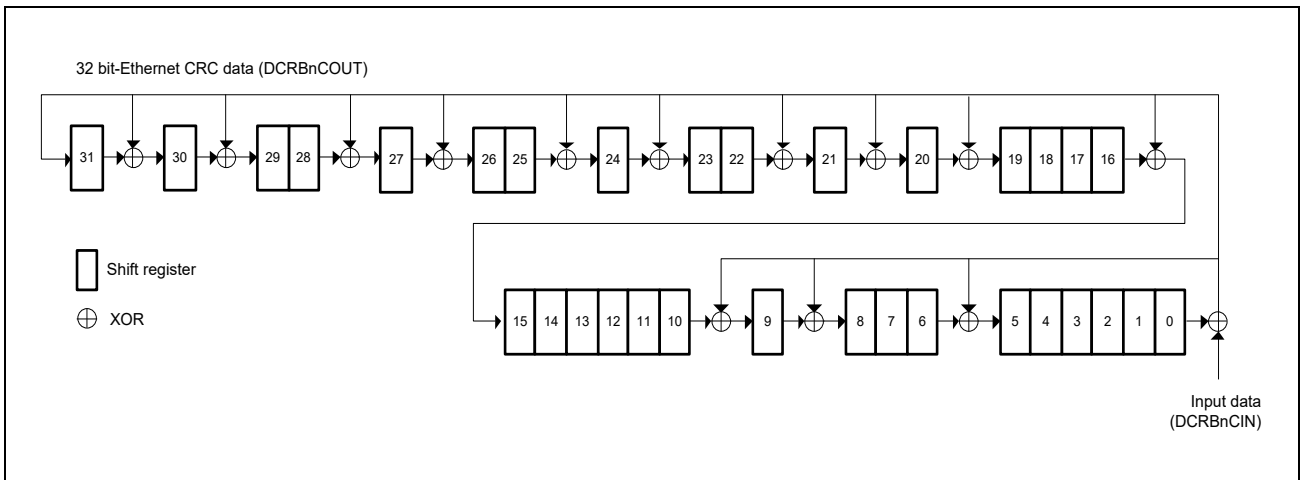


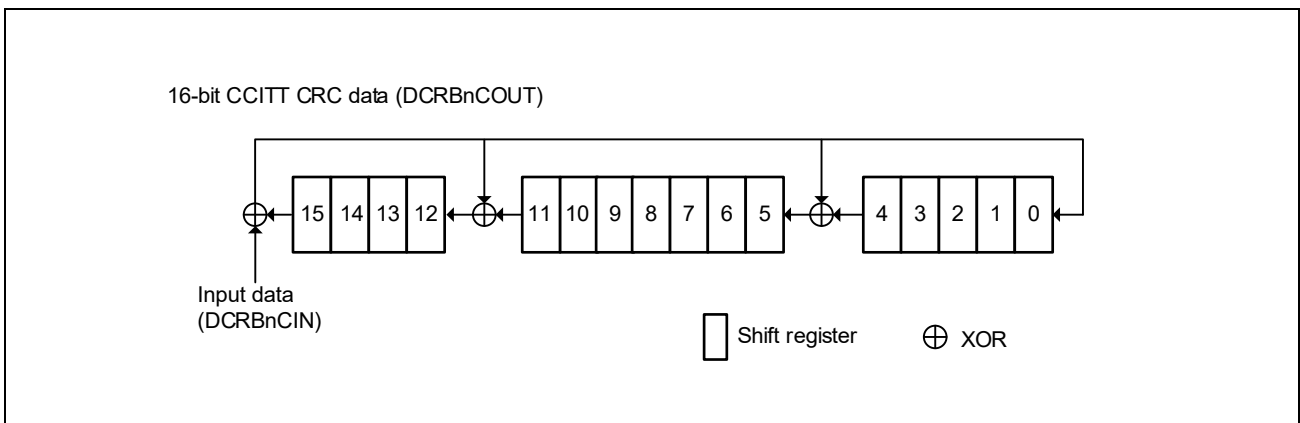
Figure 40.1 Block Diagram of Data CRC Function B

40.2.3 Configuration of CRC Calculation

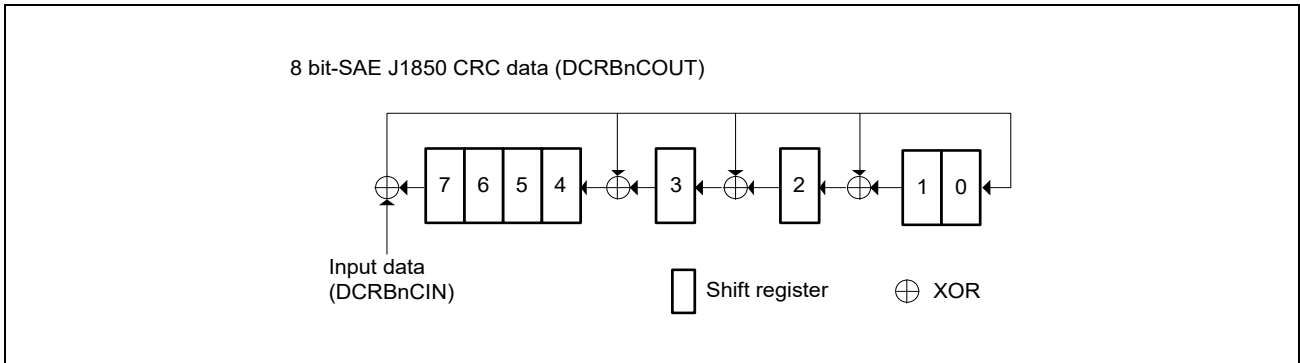
- 32-bit Ethernet



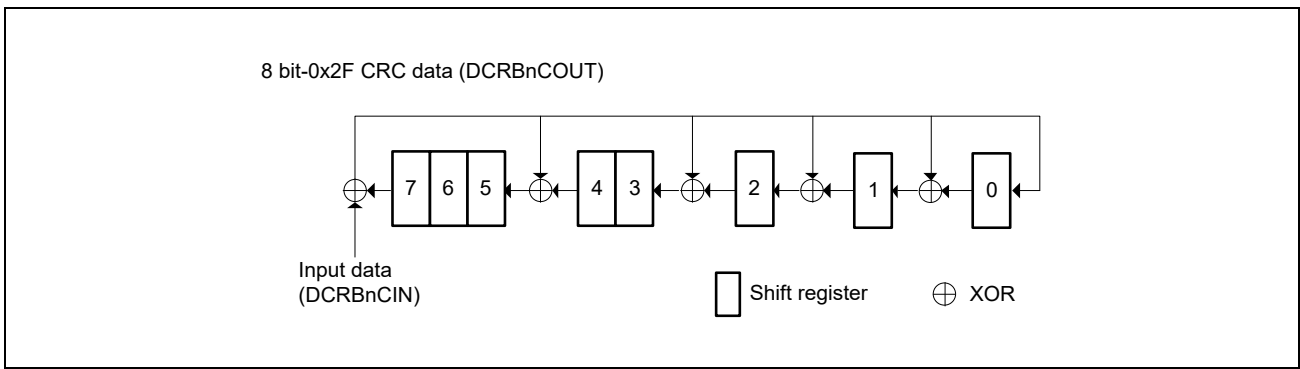
- 16-bit CCITT



- 8-bit SAE J1850



- 8-bit 0x2F



40.3 Registers

40.3.1 List of Registers

Table 40.5 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
DCRBn	DCRB Input Register	DCRBnCIN	<DCRBn_base> + 00 _H	32	—
	DCRB Data Register	DCRBnCOUT	<DCRBn_base> + 04 _H	32	—
	DCRB Control Register	DCRBnCTL	<DCRBn_base> + 20 _H	8	—

40.3.2 DCRBnCIN — DCRB Input Register

This register holds the input data for CRC calculation. The effective bit width used for CRC calculation must be set by DCRBnCTL.DCRBnISZ[1:0].

When data is written to this register, the CRC code is generated. The CRC calculation is immediately started after the DCRBnCIN register is written. The DCRBnCOUT register must be initialized with the initial value before the first data of the data block is written to the DCRBnCIN register.

The byte order in DCRBnCIN depends on the selected CRC generating function:

- 32-bit Ethernet CRC polynomial generation (DCRBnCTL.DCRBnPOL[1:0] = 00_B)

The byte order is LSB (least significant byte) first. This means that if the CRC input bit width is 8 bits (DCRBnISZ[1:0] = 10_B), the input values are in bits 7 to 0 of the DCRBnCIN register and bit 0 (the LSB) is the first bit of the input data.

- 16-bit CCITT CRC polynomial generation (DCRBnCTL.DCRBnPOL[1:0] = 01_B)

The byte order is MSB (most significant byte) first. This means that if the CRC input bit width is 8 bits (DCRBnISZ[1:0] = 10_B), the input values are in bits 7 to 0 of the DCRBnCIN register and bit 7 (the MSB) is the first bit of the input data.

- 8-bit SAE J1850 CRC polynomial generation (DCRBnCTL.DCRBnPOL[1:0] = 10_B)

The byte order is MSB (most significant byte) first. This means that if the CRC input bit width is 8 bits (DCRBnISZ[1:0] = 10_B), the input values are in bits 7 to 0 of the DCRBnCIN register and bit 7 (the MSB) is the first bit of the input data.

- 8-bit 0x2F CRC polynomial generation (DCRBnCTL.DCRBnPOL[1:0] = 11_B)

The byte order is MSB (most significant byte) first. This means that if the CRC input bit width is 8 bits (DCRBnISZ[1:0] = 10_B), the input values are in bits 7 to 0 of the DCRBnCIN register and bit 7 (the MSB) is the first bit of the input data.

Value after reset:		0000 0000 _H																								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	DCRBnCIN[31:16]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	DCRBnCIN[15:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W									

Table 40.6 DCRBnCIN Register Contents

Bit Position	Bit Name	Function
31 to 0	DCRBnCIN[31:0]	Input data for CRC calculation. The valid bits are: For 32 bit effective bit width: DCRBnCIN[31:0] For 16 bit effective bit width: DCRBnCIN[15:0] For 8 bit effective bit width: DCRBnCIN[7:0]

40.3.3 DCRBnCOUT — DCRB Data Register

This register stores the result of the CRC code generated by the CRC polynomial selected by DCRBnCTL.DCRBnPOL[1:0].

Value after reset: FFFF FFFF_H

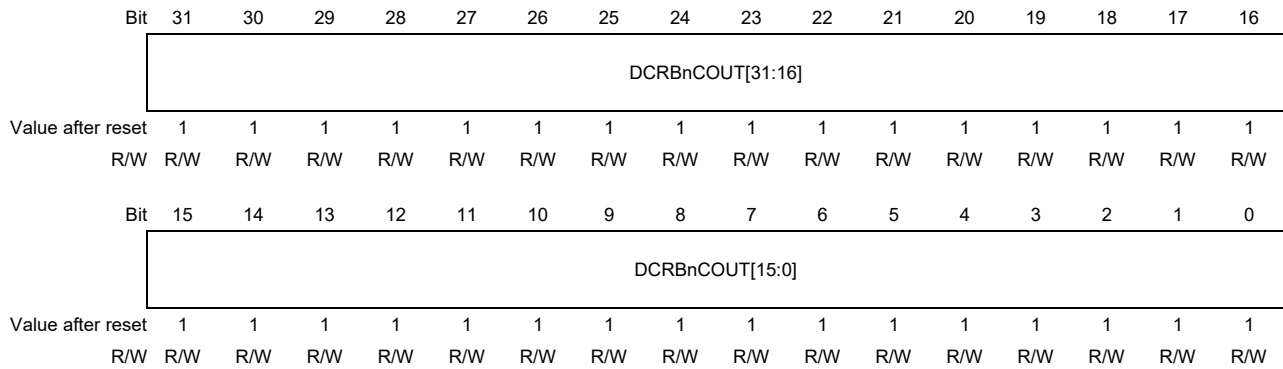


Table 40.7 DCRBnCOUT Register Contents

Bit Position	Bit Name	Function
31 to 0	DCRBnCOUT[31:0]	<p>Result of the CRC code generation.</p> <p>When the 16-bit CCITT polynomial is enabled, bits 15 to 0 show the CRC result. Bits 31 to 16 are undefined.</p> <p>When 8-bit SAE J1850 or 8-bit 0x2F polynomial is enabled, bits 7 to 0 show the CRC result. Bits 31 to 8 are undefined.</p> <p>When reading from this register, the value is XORed by hardware with the value described below for the chosen polynomial.</p> <p>32-bit Ethernet polynomial: FFFF FFFF_H</p> <p>16-bit CCITT CRC polynomial: 0000_H</p> <p>8-bit SAE J1850 polynomial: FF_H</p> <p>8-bit 0x2F polynomial: FF_H</p> <p>Therefore, for the 32-bit Ethernet polynomial, 0000 0000_H is read from this register even in the initial state.</p>

CAUTION

The read value after reset is 0000 0000_H since the 32-bit Ethernet CRC polynomial is selected as the CRC generating function after reset and the value is XORed onto FFFF FFFF_H by hardware.

This register must be initialized with the start value before the first data of the data block is written to the DCRBnCIN register.

40.3.4 DCRBnCTL — DCRB Control Register

This register controls the CRC generation process.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	DCRBnISZ[1:0]		—	—	DCRBnPOL[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W

Table 40.8 DCRBnCTL Register Contents

Bit Position	Bit Name	Function
7 to 6	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
5 to 4	DCRBnISZ[1:0]	Specifies the CRC input bit width. 00: 32 bit (DCRBnCIN[31:0]) 01: 16 bit (DCRBnCIN[15:0]) 10: 8 bit (DCRBnCIN[7:0]) 11: Setting prohibited
3 to 2	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
1 to 0	DCRBnPOL[1:0]	Specifies the CRC generating function. 00: 32-bit Ethernet CRC polynomial generation. The byte order of the DCRBnCIN register is LSB (Least Significant Byte) first. This means that if the CRC input bit width is 8 bits (DCRBnISZ[1:0] = 10 _B), the input values are in bits 7 to 0 of the DCRBnCIN register and bit 0 (the LSB) is the first bit of the input data. The half-word order of the DCRBnCIN register is LSB (Least Significant Byte) first. This means that if the CRC input bit width is 16 bits (DCRBnISZ[1:0] = 10 _B), the input values are in bits 15 to 0 of the DCRBnCIN register and bit 0 (the LSB) is the first bit of the input data. 01: 16-bit CCITT CRC polynomial generation. The byte order of the DCRBnCIN register is MSB (Most Significant Byte) first. This means that if the CRC input bit width is 8 bits (DCRBnISZ[1:0] = 10 _B), the input values are in bits 7 to 0 of the DCRBnCIN register and bit 7 (the MSB) is the first bit of the input data. 10: 8-bit SAE J1850 CRC polynomial generation The byte order of the DCRBnCIN register is MSB (Most Significant Byte) first. This means that if the CRC input bit width is 8 bits (DCRBnISZ[1:0] = 10 _B), the input values are in bits 7 to 0 of the DCRBnCIN register and bit 7 (the MSB) is the first bit of the input data. 11: 8-bit 0x2F CRC polynomial generation The byte order of the DCRBnCIN register is MSB (Most Significant Byte) first. This means that if the CRC input bit width is 8 bits (DCRBnISZ[1:0] = 10 _B), the input values are in bits 7 to 0 of the DCRBnCIN register and bit 7 (the MSB) is the first bit of the input data.

NOTE

After changing the CRC generating function (DCRBnCTL.DCRBnPOL[1:0]), the DCRBnCOUT register must be initialized.

CAUTIONS

- The CRC bit width (DCRBnCTL.DCRBnSZ[1:0]) must be set according to the data block bit width. Switching the CRC bit width is not allowed during processing of a data block (a data block consists of N bytes, half words or words). After the final CRC result is read from the DCRBnCOUT register, the bit width can be changed and the DCRBnCOUT register must be initialized with the initial value.
 - Switching the CRC polynomial (DCRBnCTL.DCRBnPOL[1:0]) is also not allowed during processing of a data block.
-

40.4 Operation

Data CRC Function B generates a CRC (cyclic redundancy check) of arbitrary data block length. The data can be forwarded to the Data CRC Function in 8-, 16- or 32-bit units. The CRC polynomial can either be selected for 32-bit Ethernet, 16-bit CCITT, 8-bit SAE J1850 or 8-bit 0x2F polynomial CRC, and the initial starting value must be set at the DCRBnCOUT register before the first write access to the DCRB input register (DCRBnCIN) is performed.

After the last write access to the DCRBnCIN register is performed, the result can be read out from the DCRBnCOUT register after one clock period.

The flow chart below shows the CRC generation procedure.

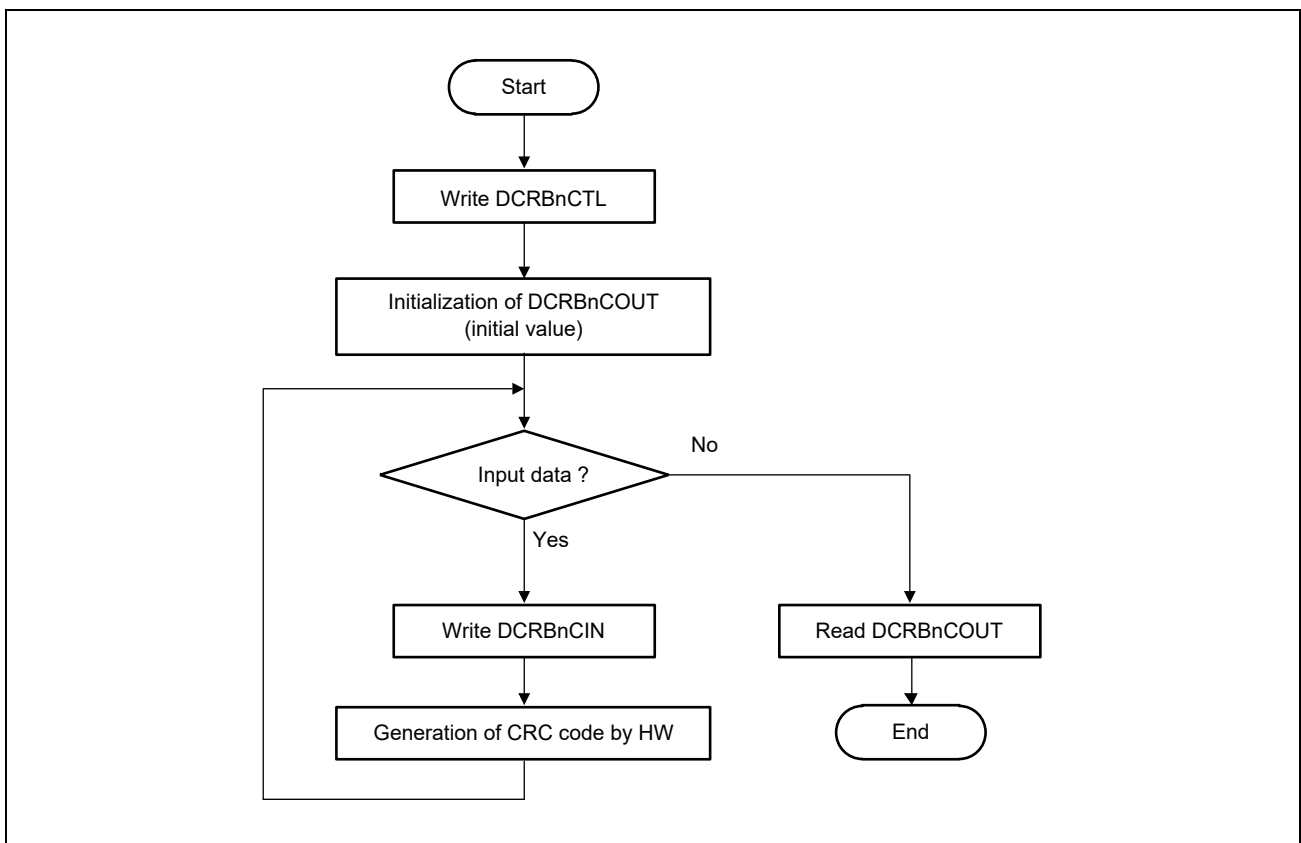


Figure 40.2 Data CRC Function B Flow Diagram

NOTES

- The initial value to be written to DCRBnCOUT can be chosen by the user.
- When reading from register DCRBnCOUT, the value is XORed by hardware with the value specified in the chosen polynomial.
- Generation of CRC code and storage to DCRBnCOUT are done by hardware.
- All registers are accessible by CPU or by CPU-independent read/write access via DMA/DTS. Independent from access of the DCRB via CPU or DMA/DTS the verification of the CRC code (CRC-check result in DCRBnCOUT) has to be performed by SW.

Section 41 Flash Memory

This product incorporates Code Flash Memory and Data Flash Memory.

Product Name	Code Flash Memory		Data Flash Memory	
	User Area	User Boot Area	Data Area	Exclusively for ICUM
RH850/E2x-FCC1 (for E2M)	8M bytes	64K bytes	192K bytes	32K bytes
RH850/E2M	8M bytes	64K bytes	192K bytes	32K bytes

41.1 Features

- Code Flash

- Capacity: 8 MB of User Area and 64 KB of User Boot Area.
- Multi banked configuration in Multi CPU devices.
- Program unit: 512 bytes
- Erase unit: 16 KB for 8 blocks and 64 KB for remaining blocks
- OTP (One Time Programmable) supported for each block.

- Data Flash

- Capacity: 192 KB (dual banked: 96 KB + 96 KB) and 32 Kbytes for data area exclusively for ICUM
- Program unit: 4, 8, 16, 32 bytes
DMA can initiate 4 bytes program in multiple time without software overhead. (within one Data Flash*1)
- Erase unit: 64 bytes x N (N = 1, 2, 3 ...) (within one Data Flash*1)

Note 1. The 96 KBytes Data Flash bank consists of two 48 KBytes Data Flash (EEP) called EEP0-0 and EEP0-1, or EEP1-0 and EEP1-1. 32 KBytes Data Flash bank for ICUM consists of one 32 KBytes Data Flash called EEP2. These EEPs are treated as different Data Flash.

- Programming method

- Serial programming: Programming of flash memory by external flash memory programmer through serial interface or CAN/CAN-FD interface.
This Mode uses the program to control flash memory rewriting included in a microcomputer (Boot Firmware). Boot Firmware communicate with external flash memory programmer by a serial interface or CAN/CAN-FD interface.
When using CAN/CAN-FD interface, Boot Firmware receives a user program for flash memory rewriting and starts this.
- Self programming: Programming of flash memory by a user program written on Code Flash already.

- Support for security functions to protect against illicit tampering and illicit reading with data in flash memory
- Support for protection functions to protect against erroneous programming / erasure of the flash memory
- ECC support for error detection and correction for both Code Flash and Data Flash
 - Reading erased Code Flash results in an ECC error.
 - Reading erased Data Flash results in an ECC error with high probability.
Data Flash is complementary read (CR) Flash. The read value of erased CR Flash is undefined.
- Background Operation (BGO) support
 - Code Flash Read is possible during programming / erasing of Data Flash.
 - One bank Code Flash Read is possible during programming / erasing of the other bank of Code Flash.
 - One bank Data Flash Read is possible during programming / erasing of the other bank of Data Flash.
- Dual Operation support
 - The bank Data Flash for ICUM programming / erasing is possible during programming / erasing of the other bank of Data Flash.
- Suspend/Resume support for both Code Flash and Data Flash
 - Programming / erasing operation can be suspended. Suspended operation can be resumed.
It can be used for interruption processing received during self- programming.
- The Settings of this product can be configured in Flash Extra area of Flash Memory.
Flash Extra Areas is divided into Configuration Setting Area, Security Setting Area and Block Protection Area.
 - Configuration Setting Area: To store the System Configuration Parameters. (Flash Option Byte, Reset Vector, etc.)
 - Security Setting Area: To store the Security Parameters. (ID Codes, Security Setting flag, etc.)
 - Block Protection Area: To store the Code Flash Protection Settings. (OTP flag, etc.)
- Interrupt Requests
The flash memory supports an interrupt to indicate completion of processing by the flash sequencer.

Table 41.1 Interrupt Requests

Interrupt Name	Description	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
INTFACI0ENDNM	Flash sequencer of code flash and data flash for bank A/B processing end interrupt	33	—	—
INTFACI1ENDNM	Flash sequencer of the data flash for ICUM processing end interrupt	35	—	—
INTDMAFL0	DMA request of DMA programming for the data flash bank A/B	—	group0-218	group0-124
INTDMAFL1	DMA request of DMA programming for the data flash bank for ICUM	—	group0-219	group0-125

41.2 Structure of Memory

Table 41.2 gives information on all of these areas.

Table 41.2 Area on the Flash memory

Area	Address	Peripheral Group
User Area (code flash)	See Figure 41.1 . Mapping of Code Flash Memory	—
User Boot Area (code flash)	See Figure 41.1 . Mapping of Code Flash Memory	—
Product Info Area (code flash)	See Figure 41.1 . Mapping of Code Flash Memory	—
ECC test Area (code flash)	See Figure 41.1 . Mapping of Code Flash Memory	—
Data Area (data flash)	See Figure 41.2 . Mapping of Data Flash Memory	1
Security Setting Area (Flash Extra Area)	See Figure 41.3 . Mapping of Flash Extra Area	1
Configuration Setting Area (Flash Extra Area)	See Figure 41.3 . Mapping of Flash Extra Area	1
Block Protection Area (Flash Extra Area)	See Figure 41.3 . Mapping of Flash Extra Area	1

41.2.1 Mapping of Code Flash Memory

Figure 41.1 illustrates the mapping of the code flash memory for the 8-MB (E2x-FCC1 for E2M) device. The user area in the code flash memory of this product is divided into 16- and 64-Kbyte blocks, which can be erased individually. A single block of 64-Kbyte user boot area is also incorporated. The user area can be used to store the user program. The user boot area can be used to store a non-rewritable boot program during user program operation, such as a boot program for rewriting code flash memory through a selected user interface. The Product Info Area and ECC test Area store Product information and data to test ECC decoder of code flash. (See **Section 41.9.3, Registers Related to Product Information**) These areas are not rewritable. The BGO condition is same as User Area (Bank A)/User Boot Area (Bank A).

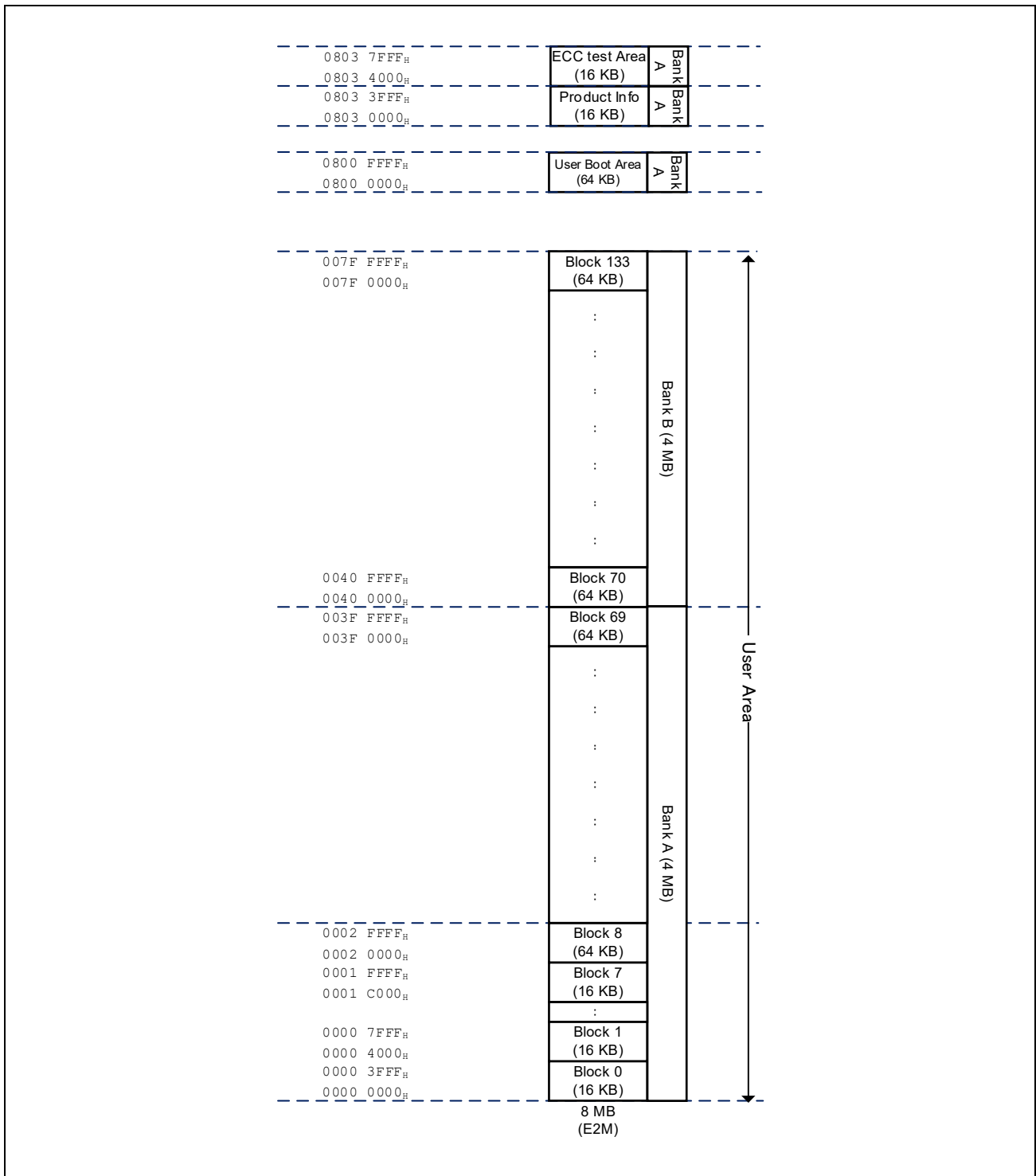


Figure 41.1 Mapping of Code Flash Memory

41.2.2 Mapping of Data Flash Memory

The data area of the data flash memory in this product is divided into 64-byte blocks, with each being a unit for erasure.

Figure 41.2 shows data flash memory map.

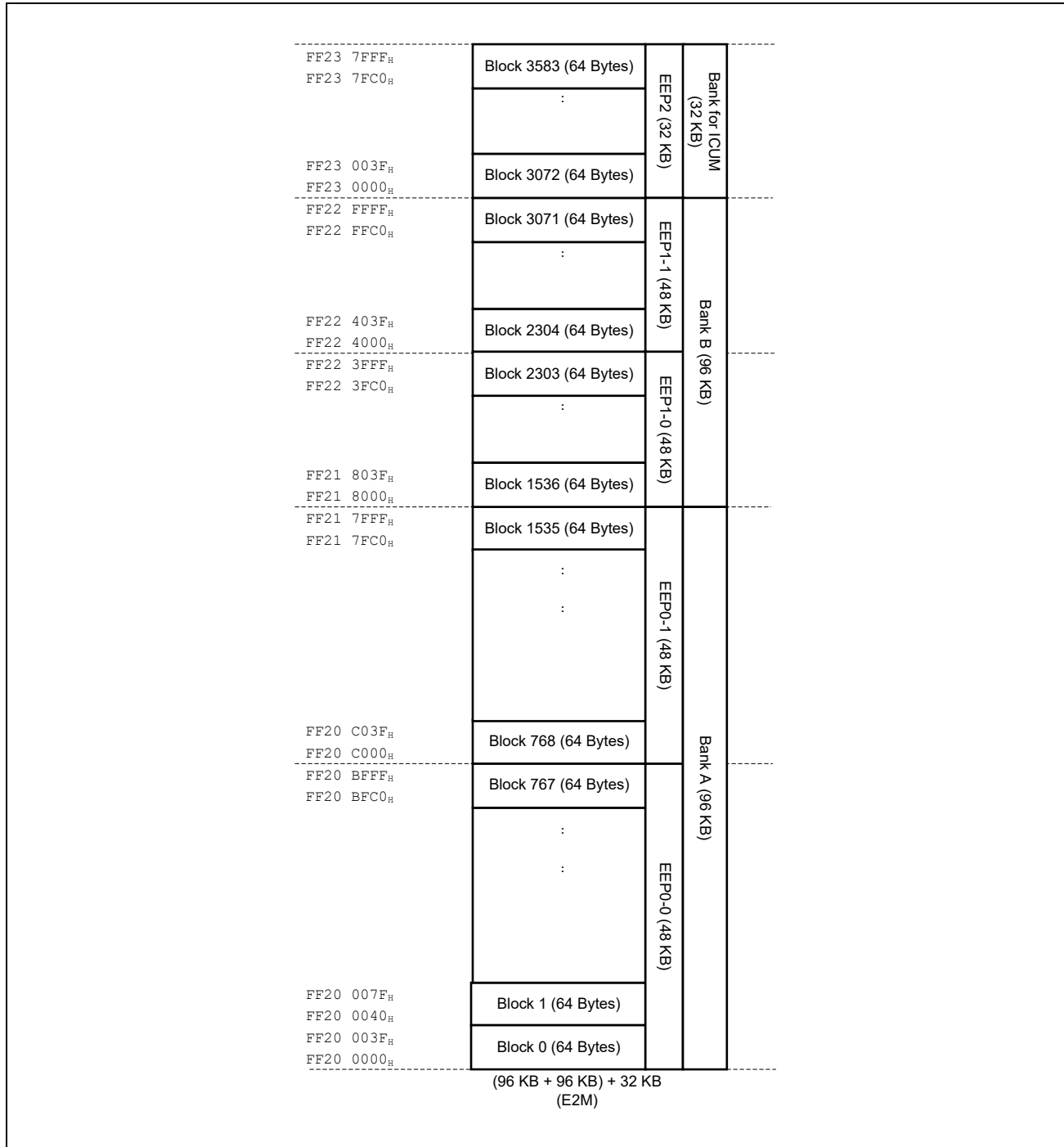


Figure 41.2 Mapping of Data Flash Memory

41.2.3 Mapping of Flash Extra Area

Flash Extra Area in this product has 3 of areas (Configuration Setting Area, Security Setting Area, and Block Protection Area).

- For Security Setting Area, refer to **Section 42.6, Security Settings Area**.
- For Configuration Setting Area, refer to **Section 41.10, Configuration Setting Area (Option Bytes, Reset Vector)**.
- For Block Protection Area, refer to **Section 42.5.1, Block Protection Function** and **Section 42.5.2, OTP (One Time Programming) Function**.

Each Extra Area supports the exclusive rewriting command (32-byte rewriting or 4-byte rewriting).

Each Extra Area can be set OTP (One Time Programming) by 4bytes unit.

Figure 41.3 shows Extra Area memory map. This area is placed in the data flash memory.

The condition of BGO and Dual Operation is the same as DataFlashBank which was placed.

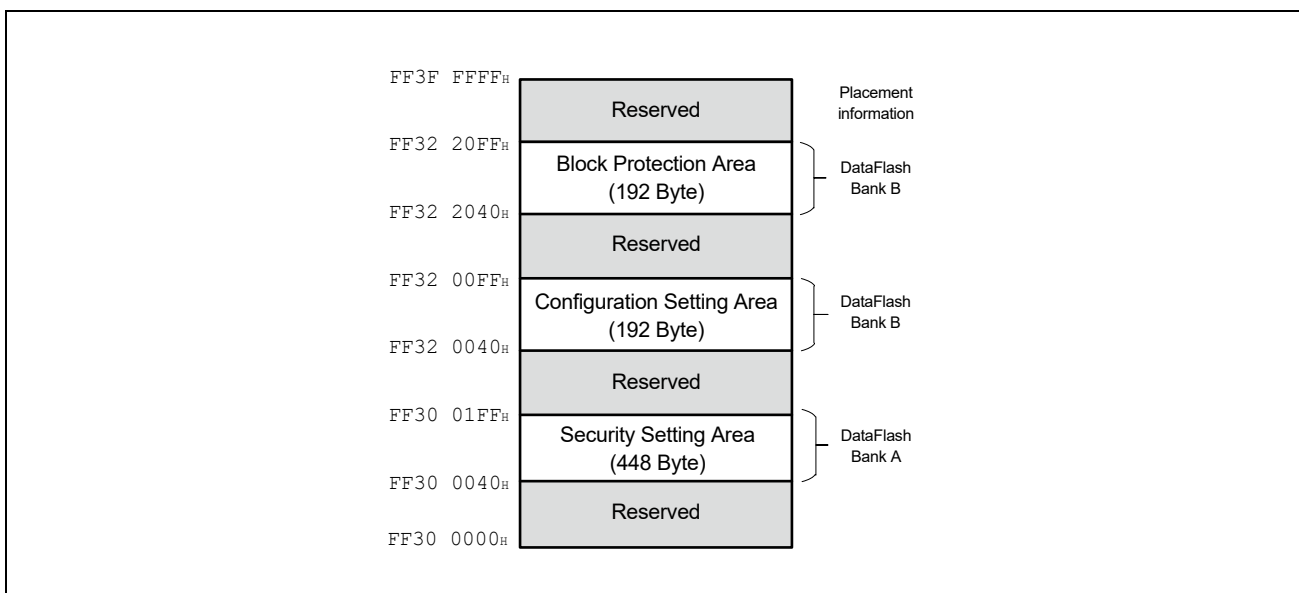


Figure 41.3 Mapping of Flash Extra Area

41.2.4 Examples of the Chip Writing

Set OTP after programming target data, because OTP setting becomes effective without reset.

The figure below is an examples of the chip writing.

Figure 41.4 shows the examples of the chip writing.

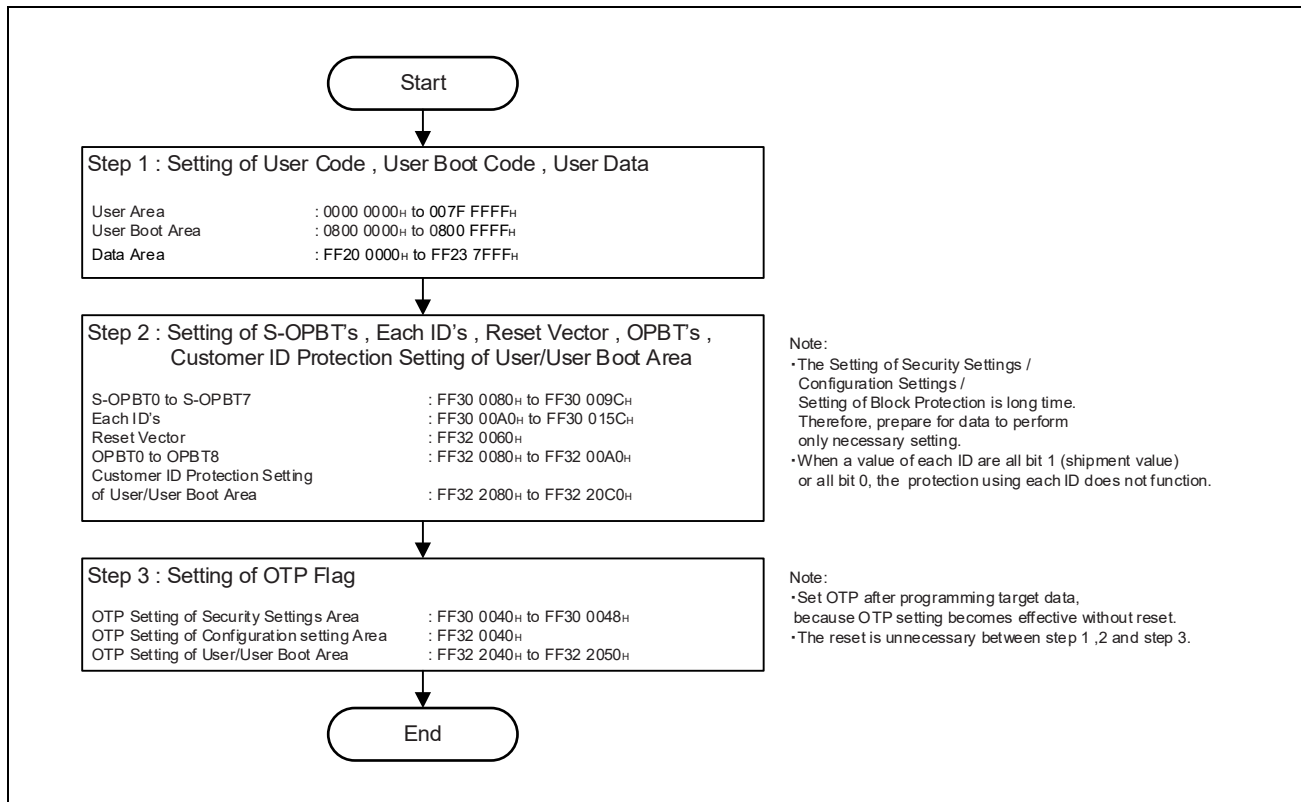


Figure 41.4 Examples of the Chip Writing

41.3 Operating Modes Associated with Flash Memory

Figure 41.5 is a diagram of the mode transitions associated with the flash memory. For the procedures for setting the modes, refer to Section 5, Operating Modes.

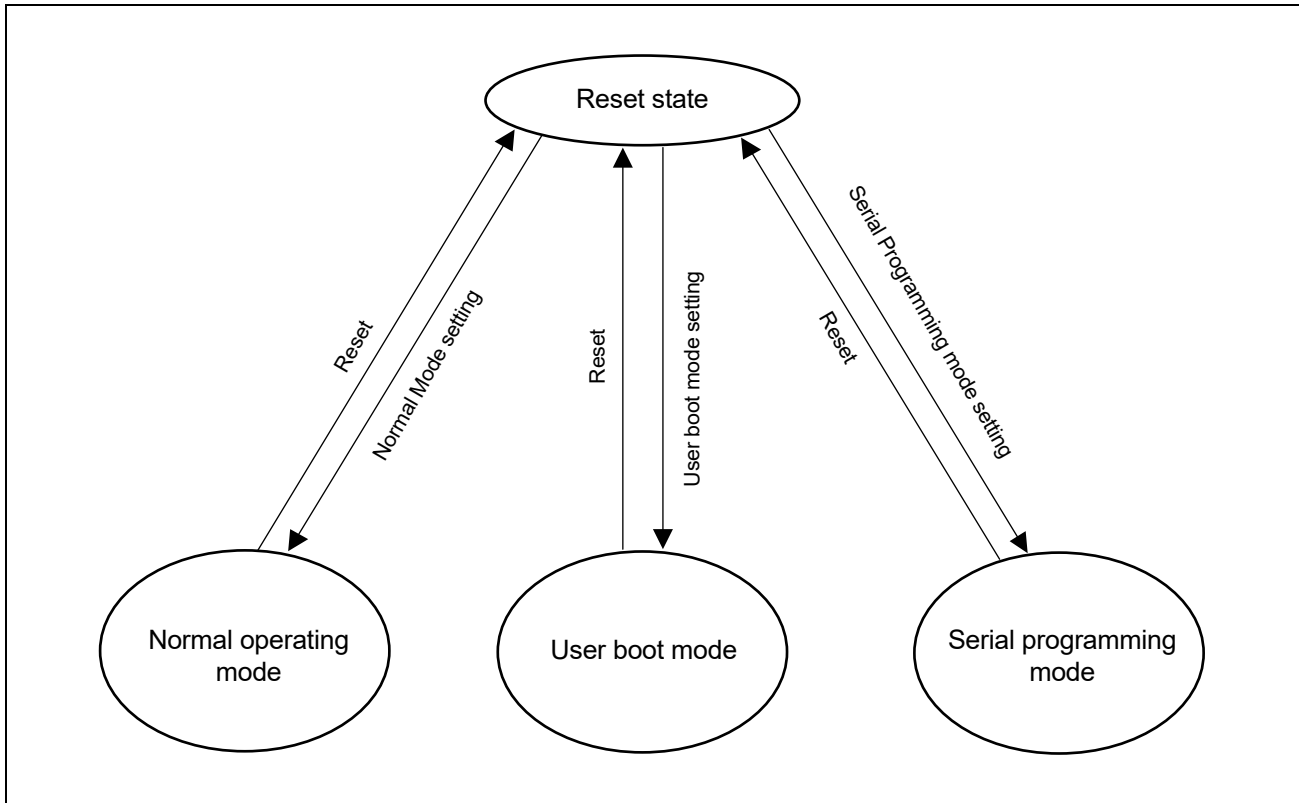


Figure 41.5 Mode Transition Associated with Flash Memory

Table 41.3 shows the flash memory area which is programmable and erasable in each mode and the boot program after reset release.

Table 41.3 Programmable and Erasable Area in Each Mode and the Boot Program after Reset Release

Item	Normal Operating Mode	User Boot Mode	Serial Programming Mode
Programmable and erasable area	<ul style="list-style-type: none"> User area Data area 	<ul style="list-style-type: none"> User area Data area 	<ul style="list-style-type: none"> User area User boot area Data area
Boot program after reset release	Program in user area (Changeable by using the variable reset vector)	Program in user boot area	Boot Firmware program for serial programming

41.4 Functional Overview

The flash memory of this device can be updated via a serial interface by a dedicated flash memory programmer (serial programming), before being mounted on the target system or on a flash adapter system.

When using CAN-FD/CAN interface for a connection with flash memory programmer, it's possible to forward a user program for rewriting to internal RAM and start.

Furthermore, security functions to prohibit updating of the user program written in the flash memory are incorporated, and this can prevent tampering by third parties.

Programming by the user program (self-programming) is suited for applications where the target system program may require updating after deployed to the end user. Protection features for the safe rewriting of the flash memory are also incorporated. Furthermore, interrupt processing during self-programming is supported, so programming can proceed at the same time as external communications, etc., and this allows programming under various conditions.

Table 41.4 gives an overview of the methods of programming and the corresponding operating modes.

Table 41.4 Methods of Programming

Method of Programming	Overview of Functionality	Operating Mode
Serial programming	<p>A dedicated flash-memory programmer allows on-board programming the flash memory after the device is mounted on the target system.</p> <p>A dedicated flash memory programmer and dedicated programming adapter board allow off-board programming of the flash memory, i.e. programming of the device before it is mounted on the target system.</p>	Serial programming mode
Self-programming	<p>The user program that is written to code flash memory in advance by serial programming executing also allows updating the flash memory.</p> <p>When data flash memory is being programmed with self-programming, the BGO function enables instruction fetch and data read from code flash memory. Thus, data flash memory can be programmed by executing a program on code flash memory prepared for flash programming.</p> <p>When code flash memory is being programmed with self-programming, instruction fetch and data read from code flash memory for a bank during the programming are prohibited. This programming needs to be carried out by executing a program prepared for flash programming that has been transferred to the Cluster-RAM or the Local-RAM in advance or a program on a different bank.</p>	Normal operating mode, User boot mode

When executing self-programming, see *the RH850/E2x-FCC1 Flash Memory User's Manual: Hardware Interface* which this product targets.

Table 41.5 lists the functions of the on-chip flash memory. Dedicated flash memory programmer commands enable serial programming, while reading of the on-chip flash memory by interface operation of flash memory or the user program enables self-programming.

Table 41.5 Basic Functions at a Glance

Function	Description in Overview	Level of Support (✓: Supported, x: Not Supported)	
		Serial Programming	Self-Programming
Blank checking	This is used to check a specified block to ensure that writing to it has not already proceeded. Results of reading from code flash memory and data flash memory to which nothing has been written after erasure are not guaranteed, so use blank checking to confirm that writing to memory has not proceeded after erasure.	✓	✓
Block erasure	This is for erasing the contents of a specified block of memory.	✓	✓
Programming	This is for writing to a specified address.	✓	✓
Reading	Data that have been written to the flash memory are read out.	✓	✓
Setting of Security Settings (Setting an ID Codes)	Security Settings (ID codes, Debugger connection prohibited setting, etc.) are set and the security is made effective. An ID setting is made for use in controlling the connection of a dedicated flash memory programmer for serial programming, controlling of the on-chip debugger, and etc. Refer to Section 42, Basic Hardware Protection (BHP) for Security settings.	✓	✓
Setting of Block Protection	A specified block of code flash memory is set for Customer ID Protection Setting and OTP (one-time programming). (OTP can only be set, that is, it is not possible to release a block's OTP setting). Refer to Section 42, Basic Hardware Protection (BHP) for settings.	✓	✓
Setting of Configuration Settings	Configuration Settings (Option bytes, Reset Vector, etc.) are set to change them from the initial values for this product.	✓	✓
CAN-FD/CAN Bootstrap	Transfer user program by CAN-FD/CAN interface to internal RAM, and execute the program.	✓	x

For details on serial programming, refer to *the Renesas Flash Programmer Flash Programming Software User's Manual*.

For details on self-programming, refer to *the RH850/E2x Flash Memory User's Manual: Hardware Interface* which this product targets.

The on-chip flash memory supports various security functions.

Setting of OTP and authentication of the ID code are security functions for use with serial programming and self-programming.

In serial programming, authentication of the ID code, and prohibiting connection of a dedicated flash memory programmer are available for use as security functions.

Table 41.6 lists security functions that are supported by the on-chip flash memory and **Table 41.7** lists Available Operations and Security Settings. Security Settings is established in the Security Setting Area. For Security Settings, refer to **Section 42, Basic Hardware Protection (BHP)**.

Table 41.6 Summary of Security Functions

Function	Description
OTP	OTP can be individually set for each block of the user area and the user boot area of code flash memory. When the OTP setting is made for an area, programming by serial programming and by self programming is prohibited. Once set, the OTP setting cannot be released.
ID authentication	The result of ID authentication can be used to control the connection of a dedicated flash memory programmer for serial programming. The result of ID authentication can also be used to control enabling of code flash memory and data flash memory writing by self-programming.
Prohibition of connection of a dedicated flash memory programmer	The connection of a dedicated flash memory programmer for serial programming is prohibited.

Table 41.7 Available Operations and Security Settings

Function	All Security Settings and Erasure, Programming, and Read Operations (✓: Executable, ×: Not Executable, —: Not Supported)		Point for Caution Regarding the Security Setting	
	Serial Programming	Self Programming	Serial Programming	Self Programming
OTP	<ul style="list-style-type: none"> • Areas for which OTP is set – Block erasure commands: × – Programming commands: × – Read commands: ✓ • Areas for which OTP is not set – Block erasure commands: ✓ – Programming commands: ✓ – Read commands: ✓ 	<ul style="list-style-type: none"> • Areas for which OTP is set – Block erasure: × – Programming: × – Reading: ✓ • Areas for which OTP is not set – Block erasure: ✓ – Programming: ✓ – Reading: ✓ 	The OTP setting cannot be released.	The OTP setting cannot be released.
ID authentication	<ul style="list-style-type: none"> • When the ID codes do not match – Block erasure commands: × – Programming commands: × – Read commands: × • When the ID codes match – Block erasure commands: ✓ – Programming commands: ✓ – Read commands: ✓ 	<ul style="list-style-type: none"> • When the ID codes do not match – Block erasure: ×^{*1} – Programming: ×^{*1} – Reading: ✓ • When the ID codes match – Block erasure: ✓ – Programming: ✓ – Reading: ✓ 	Serial programming ID authentication is always in effect.	Customer ID authentication enabled when Customer ID Protection setting bit is set. / DataFlash ID authentication enabled when DPROT bit is set.
Prohibition of the connection of a dedicated flash memory programmer	<ul style="list-style-type: none"> • Block erasure commands: × • Programming commands: × • Read commands: × 	<ul style="list-style-type: none"> • Block erasure: ✓ • Programming: ✓ • Reading: ✓ 	Since execution of the serial programming command is prohibited, initialization of the setting for prohibition is not possible.	If OTP is not set, initialization of the settings prohibited is possible after ID authentication.

Note 1. The block where protection setting was accomplished.

The on-chip flash memory supports various protection functions. **Table 41.8** lists protection functions that are supported by the on-chip flash memory.

Table 41.8 Summary of Protection Functions

Function	Description
User boot protection	Programming or erasure of the user boot area by self programming is prohibited. Programming or erasure of the user boot area by serial programming is available.
Variable reset vector	The protection settings include control of the reset vector. As shown in Figure 41.6 , after programming of a new boot program while leaving the existing boot program in place, changing the reset vector is a safe way to change to the area holding the new boot program. The areas that can be specified by using the reset vector are the user area.

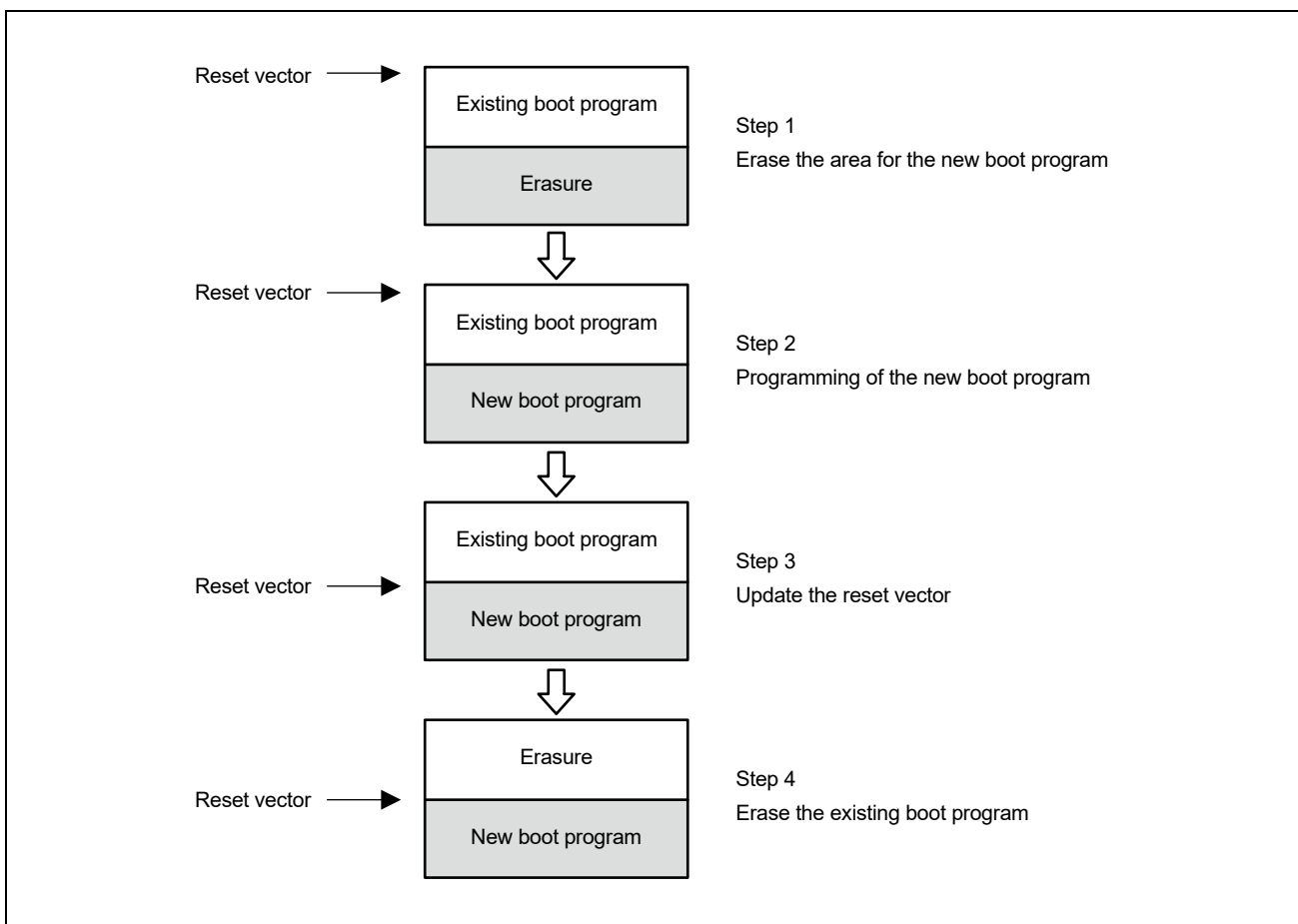


Figure 41.6 Utilizing the Variable Reset Vector Function to Update the Boot Program

41.5 Serial Programming

A dedicated flash memory programmer can be used to handle the flash memory in serial programming mode.

Serial Programming

The microcontroller is mounted on the system board at the time of serial programming. Providing a connector to the board enables handling of the target microcontroller by the flash memory programmer to proceed.

In one example, the recommended environment for programming the flash memory of the microcontroller with data is the Renesas Flash Programmer Flash Programming Software.

NOTE

For details on Renesas Flash Programmer, see *the Renesas Flash Programmer Flash Programming Software User's Manual*.

41.6 Communication Modes

41.6.1 Asynchronous Flash Programming Interface — 2-Wire UART

The double-wire asynchronous serial programming interface, 2-wire UART is connected to the flash memory programmer with the following ports.

- FPDR(TDI): Receive data input
- FPDT(TDO): Transmit data output

41.6.2 Synchronous Flash Programming Interface CSI

The synchronous serial programming interface CSI is connected to the flash memory programmer with the following ports.

- FPDR(TDI): Receive data input
- FPDT(TDO): Transmit data output
- FPCK(TCK): Serial clock input

The flash memory programmer outputs the serial data clock SCK, and the microcontroller operates as a slave.

41.6.3 CAN-FD/CAN Bootstrap

Transfer user program by CAN-FD/CAN interface to internal RAM, and execute the program. (e.g. User customized flash programming software, etc.).

CAN-FD/CAN interface is connected to the flash memory programmer with the following ports.

- CRX: P32_0, Schmitt1 input type
- CTX: P32_1, Push-pull output

Procedure summary of a CAN-FD/CAN bootstrap.

step 1: Select CAN-FD/CAN boot. (Please refer **Section 41.6.4, Selection of Communication Method**)

step 2: Boot Firmware starts, and CAN-FD/CAN communication with external flash memory programmer is established.

step 3: A user program is forwarded from external flash memory programmer to internal RAM of the microcontroller via CAN-FD/CAN interface. Boot Firmware receives a user program for flash memory rewriting and starts this.

step 4: Processing is moved from a Boot Firmware to a forwarded user program. The transfer program can not return to Boot Firmware.

41.6.4 Selection of Communication Method

In this product, communication method can be selected by pulse input to the MD0 pin (up to 5 pulses) after transition to the Serial programming mode. The MD0 pulse is generated by a dedicated flash memory programmer.

Figure 41.7 shows the relation between the number of pulses and communication method.

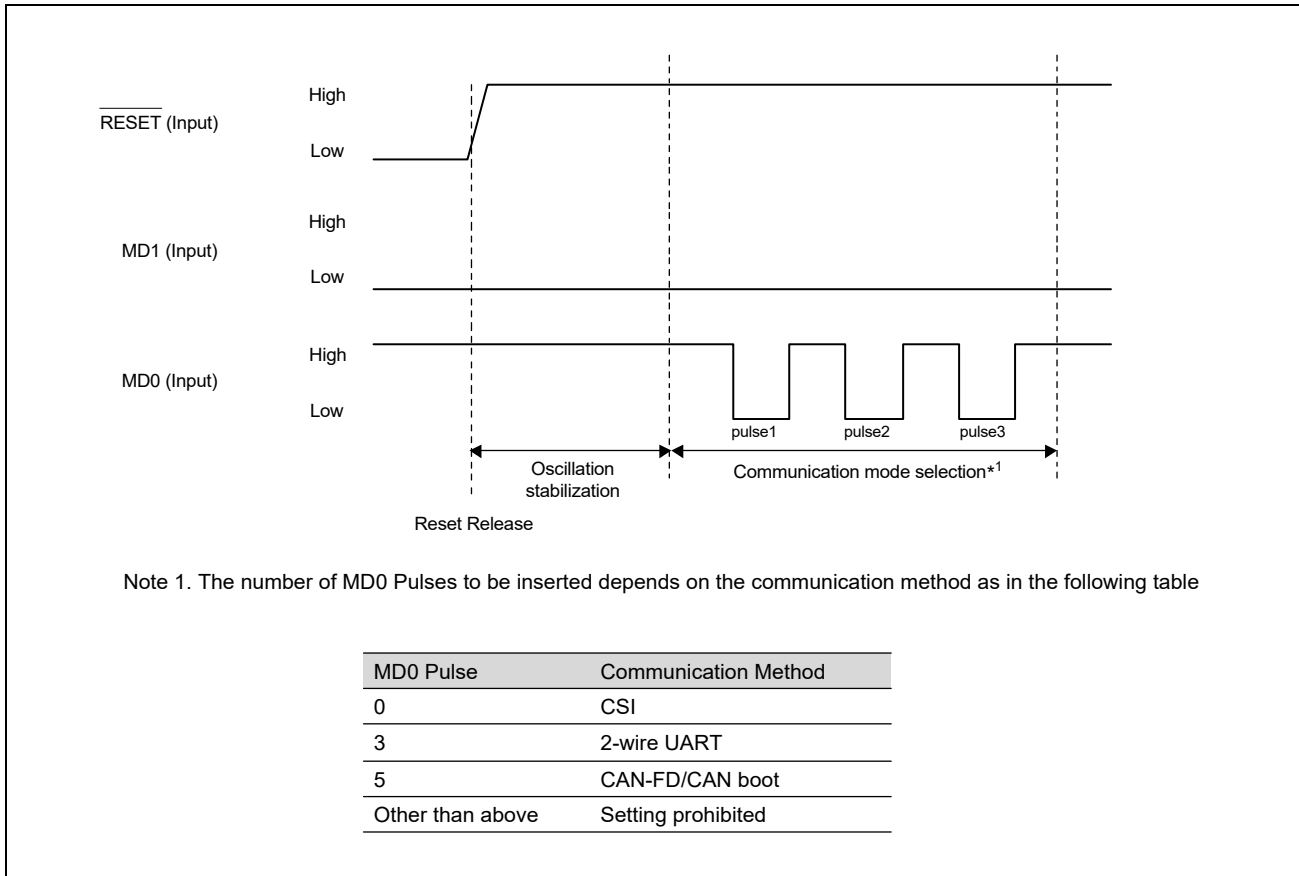


Figure 41.7 Selection of Communication Method

41.7 Self-Programming

41.7.1 Outline

This product supports programming of the flash memory by user program itself. The code flash and data flash memory can be programmed by using the commands of flash memory application command interface (FACI) for flash memory programming in user's applications. Therefore, update of the user program and programming of constant data fields can be possible.

When the data flash memory is programmed, the background operation facility makes it possible to execute a programming program from the code flash memory to program the data flash memory. Furthermore, the programming program can be copied to the local RAM or the cluster RAM in advance and executed to program the data flash memory.

When the code flash memory is programmed, the background operation facility makes it possible to execute a programming program from the code flash memory of a bank to program the code flash memory of other bank. Furthermore, the programming program can be copied to the local RAM or the cluster RAM in advance and executed to program the code flash memory.

For comprehensive information on flash self-programming, see *the RH850/E2x-FCC1 Flash Memory User's Manual: Hardware Interface* which this product targets.

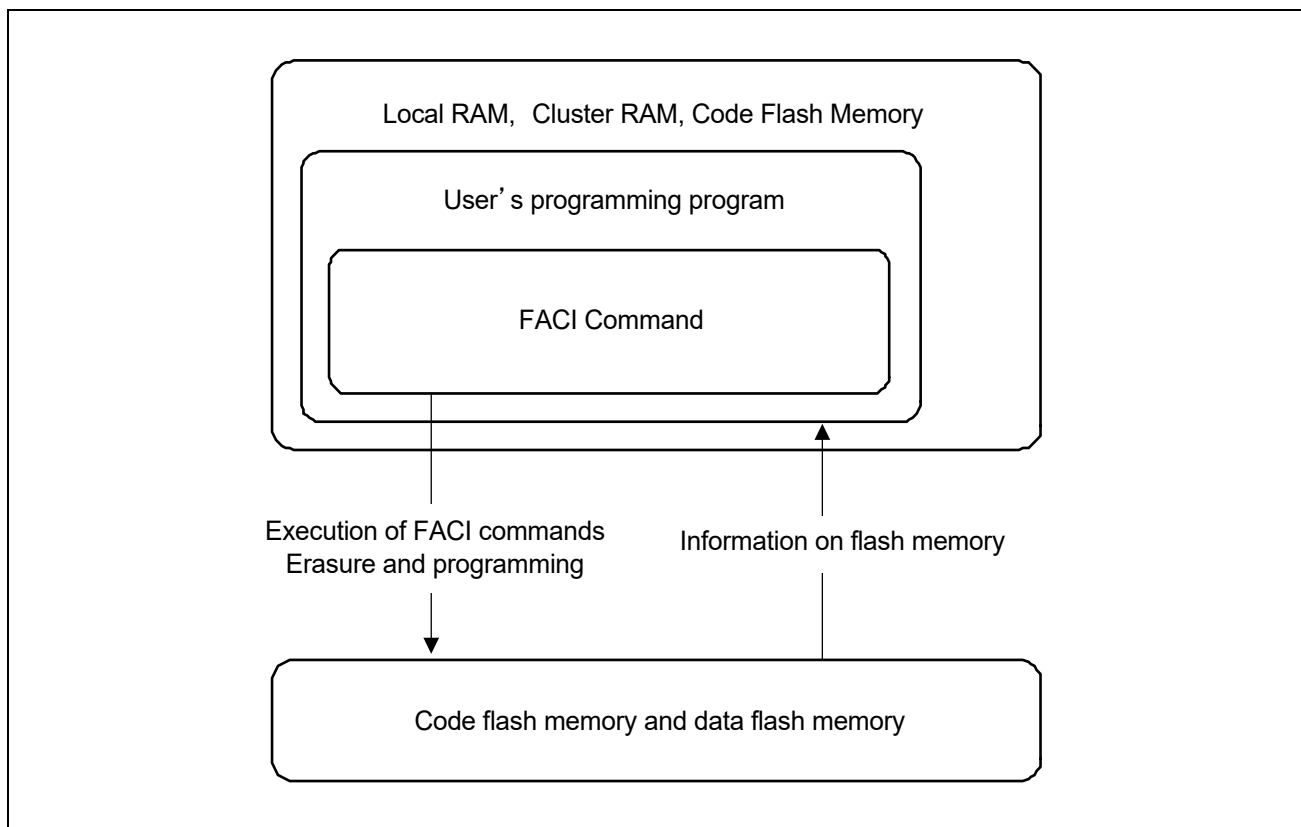


Figure 41.8 Concept of Self-Programming

41.7.2 Background Operation

Background operations can be used when the combination of the flash memory for programming/erasure and the flash memory for reading is any of those listed below.

Table 41.9 Conditions under which Background Operation is Usable

Range for Programming / Erasure	Range for Reading
Data Flash Memory	Code Flash Memory
One bank of Code Flash Memory	Other bank of Code Flash Memory The-bank of Data-Flash Memory for ICUM
One-bank of Data Flash Memory	Other bank of Data flash memory

41.7.3 Dual Operation

Dual operations can be used when the combination of the flash memory for programming/erasure and the flash memory for programming / erasing is any of those listed below.

Table 41.10 Conditions under which Dual Operation is Usable

Range for Programming / Erasure	Range for Programming / Erasure
The-bank of Data Flash Memory for ICUM	Other bank of Data Flash Memory

41.8 Reading Flash Memory

41.8.1 Reading Code Flash Memory

Special settings are not required to read code flash memory in normal mode and user boot mode. Data can simply be read out through access to addresses in the code flash memory.

Reading from an area of code flash memory that has been erased but not yet been programmed again (i.e. that is in the non-programmed state) can lead to the detection of an ECC error and generation of the corresponding exception. Furthermore, since the values of data cannot be guaranteed when an ECC error has been generated, use blank checking when you need to confirm that an area is in the non-programmed state. See **Section 38, Functional Safety** for the details on ECC.

41.8.2 Parallel Access to the Code Flash memory

The code flash memory is multi banked configuration. One bus master may access one bank and another bus master access other bank without a wait for arbitration of contention for the access path.

When multiple bus masters access the same bank access is arbitrated in round-robin fashion.

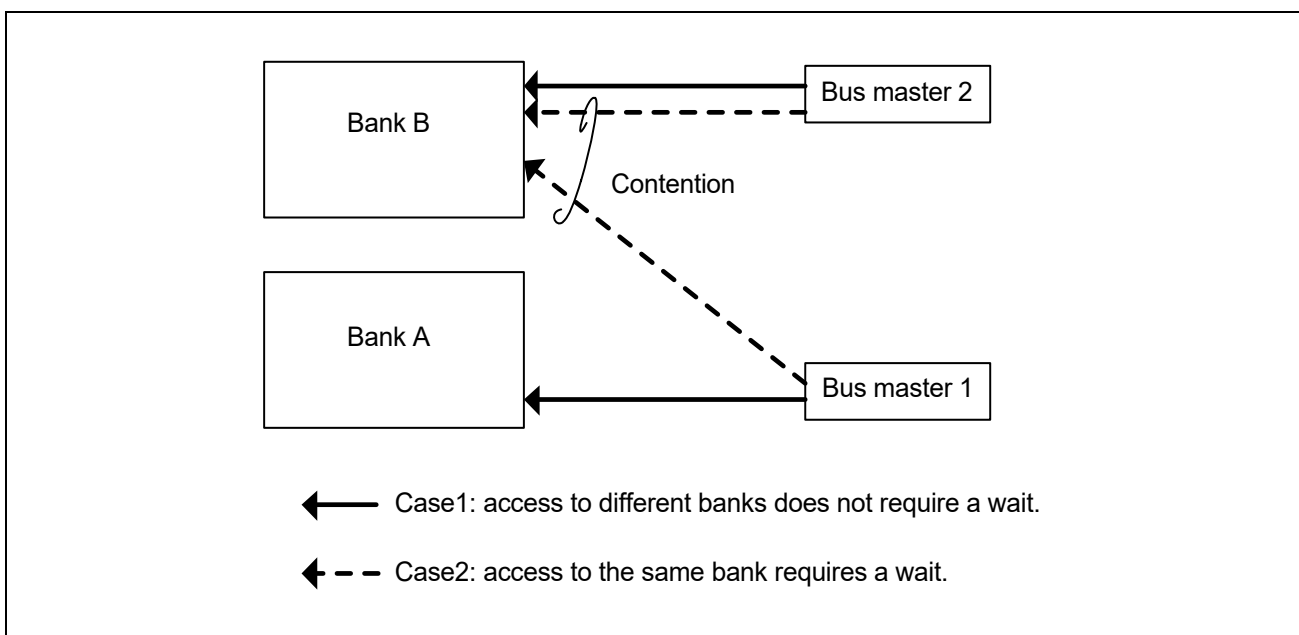


Figure 41.9 Operation in Access to Banks A and B (Example)

41.8.3 Reading Data Flash Memory

Values read from data flash memory that has been erased but not yet been programming again are undefined. Use blank checking when you need to confirm that an area is in the non-programmed state.

41.9 Description of Registers

41.9.1 Register Base Address

Register base addresses are listed in the following table.

Register addresses are given as offsets from the base addresses in general.

Table 41.11 Register Base Address

Base Address Name	Base Address	Bus Group
<SYSCTRL_base>	FF70 0000 _H	Peripheral Group 6

41.9.2 Registers Related to Write and Erase Protection of Flash Memory

Table 41.12 shows the list of registers related to write and erase protection of flash memory.

Table 41.12 Registers Related to Write and Erase Protection of Flash Memory

Unit Name	Register Name	Symbol	Address	Access Size
SYSCTRL	FHVE15 Control Register	FHVE15	<SYSCTRL_base> + 3804 _H	32
SYSCTRL	FHVE3 Control Register	FHVE3	<SYSCTRL_base> + 3800 _H	32

Table 41.13 Register Reset Conditions

Symbol	Reset Source					
	Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
FHVE15	✓	✓	✓	✓	—	—
FHVE3	✓	✓	✓	✓	—	—

41.9.2.1 FHVE15 — FHVE15 Control Register

FHVE15 is a readable / writable register for software protection of flash memory against programming and erasure. Set both the FHVE15 and FHVE3 registers in the programmable and erasable state (0000 0001_H) to program or erase flash memory.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FHVE 15CNT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 41.14 FHVE15 Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved
0	FHVE15CNT	0: Programming / erasure / DataFlashBlankChecking are disabled. 1: Programming / erasure / DataFlashBlankChecking are enabled.

41.9.2.2 FHVE3 — FHVE3 Control Register

FHVE3 is a readable / writable register for software protection of flash memory against programming and erasure. Set both FHVE15 and FHVE3 in the programmable and erasable state (0000 0001_H) to program or erase flash memory.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FHVE3 CNT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 41.15 FHVE3 Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved
0	FHVE3CNT	0: Programming / erasure / DataFlashBlankChecking are disabled. 1: Programming / erasure / DataFlashBlankChecking are enabled.

41.9.3 Registers Related to Product Information

In Product Info Area in **Figure 41.1**, Product information (Product Name) is stored. **Table 41.16** shows the list of registers related to product information.

Other addresses in this domain are reservations. Please do not access it.

Table 41.16 List of Registers Related to Product Information

Register Name	Symbol	R/W	Value after Reset	Address	Access Size
Product Number / Product Version Storage Register	PRDNUM	R	See Table 41.19	0803 0040 _H	32
Product Name Storage Register (1)	PRDNAME1	R	See Table 41.17	0803 00A0 _H	32
Product Name Storage Register (2)	PRDNAME2	R	See Table 41.17	0803 00A4 _H	32
Product Name Storage Register (3)	PRDNAME3	R	See Table 41.17	0803 00A8 _H	32
Product Name Storage Register (4)	PRDNAME4	R	See Table 41.17	0803 00AC _H	32

Table 41.17 Relationship between Product Model Name and Value of PRDNAME

Product Name	PRDNAME4	PRDNAME3	PRDNAME2	PRDNAME1
R7F702Z02C	2020 2020 _H	2020 2032 _H	305A 3230 _H	3746 3752 _H
R7F702Z04C	2020 2020 _H	2020 2034 _H	305A 3230 _H	3746 3752 _H
R7F702002A	2020 2020 _H	2020 2032 _H	3030 3230 _H	3746 3752 _H

Note: For Product Name, refer to **Section 1.4, Order Information**.

41.9.3.1 PRDNAME_n; n = 1 to 4 — Product Name Storage Register (n = 1 to 4)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PRDNAME _n [31:16] (n = 1 to 4)															
Value after reset	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PRDNAME _n [15:0] (n = 1 to 4)															
Value after reset	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. This is the value indicated in **Table 41.17**.

Table 41.18 List of Registers Related to Product Information

Bit Position	Bit Name	Function
31 to 0	PRDNAME _n [31:0]	Product name: Indicate product model names by using 16-byte ASCII code. PRDNAME1[31:0]: Fourth to first bytes of product model name PRDNAME2[31:0]: Eighth to fifth bytes of product model name PRDNAME3[31:0]: Twelfth to ninth bytes of the product model name PRDNAME4[31:0]: Sixteenth to thirteenth bytes of the product model name

41.9.3.2 PRDNUM — Product Number / Product Version Storage Register

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	DID_RN[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0/1*	0/1*	0/1*	0/1*
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DID_PN[15:0]															
Value after reset	0/1*	0/1*	0/1*	0/1*	0/1*	0/1*	0/1*	0/1*	0/1*	0/1*	0/1*	0/1*	0/1*	0/1*	0/1*	0/1*
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The value is dependent on products.

Table 41.19 List of Registers Related to Product Number / Product Version Storage Register

Bit Position	Bit Name	Function
31 to 20	—	Reserved
19 to 16	DID_RN[3:0]	Revision Number These bits indicate product revision The value is the same as SDID[31:28] (See Section 46, Boundary Scan).
15 to 0	DID_PN[15:0]	Product Number These bits indicate product number. The value is the same as SDID[27:12] (See Section 46, Boundary Scan).

41.10 Configuration Setting Area (Option Bytes, Reset Vector)

The flash memory has the extended Extra area (option bytes) to store data specified by the user for various purposes. Changes in settings such as initial setting of peripheral functions using option bytes become effective after release from the reset state. (Refer to **Section 10.3.4, FACL Reset Transfer**.) OTP setting of Configuration setting Area becomes effective without reset. **Table 41.20** shows option byte setting area. For setting and reading option bytes, see *the Renesas Flash Programmer Flash Programming Software User's Manual*, or *the RH850/E2x-FCC1 Flash Memory User's Manual: Hardware Interface*.

Table 41.20 Setting Area of Reset Vector and Option Bytes

Name	Address	Initial State of Shipped Product*1	Write Protection ID*2	Read Protection ID*3
OTP Setting of Configuration setting Area*4,*6	FF32 0040 _H	FFFF FFFF _H	—	—
Reserved	FF32 0044 _H to FF32 005C _H	FFFF FFFF _H *5	—	—
Reset Vector	FF32 0060 _H	0000 0000 _H	Customer ID	—
Reserved	FF32 0064 _H to FF32 007C _H	FFFF FFFF _H *5	Customer ID	—
Option bytes 3 to 0 (OPBT0)	FF32 0080 _H	3FFF 0010 _H	Customer ID	—
Option bytes 7 to 4 (OPBT1)	FF32 0084 _H	FFFF 0000 _H	Customer ID	—
Option bytes 11 to 8 (OPBT2)	FF32 0088 _H	FFFF FFFF _H	Customer ID	—
Option bytes 15 to 12 (OPBT3)	FF32 008C _H	FDFE FFFF _H	Customer ID	—
Option bytes 19 to 16 (OPBT4)	FF32 0090 _H	FFFF FFFF _H	Customer ID	—
Option bytes 23 to 20 (OPBT5)	FF32 0094 _H	FFFF EFFF _H	Customer ID	—
Option bytes 27 to 24 (OPBT6)	FF32 0098 _H	FFFF FFEB _H	Customer ID	—
Option bytes 31 to 28 (OPBT7)	FF32 009C _H	FFFF 899F _H	Customer ID	—
Option bytes 35 to 32 (OPBT8)	FF32 00A0 _H	FFFF FFFF _H	Customer ID	—
Option bytes 39 to 36 (OPBT9) Reserved	FF32 00A4 _H	FFFF FFF* _H *5	Customer ID	—
Option bytes 43 to 40 (OPBT10) Reserved	FF32 00A8 _H	FFFF FFFF _H *5	Customer ID	—
Option bytes 47 to 44 (OPBT11) Reserved	FF32 00AC _H	FFFF FFFF _H *5	Customer ID	—
Option bytes 51 to 48 (OPBT12) Reserved	FF32 00B0 _H	6000 0055 _H *5	Customer ID	—
Option bytes 55 to 52 (OPBT13) Reserved	FF32 00B4 _H	6000 0056 _H *5	Customer ID	—
Option bytes 59 to 56 (OPBT14) Reserved	FF32 00B8 _H	F8DC 0101 _H *5	Customer ID	—
Option bytes 63 to 60 (OPBT15) Reserved	FF32 00BC _H	258C 0401 _H *5	Customer ID	—
Reserved	FF32 00C0 _H to FF32 00FC _H	FFFF FFFF _H *5	—	—

Note 1. This is the initial value of the shipped product. The value can be modified.

Note 2. When all bits of Customer ID are all "1" or all "0", the ID authentication is no needed to set.

Note 3. When Sub mode is OCD mode, the Customer ID authentication is needed to read.

Note 4. The detail of this function, see **Section 42, Basic Hardware Protection**.

(**Table 42.9** shows **Mapping of OTP Setting for extra Area**.)

Note 5. It is prohibited to change from a read value.

Note 6. Set OTP after programming target data, because OTP setting becomes effective without reset.

41.10.1 Reset Vector PE

Customer ID authentication is needed for write this option byte. When all bits of Customer ID are all “1” or all “0”, the ID authentication is no needed to change the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	OPT_RBASE[22:16]						
Value after reset	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OPT_RBASE[15:9]							—	—	—	—	—	—	—	—	—
Value after reset	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by Configuration Programming.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 41.21 Reset Vector Contents

Bit Position	Bit Name	Function
31 to 23	—	Reserved (The setting value should be 0.)
22 to 9	OPT_RBASE[22:9]	These bits indicate the reset vector when there is a reset. Any place of code flash address can be configured at Normal Operation mode.
8 to 0	—	Reserved (The setting value should be 0.)

41.10.2 OPBT0 — Option Byte 3 to 0 Bit Arrangement

Customer ID authentication is needed for write this option byte. When all bits of Customer ID are all “1” or all “0”, the ID authentication is no needed to change the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OPWD RUN	OPWD WMS	OPWD OVF2	OPWD OVF1	OPWD OVF0	OPWD INT	—	OPWD WS1	OPWD WS0	—	—	—	—	—	—	—
Value after reset	0*1	0*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OPWDWOST[15:0]															
Value after reset	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	1*1	0*1	0*1	0*1	0*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 41.22 OPBT0 Contents

Bit Position	Bit Name	Function																																				
31	OPWDRUN	This bit sets the start mode of WDTB0. 0: WDTB0 software trigger start mode 1: WDTB0 default start mode																																				
30	OPWDWMS	0: Window Size of Window Open function is set by WDTB0WS1-0. WDTB0TIT outputs When the counter reaches 75% of the overflow setting defined by WDTB0MD.WDT0OVF2-0 1: Window Size of Window Open function is set by WDTB0WOST WDTB0TIT outputs When t WDTB0WIS matches WDT counter.																																				
29 to 27	OPWDOVF2 to OPWDOVF0	These bits select the overflow interval time of WDTB0. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>OPWDOVF2</th><th>OPWDOVF1</th><th>OPWDOVF0</th><th>Overflow Interval Time</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>2⁹ / WDTBTCKI</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>2¹⁰ / WDTBTCKI</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>2¹¹ / WDTBTCKI</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>2¹² / WDTBTCKI</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>2¹³ / WDTBTCKI</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>2¹⁴ / WDTBTCKI</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>2¹⁵ / WDTBTCKI</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>2¹⁶ / WDTBTCKI</td></tr> </tbody> </table>	OPWDOVF2	OPWDOVF1	OPWDOVF0	Overflow Interval Time	0	0	0	2 ⁹ / WDTBTCKI	0	0	1	2 ¹⁰ / WDTBTCKI	0	1	0	2 ¹¹ / WDTBTCKI	0	1	1	2 ¹² / WDTBTCKI	1	0	0	2 ¹³ / WDTBTCKI	1	0	1	2 ¹⁴ / WDTBTCKI	1	1	0	2 ¹⁵ / WDTBTCKI	1	1	1	2 ¹⁶ / WDTBTCKI
OPWDOVF2	OPWDOVF1	OPWDOVF0	Overflow Interval Time																																			
0	0	0	2 ⁹ / WDTBTCKI																																			
0	0	1	2 ¹⁰ / WDTBTCKI																																			
0	1	0	2 ¹¹ / WDTBTCKI																																			
0	1	1	2 ¹² / WDTBTCKI																																			
1	0	0	2 ¹³ / WDTBTCKI																																			
1	0	1	2 ¹⁴ / WDTBTCKI																																			
1	1	0	2 ¹⁵ / WDTBTCKI																																			
1	1	1	2 ¹⁶ / WDTBTCKI																																			
26	OPWDINT	This bit enables or disables a 75% interrupt request of WDTB0 (WDTB0TIT). 0: WDTB0TIT disabled 1: WDTB0TIT enabled																																				
25	—	Reserved (The setting value should be 1.)																																				
24, 23	OPWDWS1, OPWDWS0	These bits select the window-open period of WDTB0. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>OPWDWS1</th><th>OPWDWS0</th><th>Window-Open Period</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>25%</td></tr> <tr><td>0</td><td>1</td><td>50%</td></tr> <tr><td>1</td><td>0</td><td>75%</td></tr> <tr><td>1</td><td>1</td><td>100%</td></tr> </tbody> </table>	OPWDWS1	OPWDWS0	Window-Open Period	0	0	25%	0	1	50%	1	0	75%	1	1	100%																					
OPWDWS1	OPWDWS0	Window-Open Period																																				
0	0	25%																																				
0	1	50%																																				
1	0	75%																																				
1	1	100%																																				
22 to 16	—	Reserved (The setting value should be 1.)																																				
15 to 0	OPWDWOST	This OPBT[15:0] is Window Open Start register for setting the start timing of the window open.																																				

41.10.3 OPBT1 — Option Byte 7 to 4 Bit Arrangement

Customer ID authentication is needed for write this option byte. When all bits of Customer ID are all “1” or all “0”, the ID authentication is no needed to change the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WDT_CLK_SEL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OPDWWIS [15:0]															
Value after reset	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 41.23 OPBT1 Contents

Bit Position	Bit Name	Function
31	WDT_CLK_SEL	WDTB/SWDT counter clock source can be selected by using the option bytes. 0: CLK_MOSC source is selected 1: CLK_IOSC source is selected
30 to 16	—	Reserved (The setting value should be 1.)
15 to 0	OPDWWIS	This OPBT[15:0] is WDT Interrupt Output Timing Setting Register

41.10.4 OPBT2 — Option Byte 11 to 8 Bits Arrangement

Customer ID authentication is needed for write this option byte. When all bits of Customer ID are all “1” or all “0”, the ID authentication is no needed to change the setting.

For details of this OPBT2, see *the RH850/E2x SWDT User's Manual*.

41.10.5 OPBT3 — Option Byte 15 to 12 Bits Arrangement

Customer ID authentication is needed for write this option byte. When all bits of Customer ID are all “1” or all “0”, the ID authentication is no needed to change the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	PE1_DISABLE	STARTUPPE	—	—	—	—	—	—	PE1_FP_SIMD_EN	PE0_FP_SIMD_EN
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	0*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HWBIST	—	—	—	TESTSET[1:0]	LBISTSEL[1:0]	—	—	—	—	—	—	—	—	STMSEL1	STMSEL0
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 41.24 OPBT3 Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	—	Reserved (The setting value should be 1.)
25	PE1_DISABLE	PE1_DISABLE bit 0: PE1 enable 1: PE1 disable
24	STARTUPPE	select whether PE0 start 0: PE0 not run (when the ICUMD is valid only.) 1: PE0 run
23 to 18	—	Reserved (The setting value should be 1.)
17	PE1_FP_SIMD_EN	PE1_FP-SIMD Enable bit 0: PE1_FP_SIMD disable 1: PE1_FP_SIMD enable
16	PE0_FP_SIMD_EN	PE0_FP-SIMD Enable bit 0: PE0_FP_SIMD disable 1: PE0_FP_SIMD enable
15	HWBIST	Power-up BIST enable 0: BIST is skipped 1: BIST is executed
14 to 12	—	Reserved (The setting value should be 1.)
11,10	TESTSET[1:0]	Power-up BIST selection 0 0 _B : Prohibited 0 1 _B : LBIST Only 1 0 _B : MBIST Only 1 1 _B : LBIST and MBIST For BIST, four operation settings can be selected by combining TESTSET[1:0] and LBISTSEL[1:0]. Refer to Table 41.25, BIST Scenario Use Case.

Table 41.24 OPBT3 Contents (2/2)

Bit Position	Bit Name	Function
9,8	LBISTSEL[1:0]	At Power-up BIST scenario selection 00 _B : BIST scenario 1 01 _B : BIST scenario 2 10 _B : BIST scenario 3 11 _B : BIST scenario 4 For BIST, four operation settings can be selected by combining TESTSET[1:0] and LBISTSEL[1:0]. Refer to Table 41.25, BIST Scenario Use Case .
7 to 2	—	Reserved (The setting value should be 1.)
1, 0	STMSEL1, STMSEL0	These bits select operating mode and startup area. When all of the MD0 and MD1 pins are 0, the following operating mode and startup area are selected depending on the combination of the STMSEL1 and STMSEL0. For details, see Section 5, Operating Modes .

Table 41.25 BIST Scenario Use Case

LBISTSEL[1:0]* ¹	TESTSET[1:0]* ¹	Function	Run time* ²	Target ASIL* ³
00	10	MBIST Execution	5ms (MBIST 5ms)	—
01	11	LBIST and MBIST Execution	10ms (LBIST 5ms, MBIST 5ms)	ASIL B
10	01	LBIST Execution	15ms (LBIST 15ms)	ASIL D
11	11	LBIST and MBIST Execution	20ms (LBIST 15ms, MBIST 5ms)	ASIL D

Note 1. Settings other than above are prohibited.

Note 2. Run time is calculated by the typical frequency of the Internal OSC. For details, see *the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware*.

Note 3. ISO26262 compliance for LFM by LBIST.

41.10.6 OPBT4 — Option Byte 19 to 16 Bit Arrangement

Customer ID authentication is needed for write this option byte. When all bits of Customer ID are all “1” or all “0”, the ID authentication is no needed to change the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VDDFLT W [1:0]	VDDCLKSEL [1:0]	VDDFL TEN	—	VDDHD E	VDDL D E	VCCFLT W [1:0]	VCCCLKSEL [1:0]	VCCFL TEN	—	VCCHD E	—				
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EVCCFLT W [1:0]	EVCCCLKSEL [1:0]	EVCCFL TEN	—	EVCCCH DE	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 41.26 OPBT4 Contents (1/3)

Bit Position	Bit Name	Function										
31, 30	VDDFLT[W:1]	VMON configurations for VDD Select the minimum filtering width of digital noise filter.*1 (The minimum filtering width is given by cycle-count of the clock selected by VDDCLKSEL1, 0.) <table border="1"> <thead> <tr> <th>VDDFLT[W:1]</th> <th>Minimum Filtering Width (Cycle)</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>20 cycles</td> </tr> <tr> <td>01</td> <td>13 cycles</td> </tr> <tr> <td>10</td> <td>8 cycles</td> </tr> <tr> <td>11</td> <td>3 cycle</td> </tr> </tbody> </table>	VDDFLT[W:1]	Minimum Filtering Width (Cycle)	00	20 cycles	01	13 cycles	10	8 cycles	11	3 cycle
VDDFLT[W:1]	Minimum Filtering Width (Cycle)											
00	20 cycles											
01	13 cycles											
10	8 cycles											
11	3 cycle											
29, 28	VDDCLKSEL[1:0]	VMON configurations for VDD Select the clock of the digital noise filter from the following clocks.*1 <table border="1"> <thead> <tr> <th>VDDCLKSEL[1:0]</th> <th>Clock of the Digital Filter</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>CLK_HVIOSC frequency / 256</td> </tr> <tr> <td>01</td> <td>CLK_HVIOSC frequency / 128</td> </tr> <tr> <td>10</td> <td>CLK_HVIOSC frequency / 64</td> </tr> <tr> <td>11</td> <td>CLK_HVIOSC frequency / 32</td> </tr> </tbody> </table> <p>CAUTION: The frequency of CLK_HVIOSC is typ. 16 MHz, min. 8 MHz.</p>	VDDCLKSEL[1:0]	Clock of the Digital Filter	00	CLK_HVIOSC frequency / 256	01	CLK_HVIOSC frequency / 128	10	CLK_HVIOSC frequency / 64	11	CLK_HVIOSC frequency / 32
VDDCLKSEL[1:0]	Clock of the Digital Filter											
00	CLK_HVIOSC frequency / 256											
01	CLK_HVIOSC frequency / 128											
10	CLK_HVIOSC frequency / 64											
11	CLK_HVIOSC frequency / 32											
27	VDDFLTEN	VMON configurations for VDD Enable output filter for ERROROUT_M and VMONF.*1 0: Disable output filter for ERROROUT_M and VMONF. 1: Enable output filter for ERROROUT_M and VMONF.										
26	—	Reserved (The setting value should be 1.)										
25	VDDHDE	VMON configurations for VDD VDD High voltage detection enable.*1 0: Disable VDD high voltage detection 1: Enable VDD high voltage detection										
24	VDDLDE	VMON configurations for VDD VDD Low voltage detection enable.*1 0: Disable VDD low voltage detection 1: Enable VDD low voltage detection										

Table 41.26 OPBT4 Contents (2/3)

Bit Position	Bit Name	Function
23, 22	VCCFLTW[1:0]	VMON configurations for VCC Select the minimum filtering width of digital noise filter.*1 (The minimum filtering width is given by the cycle-count of the clock selected by VCCCLKSEL1, 0.)
	VCCFLTW[1:0]	Minimum Filtering Width (Cycle)
	00	20 cycles
	01	13 cycles
	10	8 cycles
	11	3 cycles
21, 20	VCCCLKSEL[1:0]	VMON configurations for VCC Select the clock of the digital noise filter from the following clocks.*1
	VCCCLKSEL[1:0]	Clock of the Digital Filter
	00	CLK_HVIOSC frequency / 256
	01	CLK_HVIOSC frequency / 128
	10	CLK_HVIOSC frequency / 64
	11	CLK_HVIOSC frequency / 32
		CAUTION: The frequency of CLK_HVIOSC is typ. 16 MHz, min. 8 MHz.
19	VCCFLTEN	VMON configurations for VCC Enable output filter for $\overline{\text{ERROROUT_M}}$ and VMONF.*1 0: Disable output filter for $\overline{\text{ERROROUT_M}}$ and VMONF. 1: Enable output filter for $\overline{\text{ERROROUT_M}}$ and VMONF.
18	—	Reserved (The setting value should be 1.)
17	VCCHDE	VMON configurations for VCC VCC High voltage detection enable.*1 0: Disable VCC high voltage detection 1: Enable VCC high voltage detection
16	—	Reserved (The setting value should be 1.)
15, 14	EVCCFLTW[1:0]	VMON configurations for EVCC Select the minimum filtering width of digital noise filter.*1 (The minimum filtering width is given by the cycle-count of the clock selected by EVCCCLKSEL1, 0)
	EVCCFLTW[1:0]	Minimum Filtering Width (Cycle)
	00	20 cycles
	01	13 cycles
	10	8 cycles
	11	3 cycles
13, 12	EVCCCLKSEL[1:0]	VMON configurations for EVCC Select the clock of the digital noise filter from the following clocks.*1
	EVCCCLKSEL[1:0]	Clock of the Digital Filter
	00	CLK_HVIOSC frequency / 256
	01	CLK_HVIOSC frequency / 128
	10	CLK_HVIOSC frequency / 64
	11	CLK_HVIOSC frequency / 32
		CAUTION: The frequency of CLK_HVIOSC is typ. 16 MHz, min. 8 MHz.
11	EVCCFLTEN	VMON configurations for EVCC Enable output filter for $\overline{\text{ERROROUT_M}}$ and VMONF.*1 0: Disable output filter for $\overline{\text{ERROROUT_M}}$ and VMONF. 1: Enable output filter for $\overline{\text{ERROROUT_M}}$ and VMONF.

Table 41.26 OPBT4 Contents (3/3)

Bit Position	Bit Name	Function
10	—	Reserved (The setting value should be 1.)
9	EVCCHDE	VMON configurations for EVCC EVCC High voltage detection enable.*1 0: Disable EVCC high voltage detection 1: Enable EVCC high voltage detection
8 to 0	—	Reserved (The setting value should be 1.)

Note 1. For details, see **Section 12.2, Primary Detection of Voltage Monitor (VMON)**.

41.10.7 OPBT5 — Option Byte 23 to 20 Bit Arrangement

Customer ID authentication is needed for write this option byte. When all bits of Customer ID are all “1” or all “0”, the ID authentication is no needed to change the setting.

For details of this OPBT5, see *the RH850/E2x ICUMD User’s Manual*.

41.10.8 OPBT6 — Option Byte 27 to 24 Bit Arrangement

Customer ID authentication is needed for write this option byte. When all bits of Customer ID are all “1” or all “0”, the ID authentication is no needed to change the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	STAC_DTSRAM [1:0]	STAC_DPRAM [1:0]	STAC_DFE [1:0]	STAC_FLXA [1:0]	STAC_ETHER [1:0]	STAC_GTM [1:0]						
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	0*1	1*1	0*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 41.27 OPBT6 Contents

Bit Position	Bit Name	Function
31 to 12	—	Reserved (The setting value should be 1.)
11, 10	STAC_DTSRAM[1:0]	RAM Initialization Mode for DTS RAM*1 X 0 _B : Disabled 0 1 _B : Prohibited 1 1 _B : Enabled
9, 8	STAC_DPRAM[1:0]	RAM Initialization Mode for sDMAC Descriptor RAM*1 X 0 _B : Disabled 0 1 _B : Prohibited 1 1 _B : Enabled
7, 6	STAC_DFE[1:0]	RAM Initialization Mode for DFE*1 X 0 _B : Disabled 0 1 _B : Prohibited 1 1 _B : Enabled
5, 4	STAC_FLXA[1:0]	RAM Initialization Mode for FlexRay*1 X 0 _B : Disabled 0 1 _B : Prohibited 1 1 _B : Enabled
3, 2	STAC_ETHER[1:0]	RAM Initialization Mode for Ethernet*1 X 0 _B : Disabled 0 1 _B : Prohibited 1 1 _B : Enabled
1, 0	STAC_GTM[1:0]	RAM Initialization Mode for GTM*1 X 0 _B : Disabled 0 1 _B : Prohibited 1 1 _B : Enabled

Note 1. For details, see **Section 10.3.6, RAM Initialization**.

41.10.9 OPBT7 — Option Byte 31 to 28 Bit Arrangement

Customer ID authentication is needed for write this option byte. When all bits of Customer ID are all “1” or all “0”, the ID authentication is no needed to change the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MOSC_40MHz	—	—	MOSC_EXCLK INPUTZ	—	—	—	—	—	MOSC_AMP_SEL_A[2:0]			—	MOSC_AMP_SEL_B[2:0]		
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOSC_CAP_SEL[3:0]			—	MOSC_RD_SEL_A[2:0]			—	MOSC_RD_SEL_B[2:0]			—	—	MOSC_SHTS TBY_A	MOSC_SHTS TBY_B	
Value after reset	1*1	0*1	0*1	0*1	1*1	0*1	0*1	1*1	1*1	0*1	0*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 41.28 OPBT7 Contents

Bit Position	Bit Name	Function
31	MOSC_40MHz	Main oscillation frequency selection bit 0: 20 MHz 1: 40 MHz
30, 29	—	Reserved (The setting value should be 1.)
28	MOSC_EXCLK INPUTZ	MOSC input clock select. EXCLK mode(for EXCLK/FRCUT control) 0: MOSC outputs the clock from EXTAL as it be that does not go through AMP. 1: Normal oscillation
27 to 23	—	Reserved (The setting value should be 1.)
22 to 20	MOSC_AMP_SEL_A [2:0]	MOSC trimming configuration This bit controls OSC drivability when oscillation is not stabilized.*1
19	—	Reserved (The setting value should be 1.)
18 to 16	MOSC_AMP_SEL_B [2:0]	MOSC trimming configuration This bit controls OSC drivability when oscillation is stabilized.*1
15 to 12	MOSC_CAP_SEL [3:0]	MOSC trimming configuration This bit controls internal capacitance. ¹
11	—	Reserved (The setting value should be 1.)*
10 to 8	MOSC_RD_SEL_A [2:0]	MOSC trimming configuration This bit controls Damping resistor when oscillation is not stabilized.*1
7	—	Reserved (The setting value should be 1.)
6 to 4	MOSC_RD_SEL_B [2:0]	MOSC trimming configuration This bit controls Damping resistor when oscillation is stabilized.*1
3, 2	—	Reserved (The setting value should be 1.)
1	MOSC_SHTSTBY_A	MOSC trimming configuration This bit controls OSC drivability when oscillation is not stabilized.*1 MOSC_SHTSTBY_A must be set to 1.
0	MOSC_SHTSTBY_B	MOSC trimming configuration This bit controls OSC drivability when oscillation is stabilized.*1

Note 1. For details, see Section 1.3.4, Clock Timing, in the RH850/E2x-FCC1 Electrical Characteristics, User's Manual: Hardware.

41.10.10 OPBT8 — Option Byte 35 to 32 Bit Arrangement

Customer ID authentication is needed for write this option byte. When all bits of Customer ID are all “1” or all “0”, the ID authentication is no needed to change the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	GTM_RAM_HALF_MODE	—	—	ETN_RMII_SEL	ATU_GTM_SEL	—	—	—	—	—	—	RHSB_FRQ1	RHSB_FRQ0
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 41.29 OPBT8 Contents

Bit Position	Bit Name	Function
31 to 29	—	Reserved (The setting value should be 1.)
28	GTM_RAM_HALF_MODE	GTM MCS RAM mode select 0: GTM MCS_RAM E2x-FCC1 mode 1: GTM MCS_RAM PD mode
27 to 26	—	Reserved (The setting value should be 1.)
25	ETN_RMII_SEL	Ether IF MII/RMII select 0: MII 1: RMII
24	ATU_GTM_SEL	ATU, GTM select 0: ATU: disable, GTM: enable 1: ATU: enable, GTM: disable
23 to 18	—	Reserved (The setting value should be 1.)
17	RHSB_FRQ1	RHSB1 frequency setting bit 0: CLK_UHSB (160 MHz) 1: CLK_HSB (80 MHz)
16	RHSB_FRQ0	RHSB0 frequency setting bit 0: CLK_UHSB (160 MHz) 1: CLK_HSB (80 MHz)
15 to 0	—	Reserved (The setting value should be 1.)

41.11 Block Protection Area

The flash memory has the extended Extra area to store a data specified by the user for Code Flash OTP Settings and Code Flash Customer ID Protection Settings. These settings become effective without reset.

The detail of this function, see **Section 42, Basic Hardware Protection**.

Table 41.30 Block Protection Area

Name	Address	Initial State of Shipped Product.*1	Write/Erase Protection ID*2	Read Protection ID*3
OTP Setting of User/User Boot Area*4	FF32 2040 _H to FF32 2050 _H	FFFF FFFF _H	—	—
Reserved	FF32 2054 _H to FF32 207C _H	FFFF FFFF _H *6	—	—
Customer ID Protection Setting of User/User Boot Area*5	FF32 2080 _H to FF32 20C0 _H	FFFF FFFF _H	Customer ID	—
Reserved	FF32 20C4 _H to FF32 20FC _H	FFFF FFFF _H *6	—	—

Note 1. This is the initial value of the shipped product. The value can be modified.

Note 2. When all bits of Customer ID are all "1" or all "0", the ID authentication is no needed to set.

Note 3. When Sub mode is OCD mode, the Customer ID authentication is needed to read.

Note 4. The detail of this function, see **Section 42, Basic Hardware Protection**.

Table 42.7 shows **Mapping of OTP Setting for User Area and User Boot Area**.

Note 5. The detail of this function, see **Section 42, Basic Hardware Protection**.

Figure 42.2 Block Protection Function Structure and **Table 42.6** shows **Mapping of Block Protection Setting**.

Note 6. It is prohibited to change from a read value.

41.12 Usage Notes

- (1) Reading areas where programming or erasure was interrupted
When programming or erasure of an area of flash memory is interrupted intentionally or unintentionally, the data stored in the area become undefined. To avoid undefined data that are read out becoming the source of faulty operation, take care not to fetch instructions or read data from areas where programming or erasure was interrupted.
- (2) Reading the code flash memory that has been erased but not yet been programming again
Note that reading from an area of code flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state) can lead to the detection of an ECC error and generation of the corresponding exception. Use blank checking when you need to confirm that an area is in the non-programmed state.
- (3) Prohibition of additional writing
Writing to a given area twice is not possible. If you want to update data in an area of flash memory after writing to the area has been completed, erase the area first. (Erasure is unnecessary for rewriting of flash extra area.)
- (4) Resets during programming and erasure
In the case of an external reset during programming and erasure, wait for at least the minimum value of reset pulse width once the operating voltage is within the range stipulated in the electrical characteristics after assertion of the reset signal before releasing the device from the reset state.
- (5) Allocation of vectors for interrupts and other exceptions during programming and erasure
Generation of an interrupt or other exception during programming or erasure may lead to fetching of the vector from the code flash memory. If this does not satisfy the conditions for using background operation, set the address for vector fetching to an address that is not in the code flash memory.
- (6) Checking the areas where programming or erasure was interrupted
If programming or erasure is interrupted intentionally or unintentionally, the programming or erasure state of the flash memory with undefined data cannot be verified or checked. For the area where programming or erasure was interrupted, the blank check function cannot judge whether the area is erased successfully or not. Erase the area again to prove that the corresponding area is completely erased before using.
- (7) Items prohibited during programming / erasure / blank check
Do not perform the following operations while the flash memory is programmed, erased, or checked for blank.
 - Have the operating voltage from the power supply go beyond the allowed range.
 - Update the values of FHVE15 and FHVE3.
 - Change the operating frequency of peripheral clock.
- (8) Flash memory commands prohibited before completion of clock gear-up sequence
Various commands during serial programming and self-programming must be executed after clock gear-up sequence is complete. For details of clock gear-up sequence, see **Section 15, Clock Controller**.

(9) Securing coherency of instruction cache and data buffer

After the code flash memory is programmed or erased, a reset or instruction cache and data buffer clearing are required to secure coherency of the instruction cache and data buffer. For details of instruction cache and data buffer, see **Section 3, CPU System**.

(10) Setting of OTP.

Set OTP after programming target data, because OTP setting becomes effective without reset.

Please be careful about Configuration Setting Area and Security Settings Area where OTP setting address is smaller than the address of the target.

OTP setting becomes effective in the first place if programming the target data in the order of OTP setting of Security Settings Area, Security Settings Area, OTP setting of Configuration Setting Area, and Configuration Setting Area. Therefore, the target data of Configuration Setting Area and Security Settings Area cannot be written.

Section 42 Basic Hardware Protection (BHP)

To protect the code flash memory, data flash memory and Flash extra area, this product has several Flash protection functions.

The security function that is used by the ID authentication and setting of Flash Option Byte is described in this section. The ID code and the Flash Option Byte that is described in this section are mapped at Security Settings area and Block Protection Settings area.

In the initial state of the product at shipment, the ID code are set by all - 1. These ID should be set at the first programming to Flash Option Byte. For detailed description of the programming to the flash memory, see **Section 41, Flash Memory**.

42.1 Features

ID code

- **Table 42.1** shows the ID code that can be set in this product.

The system requires each ID authentication for programming and reading the ID.

Table 42.1 ID Code

ID Code	Function
OCD ID	Protection for debug connection
Serial Programmer ID	Protection to enter the serial programming mode
RHSIF ID	Protection for RHSIF Link Partner
Customer ID	Protection for Code Flash and Flash extra area
Data Flash ID	Protection for Data Flash
C-TEST ID	Protection to enter the test mode

- The setting and function that related each ID code can select the Flash Option Byte. The system requires each ID authentication to re-programming these Flash Option Byte.

User/User boot/Data area protection by ID authentication

- The system requires 256 bit Customer ID authentication for re-programming to User area by each block.
- The Customer ID authentication is needed to read the User/User boot area when debugger is connected.
- The system requires 256 bit Data Flash ID authentication for re-programming to Data area.
- The Data Flash ID authentication is needed to read the Data area when debugger is connected.

User/User boot area protection by OTP

- User area on code flash supports OTP (One Time Programmable) functionality for each block.
- Flash extra area protection by OTP
- Dedicated configuration area supports OTP functionality for each 4 Byte unit.

42.2 Security Functions in Serial Programming Mode

Two functions are provided as security functions in serial programming mode: ID authentication and prohibition of serial programmer connection. Parallel use of these functions is not allowed.

1. Serial Programmer ID Authentication

The ID codes are checked for authenticity to protect the code flash memory, data flash memory and Flash extra area. Programming, erasure, and reading of the code flash and data flash memory can proceed upon successful ID authentication.

2. Prohibition of Serial Programmer Connection

Issuing commands for programming, erasure, or reading to the code flash memory, data flash memory and Flash extra area can be disabled in serial programming mode.

42.3 Security Functions in Debug Interfaces

This product is capable of restricting connection of the debug interfaces to protect against unauthorized access via the debug interfaces. This feature has two security levels.

- Security level 1:

Debug interface can be used. At this level, the on-chip debugging (OCD) function is protected by using OCD ID authentication. For OCD to be used, it must be unlocked by using OCD ID authentication.

- Security level 2:

At this level, the debug interface cannot be used.

These security levels can be changed by Flash Option Byte (JPDBGIF_EN) in Security Setting area of the flash memory.

When ICUMD is enabled, The C&R authentication is added to security function.

For details, see *the RH850/E2x ICUMD User's Manual*.

42.3.1 Security Levels and State of Restricting the Connection of Debug Interfaces

Table 42.2 show each security level and the corresponding security states and Figure 42.1 shows the conditions for transitions.

Table 42.2 Security Levels and State of Restricting Connection to the Debug Interfaces

State	Result of OCD ID Authentication	JPDBGIF_EN* ¹	Restriction on Debug Interface Connection
Security level 1	Unlocked	1 _B	No restriction on access via the debug interfaces
	Locked	1 _B	Restriction on access via the debug interfaces is in place.
Security level 2	—	0 _B	Connection to the debug interfaces is prohibited.

Note 1. For a detailed description of the JPDBGIF_EN bits, see Section 42.6.1, S_OPBT0 — OCD ID Related Option Bytes.

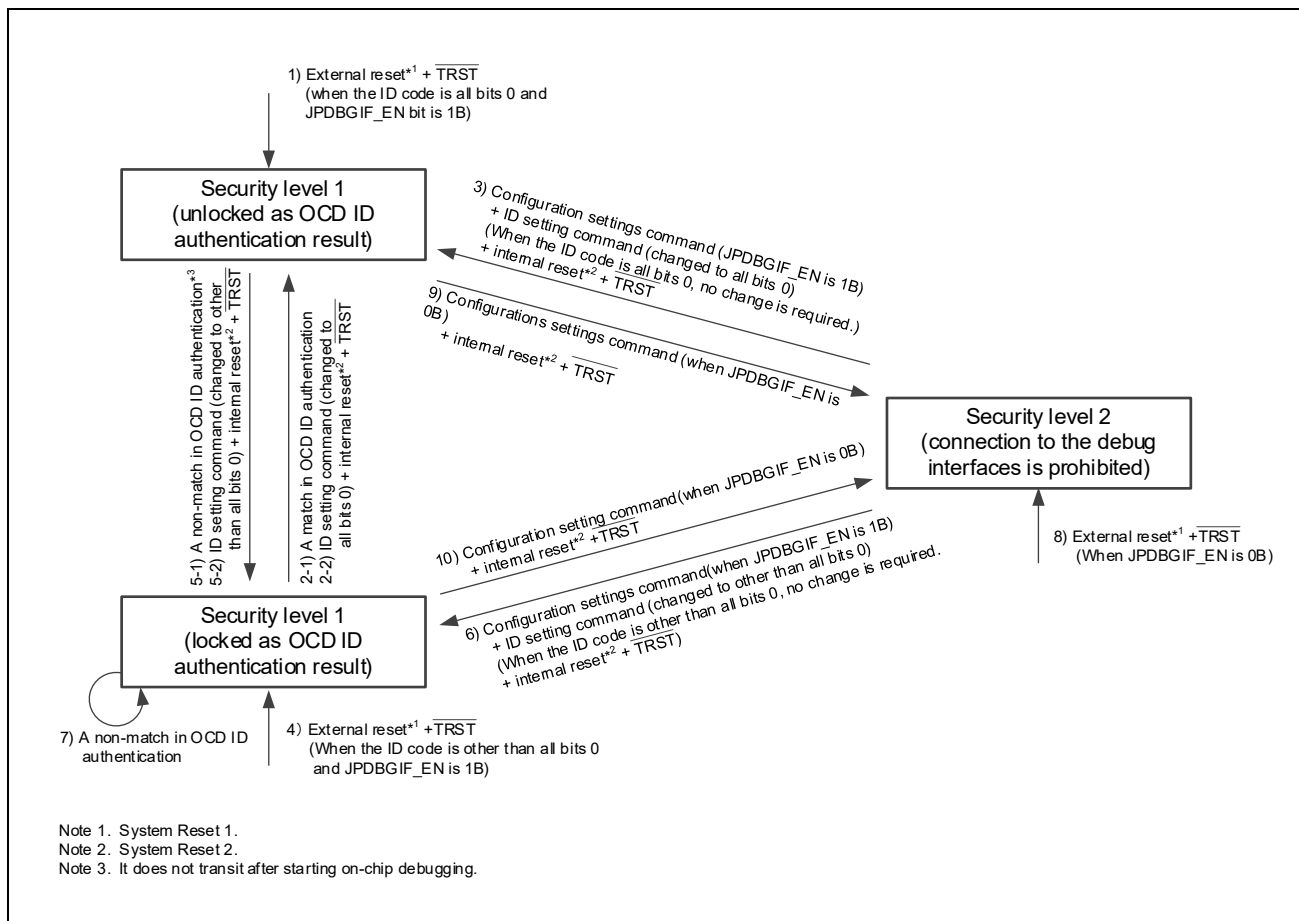


Figure 42.1 Transitions of Security Level

Conditions for transitions to each security level described in Table 42.2.

- Conditions for transitions to security level 1 (locked as OCD ID authentication result)

1. Startup in the state of security level 1 (unlocked as OCD ID authentication result)

When an external reset*¹ and a reset by TRST# are applied while the ID code is all 0s and while the value of the JPDBGIF_EN is 1_B, startup proceeds with security level 1 (unlocked as OCD ID authentication result).

2. Transition from security level 1 (locked as OCD ID authentication result) to security level 1 (unlocked as OCD ID authentication result)
 - Transition to security level 1 (unlocked as OCD ID authentication result) follows a match in OCD ID authentication.
 - When an internal reset*² and a reset by TRST# is applied after changing the ID code to “all bits 0”, startup proceeds at security level 1 (unlocked as OCD ID authentication result).
3. Transition from security level 2 to security level 1 (unlocked as OCD ID authentication result)

When an internal reset*² and a reset by TRST# is applied after changing the ID code to “all bits 0” and the value of the JPDBGIF_EN to 1_B, startup proceeds at security level 1 (unlocked as OCD ID authentication result).

 - Conditions for Transitions to Security Level 1 (locked as OCD ID Authentication result)
4. Startup in the state of security level 1 (locked as OCD ID authentication result)

When an external reset*¹ and a reset by TRST# is applied while the ID code is “other than all bits 0” and the value of the JPDBGIF_EN is 1_B, startup proceeds at security level 1 (locked OCD ID authentication result).
5. Transition from security level 1 (unlocked as OCD ID authentication result) to security level 1 (locked as OCD ID authentication result)
 - A transition to security level 1 (locked as OCD ID authentication result) follows a non-match in OCD ID authentication.*³
 - When an internal reset*² and a reset by TRST# is applied after changing the ID code to “other than all bits 0”, startup proceeds at security level 1 (locked as OCD ID authentication result).
6. Transition from security level 2 to security level 1 (locked as OCD ID authentication result)

When an internal reset*² and a reset by TRST# is applied after changing the ID code to “other than all bits 0” and the value of JPDBGIF_EN is 1_B, startup proceeds at security level 1 (locked as OCD ID authentication result).
7. Retaining security level 1 (locked as OCD ID authentication result)

In case of a non-match in OCD ID authentication, security level 1 (failure in OCD ID authentication) remains in place.

 - Transition to security level 2
8. Startup at security level 2

When an external reset*¹ and a reset by TRST# is applied while the value of JPDBGIF_EN is 0_B, startup proceeds at security level 2.
9. Transition from security level 1 (unlocked as OCD ID authentication result) to security level 2

When an internal reset*² and a reset by TRST# is applied after changing the value of the JPDBGIF_EN to 0_B, startup proceeds at security level 2.
10. Transition from security level 1 (locked as OCD ID authentication result) to security level 2

When an internal reset*² and a reset by TRST# is applied after changing the value of the JPDBGIF_EN to 0_B, startup proceeds at security level 2.

Note 1. System Reset 1.

Note 2. System Reset 2.

Note 3. It does not transit after starting on-chip debugging.

42.3.2 The Case of CAN BootStrap Mode

When the system is in CAN BootStrap Mode, debug interface can be used.

When user program is running, the security level depends on the setting of JPDBGIF_EN.

42.4 Security Functions in Connection of RHSIF Link Partner

Three functions are provided as security functions unique to RHSIF ID authentication, and prohibition of RHSIF Link Partner connection.

- RHSIF ID Authentication

ID codes are checked for authenticity to protect RHSIF Link Partner connection.

When ICUMD is enabled, The C&R authentication is added to security function.

For details, see *the RH850/E2x ICUMD User's Manual*.

- Access Window setting

User can set the access area from Link Partner that is configured the access window register. The access window register can not set from Link Partner.

The detail of this function, see **Section 27, Renesas High-Speed Serial I/F (RHSIF)**.

42.5 Code Flash / Data Flash Protection

Protection based on section password operates for non-secure area and doesn't affect the access from ICUMD. The section password is stored in Security settings area for each of code flash and data flash (Customer ID and Data Flash ID). User can unlock protection by inputting correct section password by software or external debugging tool.

Inputting target registers for software, see *the RH850/E2x-FCC1 Flash Memory User's Manual: Hardware Interface*.

Customer ID and Data Flash ID, see **Section 42.6.11, CUSTOMERIDn — Customer ID (n = 0 to 7)** and **Section 42.6.12, DATAFLASH IDn — Data Flash ID (n = 0 to 7)**.

The ID authentication for read access can be skipped by setting read protection flag in Security settings area for code and data flash respectively. (CFRPF and DFRPF)

Table 42.3 shows access protection to user area of Code flash memory. ID authentication to read access in Normal Operation mode/OCD mode is done via debugger I/F. ID authentication for reading access in serial programming mode is not needed. ID authentication for writing access is always done by SW.

Table 42.4 shows access protection to Data Flash memory. ID authentication to access is same as user area of Code flash memory. But the ID to authenticate becomes Data Flash ID.

Table 42.5 shows access protection to Extra Area (Configuration Setting, Security Setting, Block Protection Setting). ID authentication for reading/writing access is done by SW. Protection ID of Configuration Setting Area, see **Section 41.10, Configuration Setting Area (Option Bytes, Reset Vector)**. Protection ID of Security Setting Area, see **Section 42.6, Security Settings Area**. Protection ID of Block Protection Area, see **Section 41.11, Block Protection Area**.

Table 42.3 Access Protection to User Area from PEs

Mode	Sub Mode	Access from PEs		Access Via PEs Debug Master	
		Read	Erase/Write	Read	Erase/Write
Normal Operation Mode / User Boot Mode	Self-programming	Don't care	Customer ID authentication by SW	N/A	N/A
	OCD mode	Customer ID authentication by debugger I/F	Customer ID authentication by SW*1	Customer ID authentication by debugger I/F	Customer ID authentication by SW*1
	Except above	Don't care	N/A	N/A	N/A
Serial programming mode	—	Don't care	Don't care	N/A	N/A
	OCD mode (CAN/CAN-FD)	Customer ID authentication by debugger I/F	Don't care	Customer ID authentication by debugger I/F	Don't care

Note: Don't care: there are no access restriction, N/A: not applicable

Note 1. If target ID has been authenticated by debugger I/F, the ID certification by SW is unnecessary.

Table 42.4 Access Protection to Data Flash from PEs

Mode	Sub Mode	Access from PEs		Access Via PEs Debug Master	
		Read	Erase/Write	Read	Erase/Write
Normal Operation Mode / User Boot Mode	Self-programming	Don't care	Data Flash ID authentication by SW	N/A	N/A
	OCD mode	Data Flash ID authentication by debugger I/F	Data Flash ID authentication by SW*1	Data Flash ID authentication by debugger I/F	Data Flash ID authentication by SW*1
	Except above	Don't care	N/A	N/A	N/A
Serial programming mode	—	Don't care	Don't care	N/A	N/A
	OCD mode (CAN/CAN-FD)	Data Flash ID authentication by debugger I/F	Don't care	Data Flash ID authentication by debugger I/F	Don't care

Note: Don't care: there are no access restriction, N/A: not applicable

Note 1. If target ID has been authenticated by debugger I/F, the ID certification by SW is unnecessary.

Table 42.5 Access Protection to Extra Area (Configuration Setting, Security Setting, Block Protection Area) from PEs

Mode	Sub Mode	Access from Pes		Access Via PEs Debug Master	
		Read	Erase/Write	Read	Erase/Write
Normal Operation Mode / User Boot Mode	Self-programming	ID authentication by SW	ID authentication by SW	N/A	N/A
	OCD mode	ID authentication by SW*1	ID authentication by SW*1	ID authentication by SW*1	ID authentication by SW*1
	Except above	ID authentication by SW	N/A	N/A	N/A
Serial programming mode	—	ID authentication by SW	ID authentication by SW	N/A	N/A
	OCD mode (CAN/CAN-FD)	ID authentication by SW*1	ID authentication by SW*1	ID authentication by SW*1	ID authentication by SW*1

Note: Don't care: there are no access restriction, N/A: not applicable

Note 1. If target ID has been authenticated by debugger I/F, the ID certification by SW is unnecessary.

42.5.1 Block Protection Function

- User Area Erase/Write Protection

Code flash can be protected from Erase/Write by Customer ID authentication. This function can be set by 4 bit per Block of code flash. **Figure 42.2** shows an example of Block Protection structure. Once set, the OTP setting cannot be released.

- Data Flash Erase/Write Protection

Data Flash can be protected from Erase/Write by Data Flash ID authentication. This function can be set by DPROT. (see **Section 42.6.5, S_OPBT4 — Data Flash ID Related Option Bytes**)

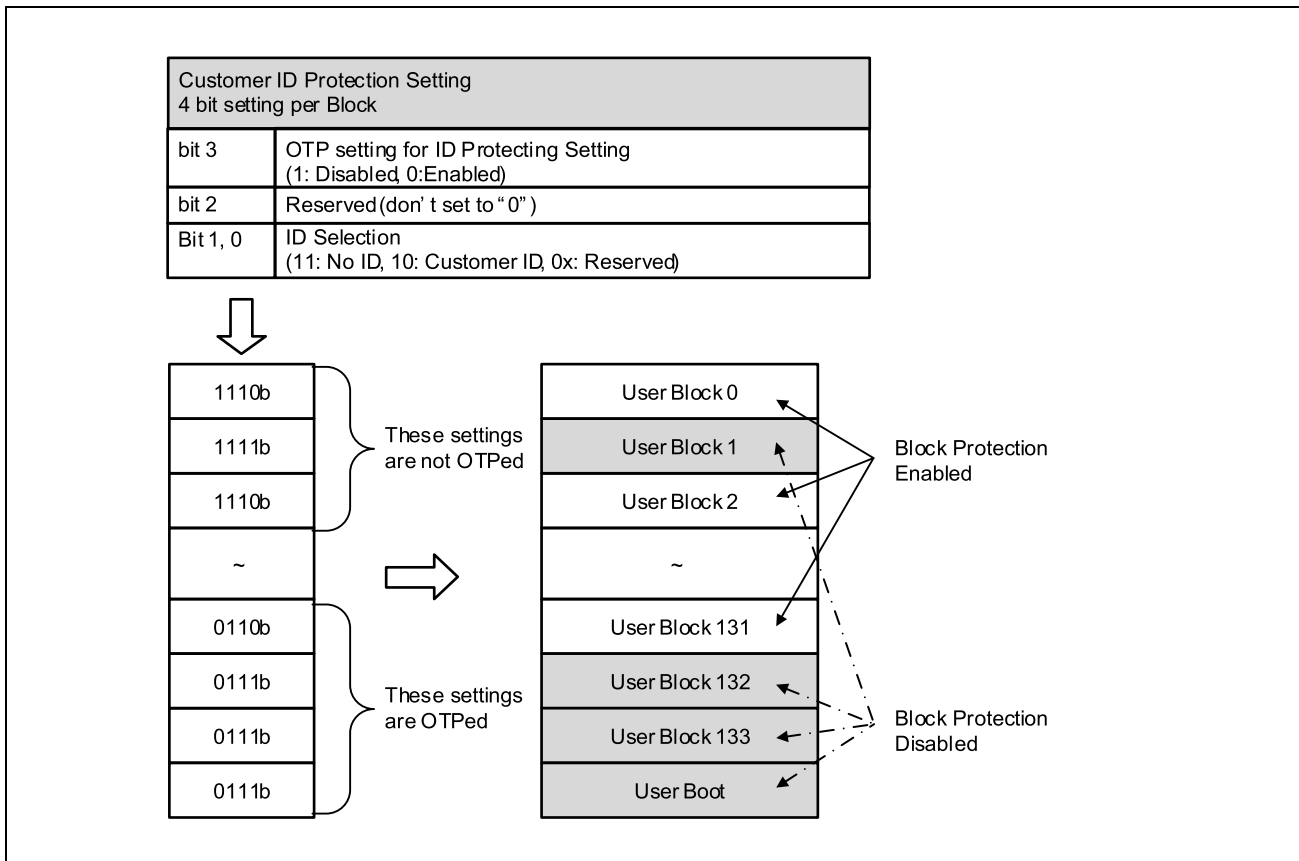


Figure 42.2 Block Protection Function Structure

Table 42.6 Block Protection Setting Mapping*1

Address	Bit	Protection Area
FF32 2080	[3:0]	User Block 0
	[7:4]	User Block 1
FF32 2081	[3:0]	User Block 2
	[7:4]	User Block 3
FF32 2082	[3:0]	User Block 4
	[7:4]	User Block 5
FF32 2083	[3:0]	User Block 6
	[7:4]	User Block 7
:	:	:
FF32 20C2	[7:4]	User Block 133
FF32 20C3	[3:0]	Reserved
	[7:4]	User Boot

Note 1. Refer to **Section 41.11, Block Protection Area**.

42.5.2 OTP (One Time Programmable) Function

OTP can be individually set for each block of the user area and the user boot area of code flash memory. When the OTP setting is made for an area (When OTP bit is “0”), programming by serial programming and by self programming is prohibited. Once set, the OTP setting cannot be released.

Table 42.7 shows Mapping of OTP setting for User area and User boot area.

Table 42.7 Mapping of OTP Setting (User Area and User Boot Area)*1

Address	Bit	Protection Area	
FF32 2040	[0]	User Block 0	
	[1]	User Block 1	
	[2]	User Block 2	
	[3]	User Block 3	
	:	:	:
	[31]	User Block 31	
FF32 2044	[0]	User Block 32	
:	:	:	
	[31]	User Block 63	
:	:	:	
FF32 2050	[0]	User Block 128	
	[1]	User Block 129	
	:	:	:
	[5]	User Block 133	
	[6]	reserved	
	:	:	:
	[30]	reserved	
[31]	User Boot		

Note 1. Refer to **Section 41.11, Block Protection Area**.

Table 42.8 and **Table 42.9** shows Mapping of OTP setting for extra Area.

Table 42.8 Mapping of OTP Setting (Security Settings Area)*¹

Address	Bit	Protection Area
FF30 0040	[0]	FF30 0080[31:0]
	[1]	FF30 0084[31:0]
	[2]	FF30 0088[31:0]
	[3]	FF30 008C[31:0]
:	:	:
	[31]	FF30 00FC[31:0]
FF30 0044	[0]	FF30 0100[31:0]
:	:	:
	[31]	FF30 017C[31:0]
FF30 0048	[0]	FF30 0180[31:0]
:	:	:
	[31]	FF30 01FC[31:0]

Table 42.9 Mapping of OTP Setting (Configuration Setting Area)*²

Address	Bit	Protection Area
FF32 0040	[0]	FF32 0060[31:0]
	[1]	FF32 0064[31:0]
	[2]	FF32 0068[31:0]
	[3]	FF32 006C[31:0]
:	:	:
	[23]	FF32 00BC[31:0]

Note 1. Refer to **Section 42.6, Security Settings Area**.

Note 2. Refer to **Section 41.10, Configuration Setting Area**.

42.5.3 Test Mode Entry Protection

Test mode is used to perform production test or failure analysis by Renesas. The authentication scheme to enter the test mode is configurable by Security settings area.

The password can be select from TESTID (that only Renesas uses) or C-TESTID to enter the test mode by Security settings area. For details, see *Section 16.8*, in the *RH850/E2x ICUMD User's Manual*.

42.6 Security Settings Area

The flash memory has the extended Extra area (Option Bytes) to store a data specified by the user for security settings. Changes in settings become effective after release from the reset state. (Refer to **Section 10.3.4, FACL reset transfer.**) OTP Setting of Security Settings Area becomes effective without reset. **Table 42.10** shows Security settings area. For setting and reading option bytes, see *the Renesas Flash Programmer Flash Programming Software User's Manual*, or *the RH850/E2x-FCC1 Flash Memory User's Manual: Hardware Interface*.

Table 42.10 Security Settings Area

Name	Address	Initial State of Shipped Product.*1	Write/Erase Protection ID*3	Read Protection ID*3*7
OTP Setting of Security Settings Area*2*8	FF30 0040 _H to FF30 0048 _H	FFFF FFFF _H	—	—
Reserved	FF30 004C _H to FF30 007C _H	FFFF FFFF _H *5	—	—
OCD ID Related Option Bytes (S_OPBT0)	FF30 0080 _H	FFFF FFFF _H	OCD ID	—
TEST ID /C-TEST ID Related Option Bytes (S_OPBT1)	FF30 0084 _H	FFFF FFFF _H	C-TEST ID	—
S_OPBT2	FF30 0088 _H	FFFF FFFF _H	Customer ID	—
Customer ID Related Option Bytes (S_OPBT3)	FF30 008C _H	FFFF FFFF _H	Customer ID	—
Data Flash ID Related Option Bytes (S_OPBT4)	FF30 0090 _H	FFFF FFFF _H	Data Flash ID	—
Serial Programmer ID Related Option Bytes (S_OPBT5)	FF30 0094 _H	FFFF FFFF _H	Serial Programmer ID*6	—
RHSIF ID Related Option Bytes (S_OPBT6)	FF30 0098 _H	FFFE FFFF _H	RHSIF ID	—
S_OPBT7	FF30 009C _H	FFFF FFFF _H	OCD ID & Customer ID*4	—
OCD ID (OCDIDn)	FF30 00A0 _H to FF30 00BC _H	All FFFF FFFF _H	OCD ID	OCD ID
Serial Programmer ID (SPIDn)	FF30 00C0 _H to FF30 00DC _H	All FFFF FFFF _H	Serial Programmer ID*6	Serial Programmer ID*6
Customer ID (CUSTOMERIDn)	FF30 00E0 _H to FF30 00FC _H	All FFFF FFFF _H	Customer ID	Customer ID
Data Flash ID (DATAFLASH IDn)	FF30 0100 _H to FF30 011C _H	All FFFF FFFF _H	Data Flash ID	Data Flash ID
C-TEST ID (CTESTIDn)	FF30 0120 _H to FF30 013C _H	All FFFF FFFF _H	C-TEST ID	C-TEST ID
RHSIF ID (RHSIFIDn)	FF30 0140 _H to FF30 015C _H	All FFFF FFFF _H	RHSIF ID	RHSIF ID
Reserved	FF30 0160 _H to FF30 01DC _H	FFFF FFFF _H *5	—	—
Reserved*9	FF30 01E0 _H to FF30 01FC _H	*9	*9	*9

Note 1. This is the initial value of the shipped product. The value can be modified.

Note 2. The detail of this function, see **Section 42, Basic Hardware Protection (BHP)**.
(**Table 42.8** shows Mapping of OTP setting for extra Area.)

Note 3. When all bits of Protection ID are "1" or "0", the ID authentication is no needed to set.

Note 4. When CFRPF is set (CFRPF bit is 0).

Note 5. It is prohibited to change from a read value.

Note 6. When the all bits are "1", the ID authentication is needed to connect a flash memory programmer.

Note 7. When Sub mode is OCD mode, the Customer ID authentication is needed to read.

Note 8. Set OTP after programming target data, because OTP setting becomes effective without reset.

Note 9. There are OPBT of ICUMD. For details, see *the RH850/E2x ICUMD User's Manual*.
When ICUMD is unused, please do not write it.

42.6.1 S_OPBT0 — OCD ID Related Option Bytes

OCD ID authentication is needed to write this option byte.

When all bits of OCD ID are “1” or “0”, the ID authentication is no needed to change the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	JPDBGIF_EN	DRDY_EN	—	—	—	—	—	CPUBTMSK_EN	—	—	—	DBG_OVCAN_DIS	—	—	—	—
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PFC_RESO_CFG	—	—	—	OIDDIS
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by Configuration Programming.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 42.11 OCD ID Related Option Bytes Register Contents

Bit Position	Bit Name	Function
31	JPDBGIF_EN	Security level 2 setting 0: Set security level 2 (can't use debug interface) 1: Unset security level 2 (can use debug interface)
30	DRDY_EN	Switching of the DRDY use for Nexus 0: Not use DRDY(GPIO) 1: Use DRDY
29 to 25	—	Reserved (The setting value should be 1.)
24	CPUBTMSK_EN	PE0 boot mask enable (when the TRST=H mode only) 0: PE0 boot mask disable 1: PE0 boot mask enable
23 to 21	—	Reserved (The setting value should be 1.)
20	DBG_OVCAN_DIS	Debug over Can disable 0: Debug over CAN enable 1: Debug over CAN disable
19 to 5	—	Reserved (The setting value should be 1.)
4	PFC_RESO_CFG	Switching $\overline{\text{RES_OUT}}$ pin function 0: Output to the internal reset signal 1: Output due to the Pin function setting
3 to 1	—	Reserved (The setting value should be 1.)
0	OIDDIS	OCD ID Authentication Disable 0: ID authentication is disabled (Can not entry Debug mode) 1: ID authentication is enabled

42.6.2 S_OPBT1 — TEST ID /C-TEST ID Related Option Bytes

C-TEST ID authentication is needed for write this option byte.

When all bits of C-TEST ID are “1” or “0”, the ID authentication is no needed to change the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMID
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TIDDIS
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 42.12 TEST ID /C-TEST ID Related Option Bytes Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved (The setting value should be 1.)
16	TMID	Test Mode Protection ID Selection 0: C-TEST ID is used in test mode ID authentication 1: TESTID is used in test mode ID authentication
15 to 1	—	Reserved (The setting value should be 1.)
0	TIDDIS	C-TEST ID Authentication Disable 0: ID authentication is always denied (Can not entry test mode) 1: ID authentication is enabled

NOTE

When TMID is set to “1”, test mode entry cannot be denied whether TIDDIS is set to “0”. To invalidate the entry of test mode, both TIDDIS and TMID have to be set to “0”.

42.6.3 S_OPBT2

For details of this S_OPBT2, see *the RH850/E2x ICUMD User's Manual*.

42.6.4 S_OPBT3 — Customer ID Related Option Bytes

Customer ID authentication is needed for write this option byte.

When all bits of Customer ID are “1” or “0”, the ID authentication is no needed to change the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFRPF
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 42.13 Customer ID Related Option Bytes Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved (The setting value should be 1.)
16	CFRPF	Code Flash Read Protection Flag Controlling section password authentication for code flash read access at debug mode 0: Section password authentication for read access is needed 1: Section password authentication for read access is not needed.
15 to 0	—	Reserved (The setting value should be 1.)

42.6.5 S_OPBT4 — Data Flash ID Related Option Bytes

Data Flash ID authentication is needed for write this option byte.

When all bits of Data Flash ID are “1” or “0”, the ID authentication is no needed to change the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DFRPF
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DPROT
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 42.14 Data Flash ID Related Option Bytes Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved (The setting value should be 1.)
16	DFRPF	Data Flash Read Protection Flag Controlling section password authentication for data flash read access at debug mode 0: Data Flash ID authentication for read access is needed 1: Data Flash ID authentication for read access is not needed.
15 to 1	—	Reserved (The setting value should be 1.)
0	DPROT	Data Flash Erase / Write protection Flag. Controlling section password authentication for data flash Erase / Write access 0: Data Flash ID authentication for Erase / Write access is needed 1: Data Flash ID authentication for Erase / Write access is not needed.

42.6.6 S_OPBT5 — Serial Programmer ID Related Option Bytes

Serial Programmer ID authentication is needed for write this option byte.

When all bits of Serial Programmer ID are “0”, the ID authentication is no needed to change the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CANBT D	SPD
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 42.15 Serial Programmer ID Related Option Bytes Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved (The setting value should be 1.)
1	CANBTD	CAN / CAN-FD Boot Strap Disable 0: CAN / CAN-FD Boot Strap of serial programmer command is disabled by boot firmware 1: CAN / CAN-FD Boot Strap of serial programmer command is enabled
0	SPD	Serial Programmer Disable 0: All serial programmer commands is disabled by boot firmware 1: Serial programmer commands is enabled

42.6.7 S_OPBT6 — RHSIF ID Related Option Bytes

RHSIF ID authentication is needed for write this option byte.

When all bits of RHSIF ID are “1” or “0”, the ID authentication is no needed to change the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSIF_IDAUTH_NEED
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	0*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 42.16 RHSIF Related Option Bytes Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved (The setting value should be 1.)
16	HSIF_IDAUTH_NEED	RHSIF Link Partner Authentication Setting 0: RHSIF ID authentication for Link Partner access is not needed 1: RHSIF ID authentication for Link Partner access is needed.
15 to 0	—	Reserved (The setting value should be 1.)

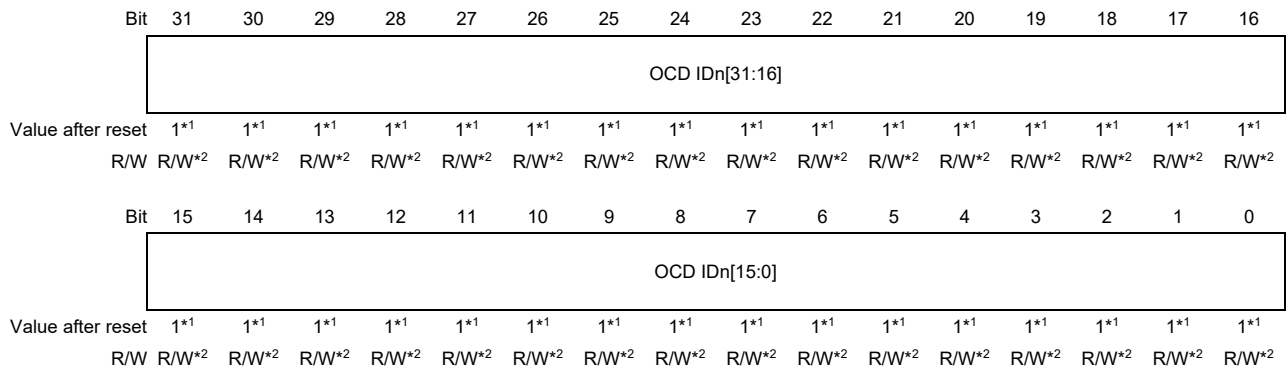
42.6.8 S_OPBT7

For details of this S_OPBT7, see *the RH850/E2x ICUMD User's Manual*.

42.6.9 OCDIDn — OCD ID (n = 0 to 7)

OCD ID authentication is needed for read / write this option byte.

When all bits of OCD ID are “1” or “0”, the ID authentication is no needed to change/read the setting.



Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 42.17 OCD IDn Register Contents

Bit Position	Bit Name	Function
31 to 0	OCD ID	ID Code for debug connection authentication. Bit mapping table is as follows: OCD ID[31:0]: OCDID0[31:0] OCD ID[63:32]: OCDID1[31:0] OCD ID[95:64]: OCDID2[31:0] OCD ID[127:96]: OCDID3[31:0] OCD ID[159:128]: OCDID4[31:0] OCD ID[191:160]: OCDID5[31:0] OCD ID[223:192]: OCDID6[31:0] OCD ID[255:224]: OCDID7[31:0]

42.6.10 SPIDn — Serial Programmer ID (n = 0 to 7)

Serial Programmer ID authentication is needed for read / write this option byte.

When all bits of Serial Programmer ID are “0”, the ID authentication is no needed to change/read the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Serial Programmer IDn[31:16]															
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Serial Programmer IDn[15:0]															
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

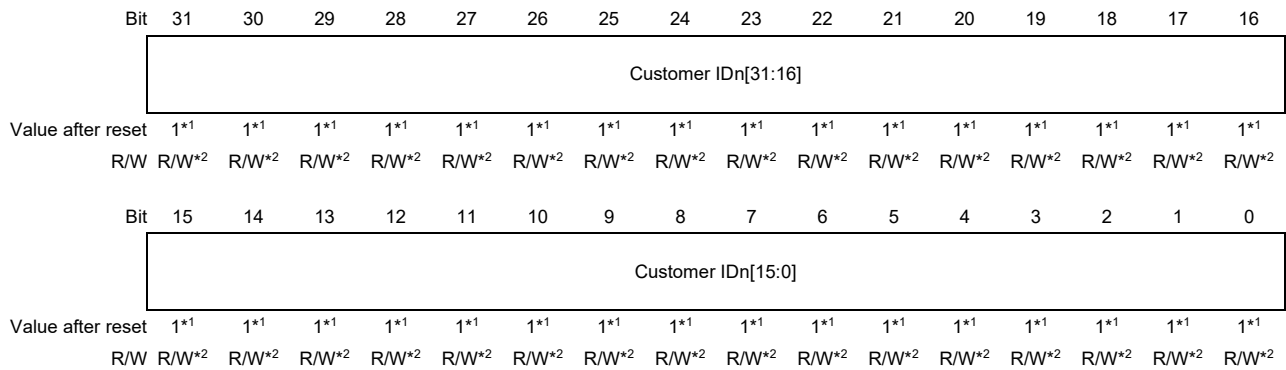
Table 42.18 SPIDn Contents

Bit Position	Bit Name	Function
31 to 0	Serial Programmer ID	ID Code for Serial Programmer authentication. Bit mapping table is as follows: Serial Programmer ID[31:0]: SPID0[31:0] Serial Programmer ID[63:32]: SPID1[31:0] Serial Programmer ID[95:64]: SPID2[31:0] Serial Programmer ID[127:96]: SPID3[31:0] Serial Programmer ID[159:128]: SPID4[31:0] Serial Programmer ID[191:160]: SPID5[31:0] Serial Programmer ID[223:192]: SPID6[31:0] Serial Programmer ID[255:224]: SPID7[31:0]

42.6.11 CUSTOMERIDn — Customer ID (n = 0 to 7)

Customer ID authentication is needed for read / write this option byte.

When all bits of Customer ID are “1” or “0”, the ID authentication is no needed to change/read the setting.



Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

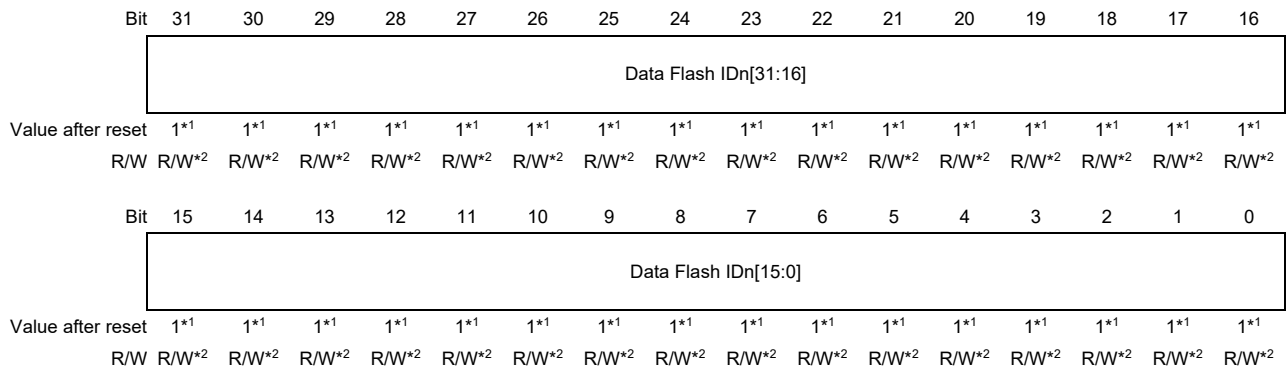
Table 42.19 CUSTOMERIDn Contents

Bit Position	Bit Name	Function
31 to 0	Customer ID	ID Code for Code Flash Protection, Configuration setting and etc. Bit mapping table is as follows: Customer ID[31:0]: CUSTOMERID0[31:0] Customer ID[63:32]: CUSTOMERID1[31:0] Customer ID[95:64]: CUSTOMERID2[31:0] Customer ID[127:96]: CUSTOMERID3[31:0] Customer ID[159:128]: CUSTOMERID4[31:0] Customer ID[191:160]: CUSTOMERID5[31:0] Customer ID[223:192]: CUSTOMERID6[31:0] Customer ID[255:224]: CUSTOMERID7[31:0]

42.6.12 DATAFLASH IDn — Data Flash ID (n = 0 to 7)

Data Flash ID authentication is needed for read / write this option byte.

When all bits of Data Flash ID are “1” or “0”, the ID authentication is no needed to change/read the setting.



Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

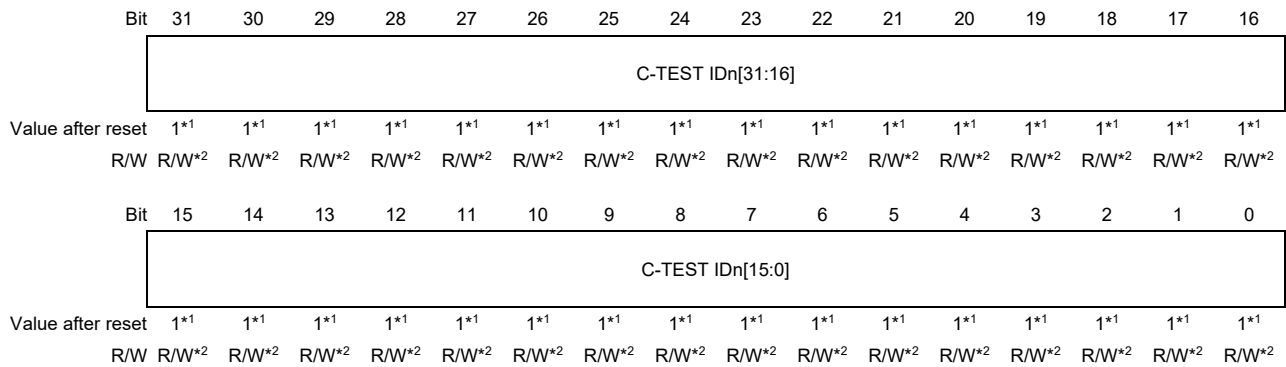
Table 42.20 DATAFLASH IDn Register Contents

Bit Position	Bit Name	Function
31 to 0	Data Flash ID	ID Code for Data Flash Protection. Bit mapping table is as follows: Data Flash ID[31:0]: DATAFLASHID0[31:0] Data Flash ID[63:32]: DATAFLASHID1[31:0] Data Flash ID[95:64]: DATAFLASHID2[31:0] Data Flash ID[127:96]: DATAFLASHID3[31:0] Data Flash ID[159:128]: DATAFLASHID4[31:0] Data Flash ID[191:160]: DATAFLASHID5[31:0] Data Flash ID[223:192]: DATAFLASHID6[31:0] Data Flash ID[255:224]: DATAFLASHID7[31:0]

42.6.13 CTESTIDn — C-Test ID (n = 0 to 7)

C-TEST ID authentication is needed for read / write this option byte.

When all bits of C-TEST ID are “1” or “0”, the ID authentication is no needed to change/read the setting.



Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACI command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 42.21 CTESTIDn Register Contents

Bit Position	Bit Name	Function
31 to 0	C-TEST ID	ID Code for test mode entry authentication. Bit mapping table is as follows: C-TEST ID[31:0]: CTESTID0[31:0] C-TEST ID[63:32]: CTESTID1[31:0] C-TEST ID[95:64]: CTESTID2[31:0] C-TEST ID[127:96]: CTESTID3[31:0] C-TEST ID[159:128]: CTESTID4[31:0] C-TEST ID[191:160]: CTESTID5[31:0] C-TEST ID[223:192]: CTESTID6[31:0] C-TEST ID[255:224]: CTESTID7[31:0]

42.6.14 RHSIFIDn — RHSIF ID (n = 0 to 7)

RHSIF ID authentication is needed for read / write this setting.

When all bits of RHSIF ID are “1” or “0”, the ID authentication is no needed to change/read the setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RHSIF IDn[31:16]															
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RHSIF IDn[15:0]															
Value after reset	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1	1*1
R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Note 1. This is the initial value of the shipped product. The value can be modified by setting the option bytes.

Note 2. Only when an FACL command or serial programming (ex. the Renesas Flash Programmer flash programming software) is used, setting (W) is possible.

Table 42.22 RHSIFIDn Register Contents

Bit Position	Bit Name	Function
31 to 0	RHSIF ID	ID Code for RHSIF Link Partner connection authentication. Bit mapping table is as follows: RHSIF ID[31:0]: HSIFID0[31:0] RHSIF ID[63:32]: HSIFID1[31:0] RHSIF ID[95:64]: HSIFID2[31:0] RHSIF ID[127:96]: HSIFID3[31:0] RHSIF ID[159:128]: HSIFID4[31:0] RHSIF ID[191:160]: HSIFID5[31:0] RHSIF ID[223:192]: HSIFID6[31:0] RHSIF ID[255:224]: HSIFID7[31:0]

Section 43 Intelligent Cryptographic Unit/Master (ICUMD)

Please refer to the separate volume about ICUMD (Security function).

Section 44 Secure Watchdog Timer (SWDT)

Please refer to the separate volume about SWDT.

Section 45 Debugging and Calibration

This microcontroller has an on-chip debug function. By using the on-chip debug emulator, programs can be debugged with the microcontroller mounted in the target system.

The debug functions incorporated in this microcontroller conform to IEEE-ISTO 5001™-2003, a Nexus debug interface standard.

Production devices support Nexus Class 1 debug functionality. Emulation devices support Nexus Class 3 debug functionality.

For details of Production devices, see **Section 1.4, Ordering Information, Table 1.1**.

For details of Emulation devices, see **Section 1.4, Ordering Information, Table 1.2**.

CAUTION

The debug functions described in this section are supported by the microcontroller but whether these functions can actually be used depends on the debugger. For details on debugging, see the user's manual of the debugger used.

45.1 Debug Interface

This microcontroller supports the following debug interfaces:

1. Nexus JTAG Interface
2. Low Pin Debug Interface (4-pin) (LPD4)
3. CAN Debug interface - A CAN port can be selected as the debug communication interface by using the 2 pin protocol version of LPD4.
4. In emulation devices, a 1-lane Aurora trace port is provided.
5. Trigger Input / Output pin ($\overline{\text{EVTI}}$ / $\overline{\text{EVTO}}$)

On-chip debugging can be performed by using these debug interfaces.

Table 45.1 Debug Interface Modes

Serial programming mode*1		Normal operating mode/user boot mode*1	Debug ports		Option bytes*3			Register*4	JP0 port usage
2-wire UART/CSJ*5	CAN/CAN-FD*5		$\overline{\text{TRST}}$	TDI(JP0_0)*2	DRDY_EN	JPDBGIF_EN	DBG_OV CAN_DISS	DBGOV CANC .DOC_EN	
1	0	0	x	x	x	x	x	x	Serial programming*7
0	1	0	0	x	x	x	x	x	GPIO*6
0	1	0	1	0	x	0	x	x	GPIO (security level 2*3)*6,*8
0	1	0	1	0	x	1	x	x	LPD4*6,*9
0	1	0	1	1	x	0	x	x	GPIO (security level 2*3)*6,*8
0	1	0	1	1	1	1	x	x	Nexus (with $\overline{\text{DRDY}}$)*6,*9
0	1	0	1	1	0	1	x	x	Nexus (without $\overline{\text{DRDY}}$)*6,*9
0	0	1	0	x	x	x	x	x	GPIO
0	0	1	1	0	x	0	x	x	GPIO (security level 2*3)*6
0	0	1	1	0	x	1	x	0	LPD4*9
0	0	1	1	1	x	0	x	x	JTAG Boundary Scan (security level 2*3)
0	0	1	1	1	1	1	x	0	Nexus (with $\overline{\text{DRDY}}$) & JTAG Boundary Scan*9
0	0	1	1	1	0	1	x	0	Nexus (without $\overline{\text{DRDY}}$) & JTAG Boundary Scan*9
0	0	1	1	x	x	1	0	1	GPIO (Debug Over CAN)

Note: x = don't care

Note 1. For details, see **Section 5, Operating Modes**.

- Note 2. Latch output at rising edge of $\overline{\text{TRST}}$.
- Note 3. For details, see **Section 42, Basic Hardware Protection (BHP)**.
- Note 4. For details, see **Section 45.8.1.5(1), DBOVCANC — Debug Over CAN Control Register**.
- Note 5. For details, see **Section 41, Flash Memory**.
- Note 6. Can be selected externally by using the boot firmware.
- Note 7. The function of the port can be switched between 2-wire UART and CSI by using the boot firmware.
- Note 8. The operation is not guaranteed if a reset is asserted while data is being output. When GPIO is used, $\overline{\text{TRST}}$ must be "0".
- Note 9. When the interface is to be switched, the JP0 port register settings must be their initial values.

45.2 Run Control Functions

(1) Debug functions

- Downloading a user-created program
- Reading and writing the memory and registers
- Running a user-created program starting at any address

(2) On-chip break functions

A maximum of 12 breakpoints can be specified at any execution address. Of the 12 breakpoints, a maximum of four breakpoints can be specified for the data access address.

The following break functions are supported:

- Relay break function: Each CPU can be configured to stop execution when another CPU hits a breakpoint.
- Individual break function: Each CPU can be configured to continue execution when another CPU hits a breakpoint.

CAUTION

Whether the individual break function can be used or not depends on the debugger.

(3) Software break function

Software breakpoints can be specified at any execution address.

(4) Forced break function

Execution of a user-created program can be interrupted forcibly by a debugger.

(5) Peripheral break control

The peripheral break function generates a stop request to the peripheral modules of the microcontroller if the user-created program is stopped, for instance upon a breakpoint hit.

(6) Forced reset function

This device (microcontroller) can be forcibly reset by a debugger.

(7) Reset mask function

A reset factor (external reset, software reset, and ECM reset) can be masked.

(8) Event detection function

Events can be detected based on execution address, data access address, data value, data value range, and sequential execution.

(9) Trigger input interface

This microcontroller incorporates an event trigger input interface to acknowledge external events. It can acknowledge an external event in response to an input from the $\overline{\text{EVTI}}$ pin.

(10) Trigger output interface

An event trigger output interface is included in this microcontroller to notify an external debug device of event detection. Output from the $\overline{\text{EVTO}}$ pin reports detection of an event trigger to the outside. Debug trigger registers control trigger output to an external debug device via a request from the CPU. Up to eight debug trigger registers are supported. Each register supports 32 software triggers.

When $\overline{\text{TRST}} / \overline{\text{LPDRST}}$ is low level, output system reset signal from $\overline{\text{EVTO}}$.

(11) Debug interrupt interface function

Execution of a user program can be forcibly suspended by asserting an input signal on the $\overline{\text{EVTI}}$ pin from the outside.

(12) Multi core debug function

The following multi-core debug functions are supported for each CPU: synchronization functions (including reset, execution, and break), synchronous setting, and simultaneous tracing for multiple cores.

45.3 Calibration Functions

This microcontroller includes emulation RAM as emulation memory for the on-chip flash memory.

(1) Real-time RAM monitoring (RRM)

The memory can be read during program execution. Because this read access uses debug-dedicated DMA, it has minimal effect on program execution.

(2) Dynamic memory modification (DMM)

The memory can be written during program execution. Because this write access uses debug-dedicated DMA, it has minimal effect on program execution.

(3) Emulation RAM (ERAM) - only in emulation devices

For details, see **Section 9, RAM**.

45.4 Trace Control Functions

This microcontroller provides several trace functions including branch PC trace and data trace for each CPU, and DMA data trace.

(1) Trace RAM - only in emulation devices

This microcontroller has 64 KB of trace RAM. The trace information in the trace RAM is accessible via Nexus and LPD4 debug interfaces.

(2) Trace filter RAM - only in emulation devices

- Advanced data trace filtering allows selection of small 4 byte sections of data which can be individually enabled for data trace.
- Trace filter RAM does not have ECC.

(3) Software trace

This function enables the obtaining of user program execution histories, data changes, etc. The software trace information can be output to the trace RAM or via Aurora debug interface.

(4) Software trace

This function enables the obtaining of user program execution histories, data changes, etc. The software trace information can be output via LPD4 debug interfaces.

(5) Trace over reset

Remaining trace data is output after a reset.

(6) Global timestamp information

The global timestamp information is the absolute time from (absolute time 0) when the trace buffer is cleared.

(7) The trace message format in conforms to the Nexus standard.

(8) Windowed instruction trace or data trace

45.5 Performance Measurement Function

This microcontroller provides a two-event interval time difference measurement function to measure the time difference between the points at which two events occur.

(1) Two-event interval time difference measurement function

- Provides a function to measure the time difference (maximum time, minimum time, accumulation time) between two events (measurement start event to measurement stop event).
- Provides a function to measure the number of measurements (the number of measurement stop events) for two-event intervals (measurement start event to measurement stop event).
- Provides a threshold value violation detection function for two-event intervals (measurement start event to measurement stop event).
- Measurement clock: Debugging clock or CPU clock
- Measurement counter: 32-bit counter × 4 channels
- Measurement items: Maximum value, minimum value, accumulation value, measurement count (measurement stop event count)
- Count conditions: Number of debugging clock cycles, measurement count (measurement stop event count); number of CPU clock cycles, instruction execution or other events.
- Measurement limitations: When the interval between the measurement start event and the measurement stop event (from measurement start to measurement stop) or between a measurement stop event and the next measurement start event (from measurement stop to measurement start) is short, it may not be possible to correctly detect the measurement start or measurement stop events due to synchronization processing (from the system clock to the debugging clock).

Note that the required interval for detection depends on the system clock and the debugging clock frequencies.

- Measurement of the cache hit rate
- Measurement of the number of CPU cycles when a pipeline stall has occurred

45.6 Debug Support Function

(1) Hot plug-in function

Debugging can be started in normal operating mode without external reset input.

(2) Security function

To prevent the contents of the flash memory from being read by an unauthorized person, a 256-bit ID code must be written to the microcontroller. If the code the user inputs when starting a debugger does not match the ID code in the microcontroller, the flash memory cannot be accessed.

For details, refer to **Section 42, Basic Hardware Protection (BHP)**.

(3) Halt after reset

The debugger can control the start of CPU0 after internal reset release.

This must be set by using an option byte. For details, see **Section 42, Basic Hardware Protection (BHP)**.

(4) ICUMD debug function

For details, see *the RH850/E2x ICUMD User's Manual*.

(5) GTM debug function

For details, see **Section 31, GTM**.

(6) Device ID function

Device ID (device identification) information can be acquired via the debug interface (Nexus).

(7) Reset output pin

The reset output pin ($\overline{\text{RES_OUT}}$) reflects the state of the internal reset condition.

For details, see **Section 2, Pin Function**.

45.7 Peripheral Break Control

The peripheral break function generates a stop request to the peripheral modules of the microcontroller if the user-created program is stopped, for instance upon a breakpoint hit.

During a peripheral break, the peripheral modules operate as follows:

1. Modules that stop unconditionally regardless of the EPC.SVSTOP setting
WDTB0, WDTB1, SWDT0
2. Modules that stop when EPC.SVSTOP = 1 and EPC_SVSTOPn setting
OSTM0, OSTM1, OSTM2, ENCA0, ENCA1, ATU-V, GTM

45.8 Debugging-Related Registers

45.8.1 Registers equipped according to the product specifications

45.8.1.1 Register Base Address

Table 45.2 Register Base Address

Base Address Name	Base Address	Bus Group
<EPC_base>	FF0B 0000H	Peripheral Group 9

45.8.1.2 Clock Supply

Table 45.3 Clock Supply

Unit Name	Unit Clock Name	Clock Supply Name
EPC	PCLK	CLK_HBUS

45.8.1.3 Reset Source

Table 45.4 Reset Source

Unit Name	Register Name	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
EPC	All registers	✓	✓	✓	✓	—	—

45.8.1.4 List of Registers

Table 45.5 List of Function Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection
EPC	Debug Over CAN Control Register	DBGOV Canc	<EPC_base> + F0 _H	8, 16, 32	

45.8.1.5 Product Specification Registers

(1) DBGOVCANC — Debug Over CAN Control Register

This register controls Debug Over CAN.

The setting of an option byte is necessary to enable the setting of this register.

For details, see **Section 42, Basic Hardware Protection (BHP)**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DOC_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 45.6 DBGOVCANC Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved When read, the value after reset is returned. When writing, write the value after reset.
0	DOC_EN	This bit is used to enable and disable Debug Over CAN. 0: Debug Over CAN is disabled. 1: Debug Over CAN is enabled.

45.9 Aurora Trace Interface

45.9.1 Overview

Aurora is lightweight link-layer protocol for communicating over high-speed serial lanes.

Nexus introduces this protocol in IEEE-ISTO 5001-2012 (Nexus standard).

This product supports a simplex TX-only interface. Only one lane is used.

Aurora for this product is fully compatible with Nexus standards and Xilinx's Aurora Protocol spec V2.2.

Table 45.7 shows the Aurora reference documents for this product.

Table 45.7 Reference Documents

Reference Document	Document Name	Reference
Aurora 8b / 10b protocol specification V2.2	aurora_8b10b_protocol_spec_sp002.pdf	http://www.xilinx.com/
The Nexus 5001 Forum™ Standard for a Global Embedded Processor Debug Interface V3.0.1	ieee_isto_5001_2012_301_Final.pdf	http://www.nexus5001.org .

45.9.2 Summary of Specifications

Summarized specifications of Aurora for this product are shown below.

The Aurora interface for this product is compliant with IEEE-ISTO 5001-2012 (Nexus standard). Its specifications are also compliant with the Nexus standards.

Table 45.8 Summary of Specifications

Item	Specifications	Note
I/F	1 lane	Simplex Tx only
Protocol specifications	Aurora protocol spec V2.2 compatible	
DC balanced order	8b / 10b	
IO level	AC coupling only	
AC specifications	IEEE-ISTO 5001-2012 compliant	Compliant with Nexus standards
PLL multiplier ratio	20:1	
Baud rate	1.25 Gbps	Reference CLK = 62.5 MHz
Data format	Streaming	
Back channel	Sideband (JTAG, LPD)	
Flow control	None	
Error detection and correction	None	

45.10 Cautions on Using On-Chip Debugger

(1) Handling of devices used for debugging

Do not install a device that was used for debugging on a mass-produced product. This is because the flash memory will be rewritten during system debugging and thus the write/erase count of flash memory cannot be guaranteed.

(2) This product does not support hot plug out for power off (including removal of the connector) of the debug tool during debug mode. Do not turn off the power of the Nexus tool (including removal of the connector) in debug mode.

(3) When ending on-chip debugging, set the $\overline{\text{TRST}}$ pin and external reset pin to low level.

(4) When a debugger is used, the program written to the microcontroller before the OCD emulator is ready to communicate with the microcontroller is executed from the reset vector. To stop the operation of the program, release the $\overline{\text{TRST}}$ before $\overline{\text{RES_IN}}$.

Section 46 Boundary Scan

This section contains a generic description of the boundary scan.

The first part in this section describes all specific properties of this product, such as the clock supply, reset sources, etc.

The remainder of the section describes the functions and registers of the boundary scan.

46.1 Features of Boundary Scan

46.1.1 Products that Incorporate Boundary Scan

This microcontroller has the JTAG interface and provides the boundary scan function conforming to the IEEE1149.1 standard.

46.1.2 Register Base Address

The JTAG interface has no base address because none of the registers can be accessed by the CPU.

46.1.3 Clock Supply

The boundary scan clock supply is shown in the following table.

Table 46.1 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
Boundary scan	TCK	tck

46.1.4 Interrupt Requests

Interrupt request is none.

46.1.5 Reset Sources

The boundary scan reset sources are shown below. Boundary scan is initialized by the following reset sources. For details about register symbols, refer to **Table 46.4**.

Table 46.2 Reset Sources

Unit Name	Register Symbol	Reset Condition					
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	Module Reset	JTAG Reset
Boundary scan	SDIR	—	—	—	—	—	✓
	SDID	—	—	—	—	—	—
	SDBPR	—	—	—	—	—	—
	SDBSR	—	—	—	—	—	—

46.1.6 External Input/Output Signals

External input/output signals of boundary scan are listed below.

Table 46.3 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal Name
Boundary scan		
TCK	Serial data input clock signal	TCK
TMS	Mode select input signal	TMS
$\overline{\text{TRST}}$	Reset input signal	$\overline{\text{TRST}}$
TDI	Serial data input signal	TDI
TDO	Serial data output signal	TDO

46.2 Overview

Boundary scan is a test method defined in the IEEE standard 1149.1 that is used to test the connection between the devices mounted on the printed-circuit board. The boundary scan of the RH850/E2x-FCC1 conforms to IEEE Std 1149.1-2001.

46.2.1 Functional Overview

This microcontroller supports the following.

- Five test signals (TCK, TDI, TDO, TMS, and $\overline{\text{TRST}}$)
- TAP controller
- Instruction register
- Bypass register
- Boundary scan register

The JTAG interface has six instructions.

- BYPASS mode
Test mode conforming to the IEEE 1149.1
- EXTEST mode
Test mode conforming to the IEEE 1149.1
- SAMPLE/PRELOAD mode
Test mode conforming to the IEEE 1149.1
- CLAMP mode
Test mode conforming to the IEEE 1149.1
- HIGHZ mode
Test mode conforming to the IEEE 1149.1
- IDCODE mode
Test mode conforming to the IEEE 1149.1

46.2.2 Block Diagram

The following block diagram shows the main components of the boundary scan.

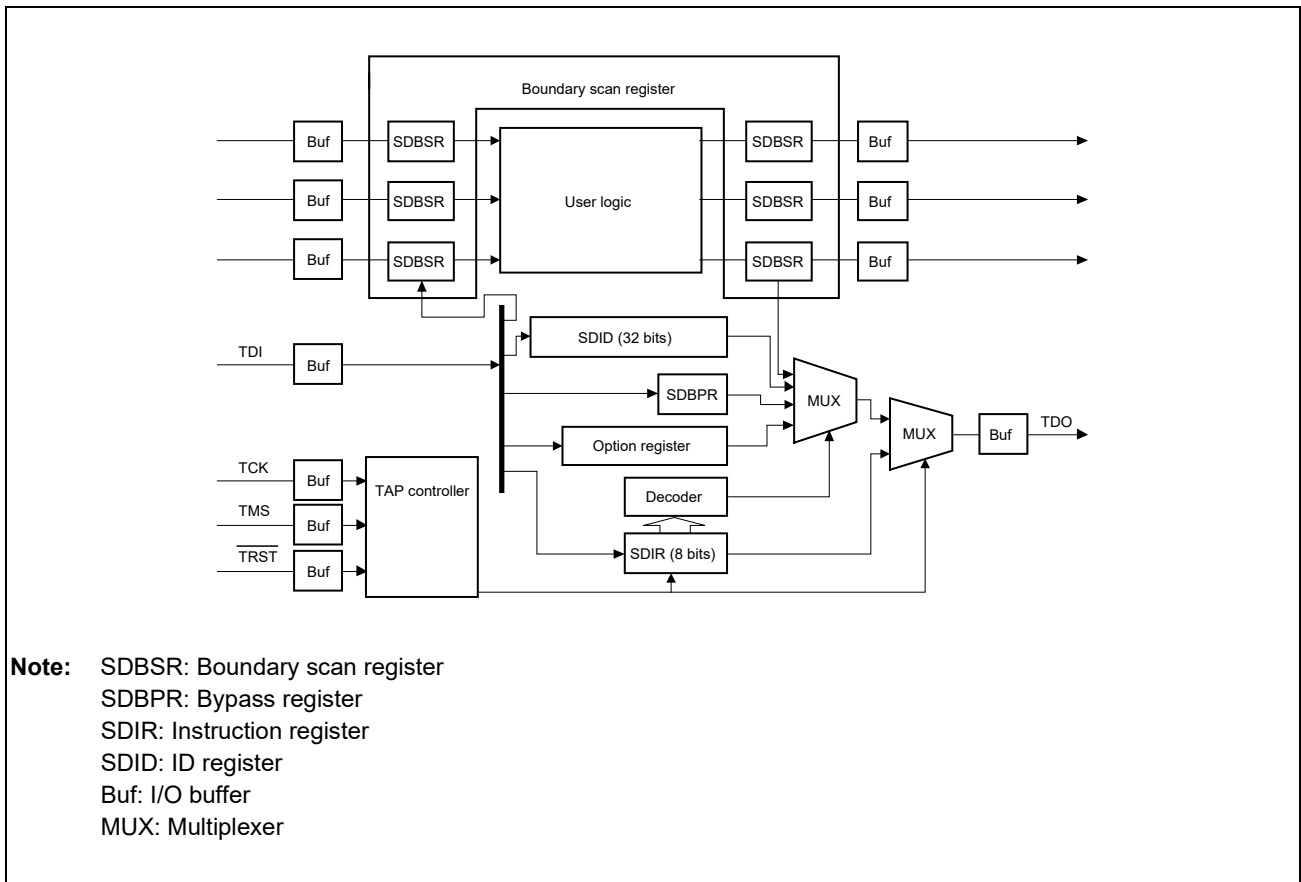


Figure 46.1 Block Diagram of JTAG Interface

46.3 Registers

46.3.1 List of Registers

The boundary scan registers are listed in the following table.

Table 46.4 List of Registers

Module Name	Register Name	Symbol	Address*1	Access	Access Protection
BSCAN	Instruction register	SDIR	—	8	—
	ID register	SDID	—	32	—
	Bypass register	SDBPR	—	1	—
	Boundary scan register	SDBSR	—	*2	—

Note 1. This register can be access by TAP controller.

Note 2. The register is a shift register to control external pins, and thus it can't define access size.

The ID register (SDID) information is listed in the following table.

For the ID register of mass produced products, please to each target specification.

Table 46.5 List of ID Register (SDID) Value

Product Type Name	Description	SDID Value
E2M 400MHz BGA292	R7F702002AEABG	2839 F447 _H
E2M 400MHz BGA373	R7F702002AEABA	283A 0447 _H
E2x-FCC1 for E2M BGA292	R7F702Z02CEDBG	483A D447 _H
E2x-FCC1 for E2M BGA373	R7F702Z04CEDBA	483D 1447 _H

Instructions can be serially transferred from the serial data input pin (TDI) and input to the instruction register (SDIR). The bypass register (SDBPR) is a 1-bit register, to which TDI and TDO are connected in BYPASS mode, CLAMP mode, and HIGHZ mode. The boundary scan register (SDBSR) is connected to TDI and TDO in SAMPLE/PRELOAD mode and EXTEST mode. The ID code register (SDID) is a 32-bit register, from which the ID code is output via TDO in IDCOD mode.

46.3.2 Details of Registers

46.3.2.1 SDIR — Instruction Register

SDIR is an 8-bit register that holds a boundary scan instruction. SDIR is initialized by a low-level input of $\overline{\text{TRST}}$ or in the TAP Test-Logic-Reset state. Operation is not guaranteed when any reserved instruction is set in this register.

Table 46.6 Boundary Scan Instructions

SDIR[7:0]	Description
0000 0000	JTAG EXTEST
0100 0000	JTAG SAMPLE/PRELOAD
1101 0000	JTAG CLAMP
1000 0000	JTAG HIGHZ
0101 0101	JTAG IDCODE (value after reset)
1111 1111	JTAG BYPASS
Other than above	Reserved

46.3.2.2 SDID — ID Register

SDID is a 32-bit register with a device specific ID.

SDID can be read via the JTAG interface pin when the IDCODE instruction is set but cannot be written to.

46.3.2.3 SDBPR — Bypass Register

SDBPR is a 1-bit register to go through the boundary scan register. When SDIR is set to BYPASS mode, CLAMP mode or HIGHZ mode, SDBPR is connected between TDI and TDO. The value after reset is undefined. SDBPR is not initialized by a reset as described in **Section 46.1.5, Reset Sources**.

46.3.2.4 SDBSR — Boundary Scan Register

SDBSR is a shift register located on the PADs for controlling the external I/O pins. When SDIR is set to SAMPLE/PRELOAD or EXTEST mode, SDBSR is connected between TDI and TDO. The value after reset is undefined. SDBSR is not initialized by a reset as described in **Section 46.1.5, Reset Sources**.

46.4 Operation

46.4.1 TAP Controller

The following block diagram shows the TAP controller internal states.

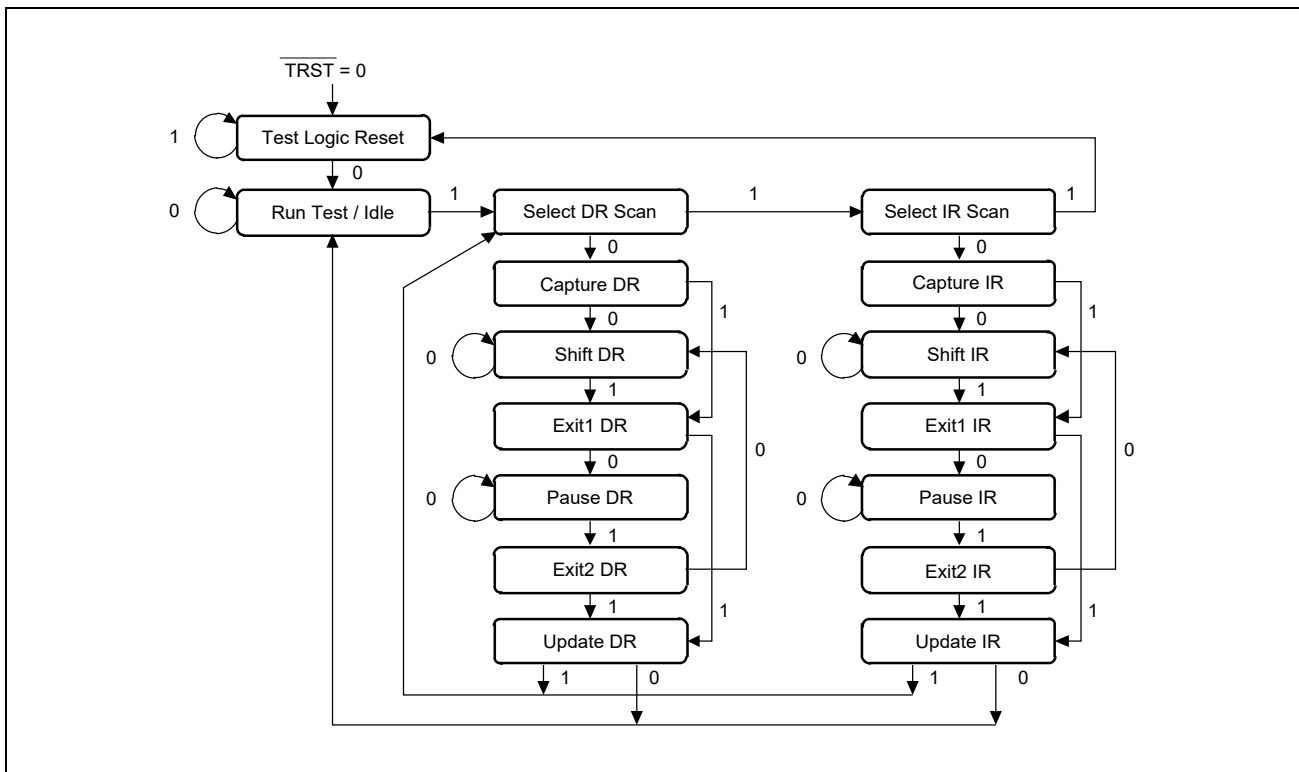


Figure 46.2 TAP Controller Internal States*1

Note 1. Transition is made according to the TMS value at the rising edge of TCK. The TDI value is sampled at the rising edge of TCK and is shifted at the falling edge. TDO is in the high-impedance state in the states other than Shift-DR (DataRegister) and Shift-IR. Asserting $\overline{\text{TRST}}$ causes transition to the test-logic-reset state asynchronously with TCK.

46.4.2 Supported Instructions

46.4.2.1 BYPASS

The BYPASS instruction is a standard instruction indispensable to bypass register operation. This instruction shortens the shift path to achieve high-speed serial data transfer of other devices on the printed-circuit board. During execution of this instruction, the test circuit has no effect on the system circuit.

46.4.2.2 SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction is used to input the value to the boundary scan register from the internal circuits of this LSI; to output the value from the scan path; and to load data onto the scan path. During execution of this instruction, the level of the input pin of this LSI is sent to the internal circuits as is, and the value of the internal circuits is output to the outside via the output pin as is. Executing this instruction has no effect on the system circuit of this LSI.

The SAMPLE operation allows taking in the snapshots of the value to be transferred to the internal circuits from the input pin or the value to be transferred to the output pin from the internal circuits to the boundary scan register, and allows reading the snapshots from the scan path. Snapshots can be taken in without preventing the normal operation of this LSI.

The PRELOAD operation allows setting the initial value to the parallel output latch of the boundary scan register from the scan path prior to the EXTEST instruction. If the EXTEST instruction is executed without PRELOAD operation, an undefined value is output from the output pin until the first scan sequence is completed (transfer to the output latch) because the parallel output latch value is always output to the output pin by the EXTEST instruction.

46.4.2.3 EXTEST

The EXTEST instruction is used to test the external circuits when this LSI is mounted on the printed circuit board. When this instruction is executed, the output pin is used to output the test data (previously set by the SAMPLE/PRELOAD instruction) from the boundary scan register to the printed-circuit board; whereas the input pin is used to take in the test result from the printed-circuit board to the boundary scan register. When the EXTEST instruction is executed N times for testing, the test data for the N th execution is scanned in at the $(N - 1)$ th scan-out. If the data is loaded onto the boundary scan register of the output pin in the Capture-DR state of this instruction, it is not used for testing the external circuits (replaced through shift operation).

46.4.2.4 CLAMP

When the CLAMP instruction is selected, the output pin outputs the boundary scan register value that has been previously set by the SAMPLE/PRELOAD instruction. While the CLAMP instruction is selected, the boundary scan register retains the previous state regardless of the TAP controller state. The bypass register is connected between TDI and TDO and operates in the same manner as when the BYPASS instruction is selected.

46.4.2.5 HIGHZ

When the HIGHZ instruction is selected, all the output pins go to the high-impedance state. While the HIGHZ instruction is selected, the boundary scan register retains the previous state regardless of the TAP controller state. The bypass register is connected between TDI and TDO and operates in the same manner as when the BYPASS instruction is selected.

46.4.2.6 IDCODE

The IDCODE instruction sets the JTAG interface pins to IDCODE mode, which is defined by the JTAG standard. When the JTAG interface is initialized (by asserting TRST or placing TAP in the test-logic-reset state), IDCODE mode is set.

46.4.3 Supported Pins

46.4.3.1 Supported Pin Information

There are supported pins related to the JTAG interface as follows.

- General purpose I/O pins.
- External reset output pin.
- Non maskable input pin.
- DRDY pin.

There are restrictions related to the JTAG interface as follows.

- NC pins.
- Power supply and GND pins.
- Reference voltage pins of A/D converters.
- Analog input pins.
- The pin to be connected to a stabilizing capacitor.
- External reset input pin.
- Debug control pins.
- Mode setting pins.
- Crystal pins.
- Error output pin.

The following table shows the list of pins not subjected to boundary scan.

Table 46.7 Pins Not Subjected to Boundary Scan

Type	Pin Name
Reference voltage pins of A/D converters.	ADSVREFH, ADSVREFL, A0VREFH, A1VREFH ^{*3} , A2VREFH, A3VREFH ^{*3}
Analog input pins.*1	ANxxx, CANxxxN, CANxxxP, DSANxxxN, DSANxxxP
The pin to be connected to Capacitor.	ADSVCL, RAMSVCL
External reset input pin.	$\overline{\text{RES_IN}}$
Debug control pins.	CICREFP ^{*2} , CICLEFN ^{*2} , TODP0 ^{*2} , TODN0 ^{*2} , TCK, TDI, TDO, TMS, $\overline{\text{TRST}}$, $\overline{\text{AURORES}}$ ^{*2}
Mode setting pins.	MD0, MD1
Crystal pins.	XTAL, EXTAL
Error output pin.	$\overline{\text{ERROROUT_M}}$

Note 1. See **Section 2.6.6, Dedicated Analog Pin Table** about detail names of analog input pins.

Note 2. The pin is implemented only E2x-FCC1 (for E2M BGA373, E2M BGA292).

Note 3. The pin is not implemented in BGA292.

46.4.3.2 Usage Note

1. Once an instruction is set, it is not modified until another instruction is issued. To issue the same instructions twice in a row, insert an instruction that has no effect on chip operation (such as BYPASS mode) between the instructions.
2. To start the system in boundary scan mode, de-assert $\overline{\text{TRST}}$ while $\overline{\text{RES_IN}}$ is low. Also, be sure to set TMS to low before de-asserting $\overline{\text{TRST}}$. Furthermore, “CPUBTMSK_EN” of OCD ID Related Option Bytes Register bit setting value must be “1” at that time. See **Section 42.6.1, S_OPBT0 — OCD ID Related Option Bytes** about “CPUBTMSK_EN”.
3. For details of boundary scan timing constraint, see *Section 1.3.1, Power ON / OFF Timing*, and *Section 1.3.14, JTAG / NEXUS Timing in the RH850/E2x-FCC1 Electrical Characteristics, User’s Manual: Hardware*.
4. If serial transfer is performed exceeding the number of bits of the register connected between TDI and TDO, the data that is input from TDI is output from TDO as is.
5. If the serial transfer sequence is corrupted, be sure to assert $\overline{\text{TRST}}$. In this case, transfer starts again from the beginning regardless of the point of transfer corruption.
6. Data is output via TDO synchronized with the falling edge of TCK.
7. To facilitate debugging, route $\overline{\text{TRST}}$ on the board in such a way that patterns can be easily cut.
8. No instructions can be accepted in power-off standby mode.
9. When using the boundary scan, the ICUM function must be disable.
10. The switching of NEXUS and boundary scan require the initialize by $\overline{\text{RES_IN}}$ and $\overline{\text{TRST}}$.

Section 47 Electrical Characteristics

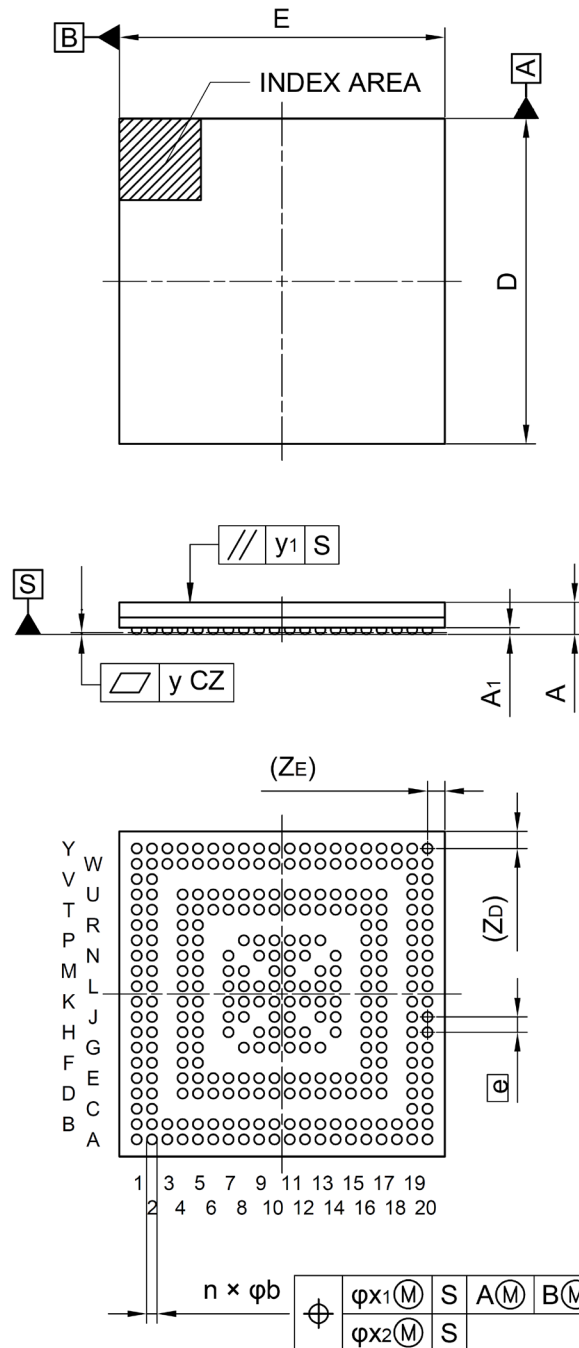
Please refer to the separate volume about Electrical Characteristics.

Appendix A List of Registers

Please refer to Appendix file of “List_of_Registers.xlsx”.

Appendix B Package

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-FBGA292-17x17-0.80	PRBG0292GD-A	1.0



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	16.90	17.00	17.10
E	16.90	17.00	17.10
A	—	—	2.00
A ₁	0.30	0.35	0.40
e	—	0.80	—
b	0.49	0.54	0.59
x ₁	—	—	0.20
x ₂	—	—	0.08
y	—	—	0.10
y ₁	—	—	0.20
Z _D	—	0.90	—
Z _E	—	0.90	—
n	—	292	—

Figure B.1 BGA 292 package for E2x-FCC1 and E2M

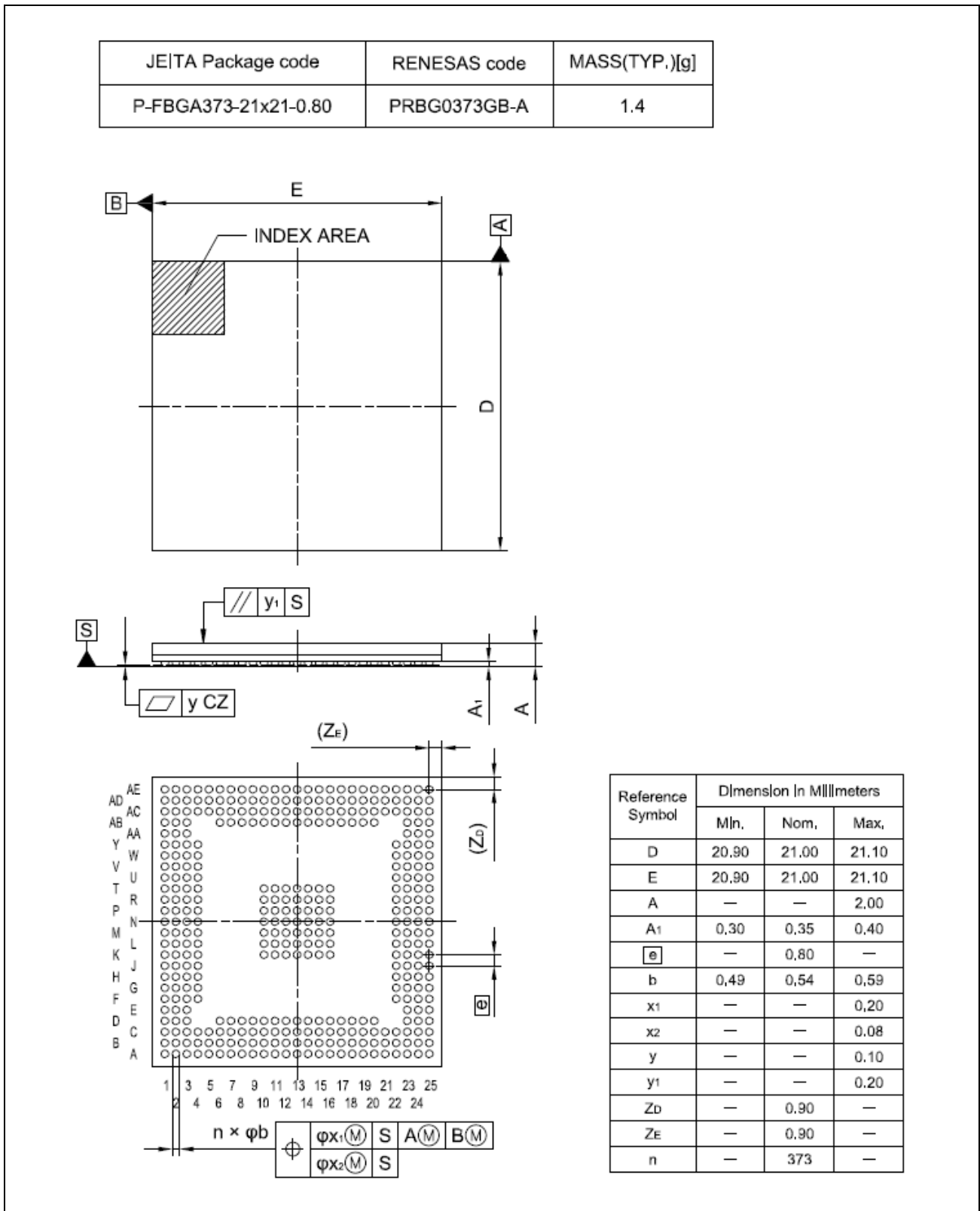


Figure B.2 BGA 373 package for E2x-FCC1 and E2M

RH850/E2x-FCC1 User's Manual: Hardware

Publication Date: Rev.0.10 May 31, 2016
Rev.1.30 Dec 26, 2019

Published by: Renesas Electronics Corporation

**SALES OFFICES****Renesas Electronics Corporation**<http://www.renesas.com>Refer to "<http://www.renesas.com/>" for the latest and detailed information.**Renesas Electronics Corporation**

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

Renesas Electronics America Inc.1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351**Renesas Electronics Canada Limited**9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004**Renesas Electronics Europe GmbH**Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327**Renesas Electronics (China) Co., Ltd.**Room 101-T01, Floor 1, Building 7, Yard No. 7, 8th Street, Shangdi, Haidian District, Beijing 100085, China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679**Renesas Electronics (Shanghai) Co., Ltd.**Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai 200333, China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999**Renesas Electronics Hong Kong Limited**Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022**Renesas Electronics Taiwan Co., Ltd.**13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670**Renesas Electronics Singapore Pte. Ltd.**80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300**Renesas Electronics Malaysia Sdn.Bhd.**Unit No 3A-1 Level 3A Tower 8 UOA Business Park, No 1 Jalan Pengaturcara U1/51A, Seksyen U1, 40150 Shah Alam, Selangor, Malaysia
Tel: +60-3-5022-1288, Fax: +60-3-5022-1290**Renesas Electronics India Pvt. Ltd.**No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700**Renesas Electronics Korea Co., Ltd.**17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

RH850/E2x-FCC1